©Faculdade de Letras da Universidade Federal de Minas Gerais
© Stefan Th. Gries

FACULDADE DE LETRAS, UFMG

Diretora

Graciela Inéz Ravetti de Góméz

Vice-Diretora

Sueli Maria Coelho

Coordenadora da Câmara de Pesquisa

Andréa Machado de Almeida Mattos

Conselho Editorial

Janayna Maria da Rocha Carvalho

Valéria Sabrina Pereira

Marcos Antônio Alexandre

Elidea Lucia Almeida Bernadino

Gabriel Caixeta de Paula Muller

Diagramação

Crysttian Arantes Paixão

Revisão

Heliana Ribeiro de Mello, Crysttian Arantes Paixão e Júlia Zara

Revisão Técnica

Crysttian Arantes Paixão

Projeto Gráfico

GlobalPrint Editora

Ficha catalográfica elaborada pelos Bibliotecários da Biblioteca FALE/UFMG

G848s.Pm Gries, Stefan Thomas, 1970-
Estatística com R para linguística : uma introdução prática /
Stefan Th. Gries ; organizadora: Heliana R. Mello ; tradução:
Heliana R. Mello... [et al.]. – Belo Horizonte : FALE/UFMG,
2019.
312 p. : il., tabs., p&b.
Tradução de: Statistics for linguistics with R : a practical
introduction.
ISBN: 978-85-7758-361-4
978-85-7758-363-8 (ebook)
Bibliografia: p. 309-312.
1. Linguística – Métodos estatísticos. 2. R (Linguagem de
programação de computador). 3. Linguística de corpus. I. Mello,
Heliana. II. Título.

CDD : 410.285

Estatística com R para a linguística

Stefan Th. Gries

Organizadora: Heliana R. Mello

**Tradução: Heliana R. Mello, Crysttian A. Paixão, André L. E. Souza e Júlia
Zara**

**Belo Horizonte
FALE/UFMG
2019**

Agradecimentos

Somos muito gratos a Stefan Th. Gries por ter estimulado e apoiado o projeto de tradução de sua obra *Statistics for Linguistics with R* para o português desde o primeiro momento em que lhe propusemos essa ideia.

Contamos com o apoio e o interesse de muitos estudantes em nossos cursos de graduação e de pós-graduação, além dos nossos orientandos, ao longo dos anos, a quem agradecemos pelas sugestões e questionamentos. É na sala de aula que podemos melhor dimensionar as necessidades didáticas e os materiais necessários ao suporte de aprendizado dos nossos alunos. O principal objetivo deste livro é justamente oferecer uma introdução prática aos estudantes de linguística que desejam utilizar métodos estatísticos viabilizados por códigos em R.

Heliana Mello agradece à equipe de professores e pesquisadores do Laboratório de Estudos Empíricos e Experimentais da Linguagem – UFMG pelo estímulo à continuidade da expansão do uso de métodos estatísticos e da linguagem R em suas pesquisas linguísticas.

Manifestamos aqui a nossa admiração e agradecimento aos colegas brasileiros que vêm estimulando o ensino de estatística aos jovens pesquisadores e estudantes da área de linguística. Nomeamos aqui Plínio Almeida Barbosa (UNICAMP), Livia Oushiro (UNICAMP) e Pablo Arantes (UFSCAR) como representantes exemplares desse esforço.

Agradecemos à Faculdade de Letras-UFMG, representada por sua Diretoria e por sua Câmara de Apoio à Pesquisa, por viabilizar a publicação deste livro.

Somos gratos aos dois pareceristas anônimos que avaliaram a nossa tradução por suas sugestões. Quaisquer erros que permaneçam são de nossa responsabilidade.

Manifestamos, também, o nosso agradecimento ao Jumarino Lopes e ao Israel Silva, colegas da Biblioteca da FALE-UFMG, pela gentil ajuda e suporte imprescindível com os processos de registro e catalogação deste livro.

H. Mello, C. Paixão, A. Souza e J. Zara

Sumário

Prefácio	11
Apresentação	13
1 Fundamentos da pesquisa empírica	15
1.1 Introdução	15
1.2 A importância da pesquisa quantitativa em linguística	17
1.3 O desenho e a lógica por trás de estudos quantitativos	18
1.3.1 <i>Scouting</i>	20
1.3.2 Hipóteses e operacionalização	21
1.3.2.1 Hipótese científica em formato textual	22
1.3.2.2 Operacionalização de suas variáveis	24
1.3.2.3 Hipóteses científicas em formato estatístico/matemático	27
1.3.3 Coleta e armazenamento de dados	28
1.3.4 A decisão	31
1.3.4.1 Valores unicaudais de distribuições de probabilidades discretas	34
1.3.4.2 Valores- <i>p</i> bicaudais de distribuições de probabilidades discretas	36
1.3.4.3 Extensão: distribuições de probabilidades contínuas	42
1.4 O desenho de um experimento fatorial: introdução	47
1.5 O design de um experimento fatorial: um outro exemplo	51
2 Fundamentos Básicos do R	55
2.1 Introdução e instalação	55
2.2 Funções e seus argumentos	58
2.3 Vetores	62
2.3.1 Como criar vetores no R	62
2.3.2 Como carregar e salvar um vetor no R	67
2.3.3 Como editar vetores no R	69
2.4 Fatores	77
2.4.1 A criação de fatores no R	77
2.4.2 Como carregar e salvar fatores no R	78
2.4.3 Como editar fatores no R	78
2.5 Estruturas de dados	81
2.5.1 Como criar um <i>data frame</i> no R	81
2.6 Como carregar e salvar <i>data frame</i> no R	83
2.6.1 Como editar uma <i>data frame</i> no R	85
2.7 Um pouco sobre programação: condicionais e <i>loops</i>	91
2.7.1 Expressões condicionais	91
2.7.2 <i>Loops</i>	92
2.8 Escrevendo suas próprias funções	94
3 Estatística descritiva	99
3.1 Estatística Univariada	99
3.1.1 Dados de frequência	99
3.1.1.1 Gráficos de dispersão e gráficos de linha	101
3.1.1.2 Gráficos de pizza	105
3.1.1.3 Gráficos de barra	106
3.1.1.4 Gráficos de Pareto	107
3.1.1.5 Histogramas	108
3.1.2 Distribuições cumulativas empíricas	109
3.1.3 Medidas de tendência central	110

3.1.3.1	Moda	110
3.1.3.2	Mediana	111
3.1.3.3	Média aritmética	111
3.1.3.4	Média geométrica	112
3.1.4	Medidas de dispersão	113
3.1.4.1	Entropia Relativa	115
3.1.4.2	Amplitude	115
3.1.4.3	Quantis e Quartis	116
3.1.4.4	Desvio médio	117
3.1.4.5	O desvio padrão/variação	118
3.1.4.6	O coeficiente de variação	119
3.1.4.7	Funções de <i>Summary</i>	120
3.1.4.8	Erro padrão	122
3.1.5	Centralização e padronização (escores-z)	123
3.1.6	Intervalo de Confiança	125
3.1.6.1	Intervalos de confiança para médias aritméticas	125
3.1.6.2	Intervalo de confiança para porcentagens	127
3.2	Estatística bivariada	128
3.2.1	Frequências e tabulação de dados	128
3.2.1.1	Gráficos de barra e mosaicos	130
3.2.1.2	<i>Spineplots</i>	130
3.2.1.3	Gráficos de linha	131
3.2.2	Médias	132
3.2.2.1	<i>Boxplots</i>	133
3.2.2.2	Gráficos de interação	134
3.2.3	Coeficientes de correlação e regressão linear	137
4	Estatística Analítica	147
4.1	Distribuições e frequências	150
4.1.1	Ajuste de distribuição	150
4.1.1.1	Uma variável dependente (de escala de proporção)	150
4.1.1.2	Uma variável dependente (categórica/nominal)	153
4.1.2	Testes para diferenças/independência	158
4.1.2.1	Uma variável dependente (ordinal/intervalar/de escala de proporção) e uma variável independente (nominal). Amostras independentes	158
4.1.2.2	Uma variável dependente (nominal/categórica) e uma variável independente (nominal/ categórica). Amostras independentes	163
4.1.2.3	Uma variável dependente (nominal/categórica). Amostras dependentes	174
4.2	Dispersões	177
4.2.1	Teste de aderência para uma variável dependente (escala de proporção)	178
4.2.2	Uma variável dependente (escala de proporção) e uma variável independente (nominal)	180
4.3	Médias	185
4.3.1	Testes de Aderência	185
4.3.1.1	Uma variável dependente (variável de escala de proporção)	185
4.3.1.2	Uma variável dependente (variável ordinal)	188
4.3.2	Testes para diferenças/independência	192
4.3.2.1	Uma variável dependente (escala de proporção) e uma variável independente (nominal) (amostras independentes)	193
4.3.2.2	Uma variável dependente (escala de proporção) e uma variável independente (nominal)(amostras dependentes)	198
4.3.2.3	Uma variável dependente (ordinal) e uma variável independente (nominal) (amostras independentes)	203

4.3.2.4	Uma variável dependente (ordinal) e uma variável independente (nominal) (amostras dependentes)	208
4.4	Coeficientes de correlação e regressão linear	212
4.4.1	A significância da correlação produto-momento	213
4.4.2	A significância do Tau de Kendall	217
4.5	Correlação e causalidade	218
5	Seleção de Métodos Multifatoriais e Multivariados	221
5.1	As noções de interação e modelo(seleção)	221
5.1.1	Interações	221
5.1.2	Seleção de modelo	226
5.1.3	Formulação do primeiro modelo	226
5.1.3.1	Seleção de um modelo final	230
5.2	Modelo Linear	232
5.2.1	Um modelo linear com preditor binário	234
5.2.2	Um modelo linear com um preditor categórico	239
5.2.3	Um modelo linear com um preditor numérico	243
5.2.4	Um modelo linear com dois preditores categóricos	245
5.2.5	Um modelo linear com um preditor categórico e um numérico	247
5.2.6	Um modelo linear com dois preditores numéricos	250
5.2.7	Processo de seleção de um modelo linear com múltiplos preditores	251
5.2.8	Modelos de regressão logística binária	259
5.2.9	Regressão logística com preditor binário	261
5.2.10	Regressão logística com um preditor categórico	269
5.2.11	Regressão logística para um preditor numérico	270
5.2.12	Uma regressão logística com dois preditores categóricos	272
5.2.13	Uma regressão linear com um preditor categórico e um numérico	273
5.2.14	Uma regressão logística com dois preditores numéricos	275
5.3	Outros modelos de regressão	278
5.3.1	Uma regressão logística ordinal com um preditor categórico e outro numérico	279
5.3.2	Um regressão multinomial com um preditor cagetórico e um numérico	283
5.3.3	Regressão de Poisson com um preditor categórico e um numérico	285
5.4	Medidas repetidas: um indicador	288
5.4.1	Uma variável independente dentro de participantes/itens	289
5.4.2	Duas variáveis independentes aninhadas em sujeitos/itens	291
5.4.3	Duas variáveis independentes, uma entre, uma dentro de sujeitos/itens	291
5.4.4	Efeitos mixos / Modelos de vários níveis	292
5.5	Análise hierárquica de <i>cluster</i>	295
6	Epílogo	307
	Referências	309

Prefácio

Este livro é uma versão revisada e estendida de [Gries \(2009b\)](#). Há quatro tipos principais de mudanças. A primeira e mais importante é uma revisão completa do Capítulo 5. Depois de ministrar dezenas de *workshops* e *bootcamps* sobre estatística para a Linguística com o R, percebi que os aspectos mais difíceis da modelagem de regressão para iniciantes são (i) entender a lógica do processo de modelagem, (ii) interpretar os resultados numéricos (especialmente com contrastes diferentes), e (iii) visualizá-los de forma reveladora.

Assim, todas as seções sobre modelagem de regressão foram reescritas desde o início. Além disso, há agora uma síntese geral de aspectos mais teóricos da modelagem que, espero, facilitará a compreensão de muitas coisas. O segundo conjunto de alterações diz respeito ao capítulo 1 e ao capítulo 4. No primeiro, discuto agora de forma melhor as noções de testes unilaterais e bilaterais (bicaudais); no último, discuto um conjunto de perguntas e uma ferramenta de visualização que podem ajudar na escolha dos testes estatísticos adequados para determinado estudo. Em terceiro lugar, acrescentei uma pequena seção sobre aspectos de programação e sobre como usuários podem escrever suas próprias funções e, tendo explicado isso, também disponibilizo para os leitores algumas funções simples que escrevi para meu próprio uso.

Portanto, esta edição não apenas corrige os erros que leitores me apontaram (e sou muito grato por eles terem feito isso, e espero que eu não tenha cometido muitos novos erros...), mas também adiciona vários pequenos ajustes e mudanças que considere necessários com base nos *workshops* de estatística e nas aulas que tenho ministrado nos últimos anos. Alguns desses ajustes estão no livro, mas muitos também estão “escondidos” no arquivo de códigos, de modo que você só irá vê-los se - como deve fazer - estudar o conteúdo deste livro usando paralelamente os códigos recomendados. Finalmente, todos os códigos do livro estão agora em um único arquivo, o que tornará a manipulação dos códigos e a busca por funções muito convenientes, eliminando também a necessidade de um índice de nomes de funções.

Espero que goste e se beneficie deste livro e das muitas mudanças incluídas nesta revisão. Como de costume, gostaria de agradecer à equipe da De Gruyter Mouton que apoiou, na verdade que propôs, a ideia de uma segunda edição. Além disso, mais uma vez devo agradecer à equipe de desenvolvimento do R (*R Development Core Team*) e a muitos colaboradores nas correções de falhas e em desenvolvimento pacotes para R e, também mais uma vez, ao R. Agradeço ao Harald Baayen por ter me exposto ao R pela primeira vez. Não consigo imaginar como seria a minha pesquisa se ele não tivesse feito isso ...

Stefan Gries

Apresentação

A tradução apresentada neste livro refere-se à obra autoral de Stefan Th. Gries, intitulada *Statistics for Linguistics with R: a practical introduction*, Editora de Gruyter Mouton, publicada em sua segunda edição (2013), originalmente em língua inglesa. O projeto de tradução do livro para o português foi iniciado por André Souza e Heliana Mello há quase uma década, quando a versão em inglês ainda se encontrava em sua primeira edição. Ao longo do caminho, juntaram-se ao projeto Crysttian A. Paixão e Júlia Zara, que agregaram experiência, conhecimento de estatística e entusiasmo à empreitada. É com muita satisfação que nós quatro trazemos ao leitor de língua portuguesa a possibilidade de estudar estatística voltada à pesquisa linguística, em língua materna.

A estatística aplicada às humanidades e ciências sociais ganhou espaço internacionalmente já há algumas décadas. No Brasil, na área de linguística, ainda há uma grande lacuna na formação dos alunos de graduação e até mesmo dos de pós-graduação em métodos estatísticos. Poucas são as universidades que oferecem disciplinas de estatística integradas ao currículo regular de linguística, o que traz como consequência uma grande defasagem em termos de formação, que se reflete nas pesquisas linguísticas desenvolvidas no Brasil.

Esta tradução vem juntar-se a outras iniciativas em curso nas universidades brasileiras para prover material de referência para o desenvolvimento de competências básicas em análise estatística ao aluno de graduação e pós-graduação em linguística. A linguagem de programação R, bem como a interface gráfica de programação R, são oferecidas gratuitamente e podem ser instaladas em qualquer tipo de sistema operacional. A comunidade internacional de programadores em R é numerosa e guiada pela mentalidade de partilha e disponibilização de código. Desse modo, há muito suporte disponível, listas de discussão e pessoas mais experientes que podem ajudar o estudante novato com suas dúvidas.

Esperamos que esse livro seja útil aos estudantes de linguística e que com ele a aprendizagem seja eficaz e motivadora.

Fundamentos da pesquisa empírica

Afirmo muitas vezes que, se você medir aquilo de que está falando e expressar em números, você conhece alguma coisa sobre o assunto; mas, quando você não o pode exprimir em números, seu conhecimento é pobre e insatisfatório.
William Thomson, Lord Kelvin. (<<http://hum.uchicago.edu/jagoldsm/Webpage/index.html>>)

1.1 Introdução

Este é um livro de introdução à estatística. No entanto, já existe um grande número de livros de introdução à estatística – por que precisamos de mais um? Bem, o presente livro é diferente de outros livros de introdução à estatística por vários motivos:

- foi escrito especialmente para linguistas: existem muitas introduções à estatística para psicólogos, economistas, biólogos e etc., mas poucos que, como este, explicam métodos e conceitos estatísticos com base em questões de caráter linguístico e para linguistas;
- explica como executar praticamente todos os métodos estatísticos manualmente e por meio de um programa estatístico, mas ao mesmo tempo não requer nenhum conhecimento profundo em matemática e nem requer horas de memorização de fórmulas e equações complexas – muitos livros introdutórios dedicam bastante tempo explicando fundamentos matemáticos (e, consequentemente, complicam mais as coisas para um leitor iniciante), outros não explicam fundamentos e já entram em discussões sobre software bem elaborados, o que frequentemente esconde a lógica de testes estatísticos por trás de uma boa GUI¹ (Interface Gráfica do Usuário);
- ele não somente explica conceitos estatísticos, testes e gráficos, como também explica o *design* de tabelas para a tabulação e análise de dados, apresenta um resumo dos principais aspectos da pesquisa experimental, bem como uma revisão de literatura sobre o assunto;
- ele utiliza programas de computador gratuitos (principalmente o R): muitos utilizam introduções ao SAS ou ao SPSS, que apresentam muitas desvantagens tais como (i) os usuários devem comprar licenças caras que dependem do número de funções que eles oferecem, da quantidade de dados que ele consegue manipular e de quanto tempo pretende utilizá-lo; (ii) estudantes e professores geralmente só conseguem utilizá-los nas universidades – que geralmente detêm as licenças de utilização; (iii) eles dependem da companhia que os produzem, caso apareça algum defeito ou em caso de atualizações;

¹ acrônimo GUI de *Graphical User Interface*

- este livro faz tudo isso de uma maneira acessível e informal: tentamos evitar o uso de linguagem muito técnica; o uso do programa será ilustrado de maneira detalhada e didática. Haverá “pausas para pensar” (do inglês “think breaks”), avisos e exercícios (com chave de resposta no site do livro), recomendações para leituras adicionais e etc., com o intuito de fazer o material o mais acessível possível.

O principal objetivo do livro é oferecer a você uma introdução a como fazer pesquisa científica de caráter quantitativo. A estrutura do livro é a seguinte: O Capítulo 1 oferece uma introdução aos fundamentos da pesquisa quantitativa: o que são variáveis, hipóteses, qual é o escopo estrutural de um estudo de caráter quantitativo e que tipo de lógica subjaz o processo de fazer pesquisa quantitativa e que tipo de raciocínio a ela instiga, como você obtém bons dados experimentais e em que tipo de formato deve armazenar seu dados?

O Capítulo 2 oferece uma visão panorâmica sobre linguagem de programação e sobre o ambiente computacional do programa R, o programa que será utilizado no livro para a execução das análises estatísticas e produção de gráficos: como criar, carregar e manipular dados para preparar a sua análise?

O Capítulo 3 explica os fundamentos básicos da estatística descritiva: como descrever seus dados, que padrões podem emergir dos dados e como representar esses padrões graficamente? O Capítulo 4 explica os fundamentos de estatística analítica: como saber se os resultados obtidos através dos testes estatísticos realmente significam alguma coisa ou emergiram como resultado do acaso? O Capítulo 5 introduz vários modelos multifatoriais, ou seja, modelos em que várias relações de causa-e-efeito são investigadas simultaneamente. Enquanto este capítulo vai ensinar-lhe um monte de coisas, serão apresentados com alguns métodos e apontadas referências adicionais algumas vezes.

Além dos capítulos com os exercícios e as pausas para pensar, o site do livro ^{2 3} são fontes importantes para consultas adicionais. Você deverá acessar o site para fazer o download dos arquivos de exercício, arquivos de dados, respostas, erratas e etc. No site ⁴ você encontrará um grupo de discussão “StatForLing with R”. Assim que você se tornar membro do grupo, você poderá:

- postar perguntas sobre estatística para linguistas (e obter resposta de alguém);
- enviar sugestões para extensões e melhorias para o livro, bem como enviar exercício e banco de dados;
- informar os autores e as outras pessoas sobre problemas que encontrar (e claro receber a mesma informação de outros leitores). Caso os comandos do R ou códigos encontrados no livro estejam diferentes dos encontrados nos sites, é bem provável que os encontrados no site são os corretos e mais atualizados.

Por último, devemos mencionar um fato importante: é impossível aprender a fazer análises estatísticas apenas lendo um livro sobre análises estatísticas. Você deve fazer análises estatísticas.

Você não irá aprender como fazer análise estatística se ler esse livro (ou qualquer outro livro sobre introdução à estatística) 15 minutos antes de apagar a luz e ir dormir. Na verdade, qualquer livro que promete essa façanha está “distorcendo” a realidade por razões comerciais. Recomendamos que a partir do Capítulo 2, você leia o livro ao lado de um computador de forma que você possa digitar os comandos do R diretamente no programa, testar funções e códigos dos arquivos encontrados no site do livro algumas vezes (especialmente no Capítulo 5), os *scripts* com os códigos fornecem informações extras importantes, pedaços de códigos adicionais, e diversas sugestões sobre como explorar mais seus dados utilizando representações gráficas e etc., e por vezes os arquivos com exercícios irão fornecer mais sugestões e mais gráficos. Mesmo que você não compreenda os códigos em uma primeira leitura, a prática direta irá ajudá-lo a apreender tudo que o livro tem a oferecer.

² <<http://www.linguistics.ucsb.edu/faculty/stgries/research/sflwr/sflwr.html>>

³ <<http://groups.google.com/group/statforling-with-r/web/statistics-for-linguists-with-r>>

⁴ <<http://groups.google.com/group/statforling-with-r>>

1.2 A importância da pesquisa quantitativa em linguística

Mencionei acima que o presente livro é uma introdução à pesquisa científica de caráter quantitativo. Quais são os principais objetivos desse tipo de pesquisa científica? Geralmente, podemos pensar em três objetivos que precisam ser descritos, pois (i) eles fazem parte de um corpo de conhecimento que todo pesquisador com algum tipo de rigor empírico deve estar ciente e (ii) eles são relevantes para a maneira como o presente livro está estruturado.

O primeiro objetivo é a descrição dos seus dados sobre algum fenômeno. Significa dizer que seus dados e os resultados devem ser reportados da maneira mais acurada e reveladora possível. Todos os métodos estatísticos descritos a seguir irão ajudá-lo a alcançar esse objetivo, principalmente os métodos descritos no Capítulo 3.

O segundo objetivo é a explicação dos seus dados, que geralmente depende das hipóteses sobre quais tipos de relações você espera encontrar nos dados. Muitas vezes, isso será suficiente para seus propósitos. No entanto, às vezes você pode também estar interessado em um terceiro objetivo: previsão: o que irá acontecer no futuro ou quando analisamos um corpo de dados diferente. Os Capítulos 4 e 5 irão introduzir métodos que tem como objetivo explicar e fazer previsões.

Quando pensamos nesses objetivos, parece surpreendente que esses métodos estatísticos não são altamente utilizados em linguística. Isso é ainda mais surpreendente quando pensamos que esses métodos são altamente utilizados em disciplinas com tópicos igualmente complexos, tais como psicologia, sociologia e economia. Em certa medida, essa situação tem haver com a maneira como a linguística se desenvolveu nos últimos dez anos, mas felizmente esse quadro está constantemente mudando agora. O número de estudos que utilizam metodologia quantitativa tem crescido (em todas as sub-áreas da linguística); o campo está sofrendo uma mudança de paradigma em direção à metodologias mais empíricas. Apesar da metodologia quantitativa ser bem comum em outras disciplinas, ela ainda encontra bastante resistência no círculo linguístico: frases do tipo “nunca precisamos de algo desse tipo antes” ou “as coisas realmente interessantes são de caráter qualitativo e não precisam de nenhuma avaliação quantitativa” ou “eu sou um linguista de campo e não preciso dessas coisas” são ainda muito frequentes.

Vamos dizer bem diretamente: tais frases não são particularmente razoáveis. Em relação à primeira frase, por exemplo, não está claro, até hoje, que não precisamos de métodos quantitativos – para provar esse ponto, alguém teria que mostrar que métodos quantitativos não contribuíram em nada para pesquisas anteriores, um ponto de vista ridículo. – e mesmo assim, não está claro que o campo da linguística não está agora em um ponto em que tais métodos são importantes. A segunda afirmação, por exemplo, na prática métodos quantitativos e qualitativos andam juntos: considerações de caráter qualitativo advêm de resultados de pesquisas quantitativas. Trabalhar quantitativamente não quer dizer apenas escrever resultados em termos numéricos – é óbvio que é necessário que haja uma discussão qualitativa das implicações dos achados quantitativos – no entanto, o que veremos a seguir é que a pesquisa quantitativa nos possibilita identificar o que realmente merece uma discussão qualitativa. Quanto à última frase: até mesmo um linguista descritivo (de campo) que trabalha com documentação de línguas quase extintas pode se beneficiar de métodos quantitativos. Se o capítulo sobre tempo verbal discute se a escolha de um tempo verbal está ou não correlacionada com o discurso indireto, um estudo quantitativo poderia mostrar se tal correlação de fato existe. Se um estudo sobre a voz média da língua atabascana tanaina (Athabaskan Dena’ina) tenta indentificar como a sintaxe e a semântica estão relacionadas aos marcadores de voz média, métodos quantitativos podem revelar resultados bem interessantes (cf. [BEREZ; GRIES, 2010](#)).

Os dois últimos pontos levam a dois argumentos de caráter mais geral: geralmente, apenas métodos quantitativos conseguem separar o joio do trigo. Vamos supor que um linguista queira testar a famosa hipótese aspectual que afirma que os aspectos imperfectivo e perfectivo são preferencialmente utilizados nos tempos presente e passado, respectivamente (cf. [SHIRAI; ANDERSEN, 1995](#)). A rigor, um linguista teria que testar todos os verbos em todas as línguas (a chamada população). É óbvio que isso não é possível, então um linguista estuda apenas uma amostra de frases para investigar a estrutura morfológica dos verbos. Vamos ainda supor que um linguista investigou uma amostra pequena contendo 39 orações de uma determinada língua e obteve os resultados mostrados na Tabela 1.

Tabela 1 – A distribuição fictícia de tempo e aspecto de um pequeno corpus

	Imperfectivo	Perfectivo	Totais
Tempo Presente	12	6	18
Tempo Passado	7	13	20
Totais	19	19	38

Esses dados parecem confirmar a hipótese aspectual: existem mais orações no presente com aspecto imperfectivo e mais orações no passado com o aspecto perfectivo. No entanto, o chamado teste Qui-Quadrado, que pode ser utilizado para dados dessa natureza, mostra que essa distribuição, na verdade, poderia ter ocorrido por acaso, com uma probabilidade p que excede a linha dos 5% geralmente utilizadas em estudos quantitativos. Dessa forma, o linguista não deveria aceitar a hipótese aspectual para essa população específica com base nessa amostra. O ponto principal é que apenas uma inspeção na tabela anterior é insuficiente – é necessário um teste estatístico para evitar que o linguista faça generalizações inválidas.

Um exemplo mais convincente é discutido por [Crawley \(2007, p. 314\)](#). Vamos supor que um estudo tenha mostrado que duas variáveis x e y estão correlacionadas de forma que quanto maior o valor de x , maior o valor de y ; cf. Figura 1.

Note, no entanto, que os dados, na verdade, contêm informação sobre uma terceira variável (com sete níveis – a até g) da qual dependem x e y . Interessantemente, se você agora inspecionar qual a relação entre x e y para cada um dos sete níveis da terceira variável, você irá notar que a relação se torna “quanto maior x ”, menor será y ; cf. Figura 2, onde os sete níveis estão indicados pelas letras. Tais padrões em dados são fáceis de passarem despercebidos – eles só poderão ser identificados através de um estudo quantitativo cuidadoso, que faz com que o conhecimento de procedimentos estatísticos seja indispensável.

Para estudantes de linguística – ao contrário de linguistas experientes – existe também uma questão prática a se considerar. Mais cedo ou mais tarde, você precisará escrever a sua dissertação ou tese. Métodos quantitativos podem ser de grande valia para evitar armadilhas subjacentes aos dados da Tabela 1 e da Figura 1. Em vista disso tudo, esperamos que seja óbvio agora que métodos quantitativos têm muito a oferecer. Esperamos que esse livro ofereça a você um conhecimento básico sólido e prático.

Esse argumento tem um aspecto adicional. Ao contrário de áreas como Crítica Literária, a Linguística é uma Ciência Empírica. Assim, é necessário – principalmente para estudantes – conhecer os métodos e pressupostos básicos de pesquisa empírica e estatística a fim de ser capaz de compreender tanto a argumentação científica em geral quanto a argumentação linguística em particular. Esse conhecimento se faz especialmente relevante em domínios tais como a linguística de corpora e psicolinguística, onde os dados são analisados com altos graus de sofisticação que um conhecimento básico da terminologia relevante é necessário. Sem o treinamento adequado, o que você entenderia de afirmações tais como “a interação entre o tamanho do objeto e o tamanho do referente não alcançou o nível padrão de significância: $F_{1,12} = 2,18$; $p = 0,166$; $\eta^2_{\text{parcial}} = 0,154$ ”? Como saber de cabeça se o fato de que, em média, um estudante de língua estrangeira do sexo feminino produz sentenças com duas palavras a mais que estudantes de sexo masculino é algo relevante ou é apenas fruto do acaso? Tais dados precisam de análise estatística séria.

1.3 O desenho e a lógica por trás de estudos quantitativos

Nesta seção, veremos detalhadamente, o desenho e a lógica que subjaz estudos de natureza quantitativa. Serão mostradas as várias fases de estudos quantitativos, a sua estrutura e discutir o raciocínio empregado neles. O documento no qual você descreve a sua pesquisa quantitativa deve ter quatro partes: introdução, metodologia, resultados e discussão – caso você discuta mais de um estudo no mesmo documento, cada estudo deverá ter a sua seção de metodologia, resultados e discussão. No final, você deve apresentar uma discussão geral. Com poucas exceções, toda a discussão apresentada nessa seção utilizará um exemplo linguístico. O exemplo é baseado no fenômeno da língua inglesa conhecido como “particle placement”,

Figura 1 – Correlação entre duas variáveis x e y

Figura 2 – Correlação entre duas variáveis x e y, controladas por uma terceira variável fictícia

ou seja, a alternância da ordem dos componentes de um verbo preposicional transitivo. Veja o exemplo (1).

Exemplo (1)

- a He picked up [SN the book].
Construção: VPO (verbo - partícula - objeto)
- b He picked [SN the book] up.
Construção: VOP (verbo - objeto - partícula)

Um aspecto interessante dessa alternância é que, na maioria das vezes, as duas construções parecem ser sinônimas e falantes nativos de inglês não conseguem explicar porque produzem (1a) em certas ocasiões e (1b) em outras ocasiões. Nas últimas décadas, vários linguistas têm tentado descrever, explicar e prever essa alternância (cf. [GRIES, 2003a](#) e [2003b](#)). Nessa seção, iremos utilizar esse fenômeno para ilustrar a estrutura de um estudo quantitativo.

1.3.1 *Scouting*

No começo de seu estudo, você precisa ter uma visão panorâmica dos trabalhos anteriores que tratam do mesmo fenômeno que você está interessado em investigar. Assim, você terá uma noção do que ainda pode e precisa ser feito. É nessa fase que você procura conhecer sobre teorias existentes que podem ser empiricamente testadas ou, o que é menos frequente, você entra em um território ainda inexplorado em que você será o primeiro a desenvolver uma nova teoria. A seguir, uma lista das atividades que geralmente são desenvolvidas nessa fase:

- uma primeira (talvez informal) caracterização do fenômeno;
- estudar a literatura relevante;
- observar a ocorrência natural do fenômeno para auxiliar nas primeiras generalizações indutivas;
- coletar informações adicionais (i.e., consultar colegas, estudantes, etc.);
- raciocínio dedutivo da sua parte.

Se você der uma rápida olhada no fenômeno de colocação de partícula, você irá notar que existe uma gama de variáveis que influenciam a escolha entre as duas construções. Uma variável é definida como um símbolo para um conjunto de estados, ou seja, um símbolo que, diferentemente de uma constante, pode exibir pelo menos dois estados ou níveis diferentes. (cf. [BORTZ; DÖRING, 1995](#), p. 6 ou cf. [BORTZ, 2005](#), p. 6) ou, mais intuitivamente, um símbolo com propriedades descritivas (cf. [JOHNSON, 2008](#), p. 4) ou como medidas de um item que podem ser numéricos ou categóricos (Evert, p.c.). As variáveis que podem influenciar a colocação das partículas são:⁵

- Complexidade: o objeto direto é um objeto direto simples (ex.: “the book”), um objeto direto com um sintagma modificador (ex.: “the brown book ou the book on the table”) ou um objeto direto com uma oração modificadora (ex.: “the book I had bought in Europe”) (e.g., cf. [FRASER, 1966](#));
- Tamanho: o tamanho do objeto direto (cf., e.g., ([CHEN, 1986](#)), cf. [HAWKINS, 1994](#)), que pode ser mensurada em sílabas, palavras, ...;
- Objeto direcional: a presença ou ausência de um sintagma preposicional (SP) direcional a direcional após o verbo frasal transitivo (ex.: He picked the book up from the table) (cf. [CHEN, 1986](#));

⁵ As variáveis são apresentadas em caixa alta simples e seus níveis em caixa alta em itálico.

- Animacy: se o referente do objeto direto é inanimado como em “He picked up the book” or animado como em “He picked his dad up” (cf. [GRIES, 2003a](#): Capítulo 2);
- Concretude: se o referente do objeto direto é abstrato como em “He brought back peace’ to the region” ou concreto como em “He picked his dad up at the station” (cf. [GRIES, 2003a](#): Capítulo 2);
- Tipo: a classe gramatical do núcleo do objeto direto: um pronome (ex.: “He picked him up this morning”), um semipronome (ex.: “He picked something up from the floor”), um substantivo lexical (ex.: “He picked people up this morning”) ou um nome próprio (e.g., “He picked Peter up this morning”) (cf. [Van DONGEN, 1919](#)).

Durante essa fase inicial, é bom representar os resultados de maneira esquemática, tabular. Uma tabela útil para tal resumo, representa quais estudos (em colunas) discutiram quais variáveis (nas linhas). A Tabela 2 seria uma boa representação das variáveis apresentadas acima. Ela permite que se reconheça imediatamente (i) quais variáveis já foram investigadas nos estudos e (ii) os estudos que investigaram o maior número de variáveis. Uma outra tabela útil mostra os níveis das variáveis e suas preferências por uma das duas construções. A Tabela 3 é um exemplo de tal representação. Você visualiza imediatamente que, para algumas variáveis, apenas um nível está associado com uma preferência construcional.

Tabela 2 – Resumo esquemático dos fenômenos colocação da partícula I

	Fraser (1966)	Chen (1986)	Hawkins (1994)	Gries (2003a)	Van Dongen (1919)
Complexidade	×				
Tamanho		×	×		
SP Direcional		×			
Animacidade				×	
Concretude				×	
Tipo					×

Tabela 3 – Resumo esquemático dos fenômenos colocação da partícula II

	Níveis que preferem Construção: vpo	Níveis que preferem Construção: vop
Complexidade		
Tamanho	longo	
SP Direcional	ausência	presença
Animacidade	inanimado	animado
Concretude	abstrato	concreto
Tipo		pronominal

A Tabela 3 já sugere que a Construção: VPO é utilizada com objetos diretos mais cognitivamente complexos: sintagmas nominais longos e complexos com substantivos lexicais se referindo a entidades abstratas. A Construção: VOP por outro lado é utilizada com as preferências opostas. É claro que para um estudo de verdade, essas impressões devem ser formuladas de maneira mais precisa. Além disso, você deve compilar uma lista de outros fatores que podem influenciar tanto a preferência construcional, como a amostragem de sentenças, ou até mesmo os participantes. A maior parte dessas informações seriam explicadas e discutidas na primeira seção do seu estudo empírico: a introdução.

1.3.2 Hipóteses e operacionalização

Uma vez que você já possui uma visão panorâmica do fenômeno de interesse e decidiu fazer um estudo empírico sobre ele, você deve formular hipóteses. O que isso quer dizer e como fazer? Para abordar essa questão, precisamos saber o que são hipóteses e que tipos de hipóteses existem.

1.3.2.1 Hipótese científica em formato textual

De acordo com [Bortz e Döring \(1995, p. 7\)](#) nós vamos considerar uma hipótese como sendo uma declarativa que seguem os seguintes critérios:

- ser uma declarativa de caráter geral que aborda mais que apenas um simples evento;
- ser uma declarativa que, mesmo que implicitamente, tem a estrutura de uma oração condicional (se... então...);
- ser potencialmente falsificável, ou seja, que seja possível pensar em eventos ou situações que contradizem a declarativa. Em muitos casos, isso implica que o cenário descrito na oração condicional seja algo testável. No entanto, essas duas características não são idênticas. Existem declarativas que são falsificáveis, mas não testáveis, por exemplo: “Se uma criança cresce sem nenhum insumo linguístico, então ela irá falar Latim”. Essa declaração é falsificável, mas obviamente, por razões éticas, não pode ser testada (cf. [STEINBERG, 1993: Seção 3.1](#)).

A declaração a seguir é uma hipótese científica de acordo com o critério acima: “A redução da idade mínima para a obtenção da carteira de motorista de 18 para 17 anos em países europeus irá dobrar o número de acidentes de trânsito nesses países”. Essa declaração é uma declaração geral que não está restrita a apenas um ponto no tempo ou apenas um país. Além disso, essa declaração pode ser parafraseada em forma de uma oração condicional: “Se alguém reduz a idade mínima... então o número de acidentes irá dobrar”. E por último, essa declaração é falsificável pois é concebível – na verdade é bem provável – que se alguém reduz a idade mínima, o número de acidentes não irá dobrar. Assim, a declaração seguinte não se caracteriza como uma hipótese científica: “A redução da idade mínima para a obtenção da carteira de motorista de 18 para 17 anos em países europeus pode dobrar o número de acidentes de trânsito nesses países”. Essa declaração é testável, pois a idade mínima pode ser reduzida, porém não é falsificável, uma vez que a palavra “pode” basicamente quer dizer que “pode ser como pode não ser”: a declaração será verdadeira se o número de acidentes de trânsito dobrar, mas será também verdadeira se não dobrar. Em outras palavras, qualquer que seja o resultado observado após a redução da idade mínima, a declaração será verdadeira.

Com relação ao fenômeno de colocação de partícula, as declarações seguintes são exemplos de hipóteses científicas:

- se o objeto direto de um verbo preposicional transitivo é sintaticamente complexo, então falantes nativos irão produzir a Construção: VPO com mais frequência do que quando o objeto direto for sintaticamente simples;
- se o objeto direto de um verbo preposicional transitivo é longo, então falantes nativos irão produzir a Construção: VPO com mais frequência do que quando o objeto direto for curto;
- se a construção verbo-preposição for acompanhada por um SP direcional, então falantes nativos irão produzir a Construção: VOP com mais frequência do que quando não há a presença de um SP direcional (e semelhante para todas as outras variáveis da Tabela 3).

Quando você formula uma hipótese, é importante que as noções utilizadas na hipótese sejam precisamente formuladas. Por exemplo, se uma teoria linguística utiliza noções tais como complexidade cognitiva ou disponibilidade discursiva, ou até mesmo algo direto e óbvio como tamanho do constituinte, é necessário que a teoria seja capaz de definir o que cada um desses termos realmente quer dizer; na seção 1.3.2.2 vamos ver essa questão mais detalhadamente.

Podemos distinguir dois tipos de hipóteses. A primeira, sobre a qual temos falado até agora, consiste de duas partes: a parte “se” (VI) e a parte “então” (VD). VI representa a Variável Independente, a variável na parte “se” da hipótese que geralmente, mas não necessariamente, causa as mudanças na parte “então” da hipótese. O VD, por sua vez, significa Variável Dependente, a variável na parte “então” da hipótese e cujos valores, variação ou distribuição, tentamos explicar. Além disso, é útil mencionar também para

propósitos futuros as variáveis de confundimento (“confounding variables”) e as variáveis moderadoras (“moderator variables”). As primeiras podem ser definidas como variáveis que estão correlacionadas com variáveis dependentes e independentes, e as últimas como variáveis (muitas vezes estranhas à concepção inicial de um estudo) que influenciam/moderam a relação entre a(s) variável(s) independente(s) e dependente(s).

Figura 3 – Diferentes tipos de variáveis

Com essa terminologia, nós podemos agora reescrever as hipóteses acima. Na primeira, a VI é a complexidade sintática do objeto direto (Complexidade com três níveis simples, com sintagma modificador, e com oração modificadora), e a VD é a escolha entre as duas construções (Construção com dois níveis VPO e VOP). Na segunda hipótese, a VI é o tamanho do objeto direto (Tamanho com valores entre 1 e x), e a VD é a mesma da hipótese anterior, ou seja, a escolha entre as duas construções (Construção com dois níveis VPO e VOP), etc.

O segundo tipo de hipótese contém apenas uma variável dependente, não possui uma variável independente que explica o comportamento da variável dependente. Nesse caso, a hipótese é apenas uma declaração sobre como são os valores, a variação ou a distribuição da variável dependente. Exemplos frequentes postulam distribuições iguais (por exemplo, frequências) ou formas particulares de distribuição (por exemplo, curvas normais em forma de sino):

- As duas construções ou, mais tecnicamente falando, os dois níveis da CONSTRUÇÃO (VPO e VOP) não são igualmente frequentes; Note novamente como uma variável independente não é mencionada.
- As extensões dos objetos diretos não são distribuídas de forma normal.

No que se segue, iremos lidar com essas duas hipóteses (com um viés para a primeira).

Assim, podemos também definir uma hipótese científica como sendo uma afirmação sobre a relação entre duas ou mais variáveis, ou, para o segundo tipo, como uma afirmação sobre uma variável em um contexto de amostragem, que se espera também ser verdadeira em contextos semelhantes e/ou para objetos semelhantes na população. Deste modo, uma vez que as variáveis potencialmente relevantes para a investigação foram identificadas, você deve formular uma hipótese relacionando as variáveis relevantes na oração condicional apropriada ou em uma paráfrase da mesma.

Depois que sua hipótese foi formulada conforme apresentado até agora, você também deve definir – antes de coletar seus dados! – quais situações ou eventos tornariam a sua hipótese falsa. Portanto, além da sua própria hipótese – também conhecida como hipótese alternativa H_1 – você também deve formular uma outra hipótese – conhecida como hipótese nula H_0 – que é o oposto lógico da sua H_1 . Para a primeira das três hipóteses apresentadas anteriormente envolvendo tanto uma variável dependente quanto uma variável independente, uma versão em formato de texto da H_0 seria da seguinte forma:

- $H_{0\text{tipo 1}}$: Se o objeto direto de um verbo preposicional transitivo é sintaticamente complexo, então falantes nativos não irão produzir a Construção VPO com mais frequência do que quando o objeto direto for sintaticamente simples.

Para a primeira das duas hipóteses acima envolvendo apenas uma variável dependente, a H_0 ficaria assim:

- $H_{0\text{tipo 1}}$: As duas construções ou, mais tecnicamente falando, os dois níveis da CONSTRUÇÃO (VPO e VOP) não são igualmente frequentes.

É crucial formular a H_0 como mencionado acima, basicamente inserindo um “não”. A idéia é que ambas as hipóteses - H_0 e H_1 - abrangem todos os resultados teoricamente possíveis. Assim, se a sua H_1 fosse “Objetos complexos levam mais à CONSTRUÇÃO: VPO do que à CONSTRUÇÃO: VOP”, então a sua H_0 não deve ser “Objetos complexos levam menos à CONSTRUÇÃO: VPO do que à CONSTRUÇÃO: VOP”, porque essas duas hipóteses não abrangem todos os resultados possíveis - elas não incluem o caso no qual as duas construções são igualmente frequentes.

Na grande maioria dos casos, o primeiro tipo de hipótese nula afirma que não há diferença entre (dois ou mais) grupos ou nenhuma relação entre a(s) variável(eis) independente(s) e a(s) dependente(s), e que qualquer diferença ou efeito obtido é fruto do acaso ou variação aleatória. O segundo tipo de H_0 tipicamente afirma que a variável dependente é distribuída de maneira randômica ou é distribuída de acordo com alguma distribuição matematicamente conhecida, como a distribuição normal. Entretanto, uma complicação adicional é que você tem que distinguir dois tipos de H_1 s: a hipótese alternativa direcional não apenas prediz que há algum tipo de efeito, diferença ou relação, mas também prevê a direção do efeito - note a expressão “com mais frequência” na hipótese 0 tipo 1 acima relacionando CONSTRUÇÃO e COMPLEXIDADE. Por outro lado, a hipótese alternativa não-direcional prediz apenas que há algum tipo de efeito, diferença ou relação sem especificar a direção do efeito. Uma hipótese não-direcional para o exemplo tipo 1 acima seria, portanto, assim:

- $H_{1do\ tipo\ 1\ não-direcional}$: Se o objeto direto de um verbo frasal (*phrasal verb*) for sintaticamente complexo, então falantes nativos irão produzir a ordem VPO, *diferentemente do que frequentemente acontece quando o objeto direto é sintaticamente simples*.

Assim, H_0 afirma que não há nenhuma correlação entre a complexidade sintática do objeto direto e a escolha da construção. Se por um acaso você encontrar alguma correlação na amostra que coletar, essa correlação foi resultado do acaso. Ambas H_1 's afirmam que há uma correlação – assim, você deve encontrar essa correlação na sua amostra. Ambas hipóteses devem ser formuladas antes da coleta de dados, de forma que não se apresente qualquer resultado como sendo o resultado previsto. Obviamente, tudo isso deve ser discutido de maneira completa e acurada na introdução do seu trabalho ou, talvez, no início da seção de metodologia.

1.3.2.2 Operacionalização de suas variáveis

A formulação de suas hipóteses em formato textual não é a última tarefa nessa fase inicial do seu estudo empírico. A maneira como as variáveis mencionadas nas suas hipóteses serão investigadas ainda não está claro. Por exemplo, assim como mencionado anteriormente, a noção de complexidade cognitiva pode ser formulada de várias maneiras distintas. Até mesmo a noção de tamanho do constituinte nem sempre é tão óbvia o quando parece: o tamanho do objeto direto pode ser medido em letras, fonemas, sílabas, morfemas, palavras, nódulos sintáticos, etc. Logo, você deve encontrar uma maneira de operacionalizar as variáveis mencionadas em sua hipótese. Em outras palavras, você precisa apontar o que será observado, contado e mensurado quando investigar as suas variáveis.

Por exemplo, imagine que você queira operacionalizar o CONHECIMENTO DE LÍNGUA ESTRANGEIRA que uma pessoa possui. Neste caso, você poderia medir:

- COMPLEXIDADE DAS FRASES que uma pessoa consegue formar na língua-alvo através de um teste (apenas orações simples? também orações compostas? também frases complexas? quantos de cada?);
- TEMPO em segundos existente entre dois erros durante uma conversação;
- NÚMERO DE ERROS A CADA 100 PALAVRAS existente em um texto que a pessoa redigiu em 90 minutos.

Qual seria, em sua opinião, o problema com as duas propostas de operacionalização apresentadas a seguir?

- QUANTIDADE DE VOCABULÁRIO ATIVO;
- QUANTIDADE DE VOCABULÁRIO PASSIVO.

Pausa para pensar

Essas duas propostas não são particularmente úteis. Apesar de serem um bom índice para medir o conhecimento que alguém possui de uma língua estrangeira, elas não são diretamente observáveis: não está claro o que seria necessariamente contado ou medido. No entanto, se você operacionalizasse a quantidade de vocabulário passivo como sendo o número de palavras que uma pessoa acerta em um teste de vocabulário (um teste que envolva palavras com frequência variada) ou em um teste de sinônimos, aí sim você saberia o que contar/mensurar. Da maneira como foram apresentadas acima, as propostas são muito vagas.

Você já deve ter notado que operacionalizar variáveis envolve a utilização de números que representem as variáveis. Tais números podem ser medidos (402 ms de tempo de reação, 12 palavras em um teste de sinônimo, ou um objeto direto de 4 sílabas), ou números que representem (codifiquem) estados ou fatores não numéricos. Dessa forma, uma variável não é definida apenas pelo papel que desempenha na hipótese – dependente versus independente – mas também pelo seu nível de mensuração:

- *variáveis nominais* ou *categóricas* são variáveis que identificam um grupo ou uma categoria. Valores distintos apenas revelam que os sujeitos que possuem esses valores distintos apresentam características distintas. Tais variáveis são conhecidas como *variáveis nominais* ou *binárias* quando podem assumir apenas dois valores ou níveis distintos; são conhecidas como variáveis categóricas quando podem assumir mais de dois valores ou níveis distintos. Utilizando nosso exemplo sobre *particle placement*, a variável SPDIRECIONAL poderia ser codificada utilizando-se o número 1 para AUSÊNCIA e 2 para PRESENÇA. No entanto, é importante notar que o fato de que o valor para PRESENÇA é duas vezes maior que o valor para AUSÊNCIA não significa nada matematicamente (apenas que os valores são diferentes). Em termos teóricos, você poderia codificar AUSÊNCIA com o número 34.2 e PRESENÇA com o número 7⁶. Alguns exemplos típicos de variáveis nominais ou categorical são ANIMACIDADE (ANIMADO vs. INANIMADO), CONCRETUDE (CONCRETO vs. ABSTRATO), TONICIDADE (TÔNICA vs. FRACA), ASPECTO (ATIVIDADE vs. ACCOMPLISHMENT vs. ACHIEVEMENT vs. ESTADO) etc.
- *variáveis ordinais* não apenas distinguem objetos como pertencentes a categorias distintas (assim como faz uma variável categórica), como também permitem que os objetos sejam ordenados de maneira significativa. No entanto, as diferenças entre as ordens não podem ser comparadas matematicamente. Conceitos ou notas são exemplos típicos de variáveis ordinais: um aluno com A (4 pontos) tirou uma nota melhor que um aluno com C (2 pontos), mas o fato de 4 ser duas vezes maior que 2 não quer dizer que o aluno com A é duas vezes melhor que o aluno com C. A depender do sistema de pontuação, o aluno com A pode ser acertado até três vezes mais que o aluno com C. No nosso exemplo linguístico, a variável COMPLEXIDADE é uma variável ordinal se você operacionalizada da seguinte forma: SN SIMPLES (1) vs. SINTAGMA MODIFICADOR (2) vs. ORAÇÃO MODIFICADORA (3). É importante que a ordem seja compatível com a variável: se a variável é chamada COMPLEXIDADE SINTÁTICA, então a ordem dos números deve crescer de acordo com o aumento da complexidade, ou seja, os números maiores devem representar objetos diretos mais complexos. Se, por outro lado, a variável é chamada SIMPLICIDADE SINTÁTICA, os números maiores devem representar maiores graus de simplicidade, ou seja, objetos direto mais simples. Outros

⁶ É comum codificar variáveis binárias utilizando 0 e 1, por dois motivos: (i) conceptualmente, variáveis dessa natureza são entendidas como a presença de alguma coisa (= 1) ou a ausência dessa mesma coisa (= 0), ou até mesmo como uma variável (cf. abaixo); i.e., no exemplo sobre colocação de partículas, a variável binária CONCRETUDE pode ser entendida como uma variável proporcional NÚMERO DE REFERENTES CONCRETOS; (ii) por razões que não iremos discutir aqui. A utilização de 0 e 1 pode ser útil computacionalmente. Alguns software estatísticos até mesmo exigem que variáveis binárias sejam codificadas dessa forma.

exemplos típicos de variáveis ordinais são NÍVEL SÓCIO-ECONÔMICO or GRAUS DE IDIOMATICIDADE or DIFICULDADE DE VOCABULÁRIO (e.g., BAIXO/1 vs. MÉDIO/2 vs. ALTO/3).

- *Variável de Razão* não apenas distingue objetos como sendo membros de categorias distintas, mas também permitem que os valores das diferentes ordens sejam comparados. Por exemplo, TAMANHO EM SÍLABAS é uma variável proporcional: quando um objeto tem seis sílabas e um outro tem três, o primeiro é diferente do segundo (informação nominal), o primeiro é maior que o segundo (informação ordinal) e o primeiro é exatamente duas vezes maior que o segundo. Outros exemplos típicos de variáveis proporcionais são salários e tempo de reação medidos em segundos.⁷

Essas diferenças podem ser claramente ilustradas em uma tabela. A Tabela 4 apresenta dados fictícios sobre o tamanho e grau de complexidade de sujeitos e objetos – quais colunas contêm que tipo de variável?

Tabela 4 – Dados fictícios sobre o tamanho e complexidade de sujeitos e objetos

Dado	Complexidade	Fonte do dado	Tamanho da sílaba	Papel Sintático
1	alta	d8y	6	objeto
2	alta	hhv	8	sujeito
3	baixa	kb0	3	sujeito
4	média	kb2	4	objeto

Pausa para pensar

DADO é uma variável categórica: cada dado recebe seu próprio número que o identifica como sendo único e tem pouca significância além disso. COMPLEXIDADE é uma variável ordinal com três níveis. FONTE DO DADO é outra variável categórica: os níveis dessa variável são os nomes dos arquivos do Corpus Nacional Britânico (BNC - *British National Corpus*). Tamanho da sílaba é uma variável de razão uma vez que o terceiro objeto pode ser descrito como sendo a metade do tamanho do primeiro. PAPEL SINTÁTICO é uma variável nominal/categórica. Essas distinções são muito importantes uma vez que esses níveis de mensuração determinam que tipo de testes estatísticos se aplicam ou não para um determinado conjunto de dados, como veremos abaixo. Para adiantar uma regra básica, é usualmente melhor trabalhar com o nível mais alto de mensuração, e eu voltarei a tocar nesse ponto em breve.

A questão da operacionalização é uma das mais importantes do processo. Se você não operacionalizar corretamente as suas variáveis, o estudo como um todo pode ser inválido, uma vez que você pode acabar não medindo o que desejava medir. Sem uma operacionalização adequada, a validade do seu estudo é posta em cheque. Vamos retornar a um exemplo anterior. Imagine que estamos investigando se os sujeitos em Inglês são maiores (mais longos) que os objetos. Se pesquisarmos as frases de um corpus, podemos encontrar a seguinte frase:

Exemplo (2)

a [*Sujeito* The younger bachelors] ate [*Objeto* the nice little parrot].

O resultado para essa frase vai depender de como a variável Tamanho é operacionalizada. Se TAMANHO for operacionalizado como o NÚMERO DE MORFEMAS, então o sujeito da frase é mais longo que o objeto direto: o sujeito obtém o valor 5 (*The, young, a* partícula comparativa *-er, bachelor*, e o marcador de plural *-s*) e o objeto direto obtém o valor 4 (*the, nice, little, parrot*). No entanto, se a variável TAMANHO for operacionalizada como o *número de palavras*, o sujeito (3 palavras) será menor que o objeto direto (4 palavras). Ainda, se TAMANHO for operacionalizado como o número de caracteres sem espaço, o sujeito e o objeto direto terão o mesmo tamanho (19 caracteres). Assim, só nesse exemplo, a operacionalização determinou o resultado.

⁷ Existe ainda um outro tipo de variável conhecida como variável intervalar que não iremos discutir aqui.

1.3.2.3 Hipóteses científicas em formato estatístico/matemático

Uma vez que você formulou sua hipótese alternativa, a hipótese nula e definiu como as variáveis serão operacionalizadas, você deve formular duas versões estatísticas das suas hipóteses. Em outras palavras, você primeiro formula as hipóteses em formato textual, então, na sua hipótese estatística, você expressa os resultados numéricos que espera encontrar com base nas suas hipóteses textuais. Tais resultados numéricos usualmente envolvem uma entre cinco fórmulas matemáticas:

- Frequências
- Médias
- Dispersões
- Correlações
- Distribuições

Vamos começar mostrando um exemplo simples de uma H_1 sobre colocação de partícula: se uma construção do tipo verbo-preposição é acompanhada de um SP direcional, então um falante nativo irá produzir os constituintes na ordem VOP mais frequentemente do que se não houver o SP direcional. Para formular a hipótese estatística correspondente, você deve responder à seguinte pergunta: se eu investigar 200 frases contendo a construção do tipo verbo-preposição, como vou saber se minha hipótese está correta ou não? (Na verdade, o procedimento é um pouco diferente, mas vamos discutir isso mais adiante). Uma possibilidade é contar o número de Construção: VPO e Construção: VOP que são acompanhadas por um SP direcional. Se houver mais SP direcionais após Construção: VOP do que após Construção: VPO, então teremos suporte para nossa hipótese. Assim, essa possibilidade envolve frequências e as hipóteses estatísticas seriam:

- H_1 direcional: $n_{\text{SP-direcional após Construção: VPO}} < n_{\text{SP-direcional após Construção: VOP}}$
- H_1 não-direcional: $n_{\text{SP-direcional após Construção: VPO}} \neq n_{\text{SP-direcional após Construção: VOP}}$
- H_0 : $n_{\text{SP-direcional após Construção: VPO}} = n_{\text{SP-direcional após Construção: VOP}}$ ⁸

A propósito: o que essas hipóteses estatísticas pressupõem?

Pausa para pensar

Elas pressupõem que você irá investigar o mesmo número de ocorrências de ambas as construções, pois, de outra forma, uma pequena frequência observada de SPs direcionais após a construção VOP - a frequência que esperamos ser maior - poderia ser simplesmente resultado de uma frequência geral menor da construção VOP. Para a variável COMPLEXIDADE, você poderia formular hipóteses semelhantes baseadas em frequências, caso a COMPLEXIDADE seja operacionalizada com base, por exemplo, no três níveis mencionados acima.

Vamos agora ver um exemplo que envolve hipótese estatística com base em médias: se o objeto direto de um verbo preposicional transitivo for longo, então falantes nativos irão produzir a construção na ordem VPO com mais frequência, comparados com quando o objeto direto for curto. Para verificar isso, podemos medir o tamanho médio dos objetos diretos na Construção: VPO e na Construção: VOP e então comparar essas médias. Você pode então escrever:

⁸ Nota: Foi dito acima que a hipótese nula é frequentemente obtida inserindo “não” na hipótese alternativa. Assim, quando a versão estatística da hipótese alternativa envolve um “<”, você pode esperar que a versão estatística da hipótese nula contenha um “≥”. Porém, nós seguiremos a convenção usual, também mencionada acima, de que uma hipótese nula afirma a ausência de diferença/efeito/correlação etc., e é por isso que escrevemos “=”. Você verá abaixo que os casos resolvidos com “≥” serão ainda invocados nos cálculos que são baseados nessas hipóteses estatísticas.

H_1 direcional: $\text{média}_{\text{Tamanho do OD na Construção: VPO}} > \text{média}_{\text{Tamanho do OD na Construção: VOP}}$

H_1 não-direcional: $\text{média}_{\text{Tamanho do OD na Construção: VPO}} \neq \text{média}_{\text{Tamanho do OD na Construção: VOP}}$

H_0 : $\text{média}_{\text{Tamanho do OD na Construção: VPO}} = \text{média}_{\text{Tamanho do OD na Construção: VOP}}$

Com operacionalizações semelhantemente óbvias, as outras hipóteses textuais acima podem ser transformadas em hipóteses estatísticas. Agora sim sabemos o que precisa ser observado para que possamos rejeitar a hipótese nula. (Vamos lidar com hipóteses que envolvem correlações, variabilidade e distribuições mais tarde).

Todas as hipóteses discutidas até agora envolvem casos simples em que uma amostra com construções do tipo verbo-partícula é avaliada com relação à diferença entre elas no que diz respeito a uma variável independente (por exemplo: SPDirecional). Os métodos estatísticos apropriados para lidar com casos desse tipo serão abordados no Capítulo 4. No entanto, nem sempre as coisas são simples assim: muitos dos fenômenos que investigamos são multifatoriais por natureza, ou seja, as variáveis dependentes são usualmente influenciadas por, ou pelo menos relacionadas a, mais de uma variável independente. Embora a lógica geral seja a mesma que a apresentada acima, surgem algumas complicações que serão discutidas no capítulo 5.

1.3.3 Coleta e armazenamento de dados

Apenas depois de definir operacionalmente todas as variáveis e definir todas as hipóteses é que você parte para a coleta de dados. Por exemplo, você faz seu experimento ou faz sua análise de corpora. No entanto, raramente você conseguirá estudar toda a população de interesse, mas terá que estudar uma amostra dessa população. Você deve então se certificar de que a amostra represente acuradamente a população de interesse. Quando uma amostra represente bem todos os seguimentos de uma população, dizemos que essa amostra é *representativa*. Quando a amostra representa bem proporcionalmente as partes da população, dizemos que essa amostra é *balanceada*. Imagine que queremos investigar a frequência e usos do marcador discursivo, *like*, na fala de adolescentes da Califórnia. Para tal, você deve compilar um corpus da fala de adolescentes da Califórnia, gravando algumas conversas de alguns adolescentes. Para obter uma amostra representativa e balanceada, a proporção dos diferentes tipos de conversas que adolescentes geralmente se engajam deve ser bem aproximadamente as mesmas na população e na amostra. Por exemplo, uma boa amostra não incluiria somente conversar entre adolescentes, mas também conversas com outras pessoas, tais como pais, professores, etc. Além disso, se possível, é importante que as proporções desses tipos diferentes de conversa sejam as mesmas na amostra e na população.

Sabemos que é importante tentar seguir essas regras o máximo possível, mas por que muitas dessas regras são consideradas um ideal teórico?

Pausa para pensar

Elas são consideradas apenas um ideal teórico porque não conhecemos todas as partes e suas proporções na população. Quem de nós saberia dizer o quanto do discurso de um típico adolescente californiano - e, a propósito, o que seria um adolescente típico californiano? - ocorre com outros adolescentes, com seus pais, com seus professores, etc. E outra coisa: como medir essa proporção? Em palavras? Sílabas? Frases? Minutos? Mesmo assim, apesar de ter que pensar nessas questões em termos de estimativas, pensar nessas questões é tão importante quanto operacionalizar suas variáveis. Se você não fizer isso, o seu estudo todo pode ficar comprometido, uma vez que você não será capaz de fazer as generalizações apropriadas dos resultados que encontrar na análise da sua amostra. Uma regra importante para escolher os membros da sua amostra é *aleatorização*. Por exemplo: imagine que você dê aos adolescentes que participarem do estudo um gravador com uma pequena lâmpada e a instrução que toda vez que a luz acender, eles devem gravar suas conversas. Você pode, então, enviar um sinal para o gravador (em que a luz irá acender) em intervalos aleatórios (determinados por um computador, por exemplo). Este

procedimento evitaria a probabilidade de se ter uma amostra enviesada e consequentemente você teria uma amostra mais representativa da população.

Vamos analisar um exemplo semelhante do campo da aquisição de primeira língua. Pesquisas mostram que o número de perguntas em gravações de conversas entre a criança e o responsável é grande. Alguns pesquisadores suspeitam que a razão para isso é por que os pais querem (conscientemente ou não) apresentar seus filhos como seres bem inteligentes. Assim, os pais perguntam o tempo todo aos seus filhos “e o que é isso?” de forma que a criança possa responder e mostrar quantas palavras diferentes eles conhecem. Alguns pesquisadores então decidiram modificar a forma de coleta de dados, e decidiram colocar um gravador no quarto da criança que liga em intervalos aleatórios, sem que os pais saibam quando ele está gravando. Os resultados mostraram que o número de perguntas diminuiu consideravelmente...

Em estudos de corpora, você irá encontrar um tipo diferente de aleatorização. Por exemplo, um pesquisador primeiramente busca todos os casos da palavra em que ele está interessado e então coloca esses casos em ordem de acordo com números aleatórios. Quando o pesquisador analisa os primeiros 20% da lista, ele tem uma amostra aleatória. Aleatorização é um dos princípios mais importantes do processo de coleta de dados.

Uma vez que seus dados estão coletados, você precisa armazená-los em um formato que facilitará manipulações, análises e anotações posteriores. É muito comum ver pessoas – estudantes e pesquisadores com experiência – imprimindo listas longas de dados para fazer anotações à mão, ou fazer linhas de concordância em um corpus utilizando um editor de texto. Tal procedimento pode até ser apropriado para bancos de dados pequenos, mas certamente não será conveniente para bancos de dados maiores. Para tais casos, a melhor maneira de manipular os dados é através de planilhas eletrônicas (ex.: OpenOffice.org Calc) ou no R. No entanto, há um conjunto de regras básicas que define o chamado “formato de caso por variável” (*case-by-variable format*) desejado e que precisa ser levado em consideração.

- i a primeira linha deve conter os nomes de todas as suas variáveis;
- ii cada uma das linhas subsequentes deve representar apenas um dado (único e exclusivamente um dado, onde eu estou usando um dado para referir-me a uma simples observação de uma variável independente;
- iii a primeira coluna deve apenas enumerar todos os n casos (de 1 a n) de forma que cada linha possa ser exclusivamente identificada e que a ordem original possa sempre ser restaurada;
- iv cada coluna subsequente deve representar uma e apenas uma variável ou característica para a qual cada dado será anotado. Em uma planilha para um estudo de corpora, por exemplo, uma coluna adicional pode conter o nome do arquivo de corpus onde a palavra foi encontrada; uma outra coluna pode conter a linha em que a palavra foi encontrada no corpus. Em uma planilha de um estudo experimental, uma coluna deve conter algum identificador único para cada participante; outras colunas podem conter a idade do participante, o sexo, os estímulos apresentados (ou condições experimentais), a ordem da apresentação dos estímulos (para verificar, por exemplo, se o desempenho do participante muda sistematicamente no decorrer do experimento), ...;
- v os dados que estão faltando são introduzidos como NA e não apenas com células vazias (o que também significa que nenhum outro nível variável deve ser abreviado como NA) com o objetivo de preservar a integridade formal do conjunto de dados (isto é, ter todas as linhas e as colunas com o mesmo número de elementos) e ser capaz de fazer estudos subsequentes sobre os dados em falta para verificar se, por há um padrão nos pontos de dados em falta que necessita ser considerado.

Algumas sugestões adicionais muito úteis especialmente para trabalhar com o R são ter os nomes das colunas na primeira linha em letras maiúsculas, nunca codificar os níveis dos níveis categóricos como números, mas como palavras/seqüências de caracteres em letras minúsculas, e não usar caracteres ‘estranhos’ tais como espaços, pontos finais, vírgulas, tabulações, #, aspas simples ou duplas, entre outros, em nomes de variáveis ou níveis.

Para assegurar que o que dissemos até agora está claro, vamos ver dois exemplos. Imagine que para o estudo de colocação de partícula você analisou algumas frases e contou o número de sílabas dos objetos diretos. A primeira pergunta é: qual é a variável dependente e a variável independente?

Pausa para pensar

A variável independente é a variável proporcional TAMANHO (medida em sílabas), que pode assumir qualquer valor inteiro positivo. A variável dependente é a variável nominal (ou categórica) CONSTRUÇÃO, que pode ser VPO ou VOP. Quando todas as hipóteses são formuladas e, subsequentemente, os dados são coletados e codificados, temos uma representação parecida com a apresentada na Tabela 5.

Tabela 5 – Uma tabela não muito boa 1

	Tamanho: 2	Tamanho: 3	Tamanho: 5	Tamanho: 6
Construção: VPO				
Construção: VOP				

Um outro exemplo: vamos investigar a hipótese de que sujeitos e objetos diretos têm tamanhos diferentes (tamanho medido em palavras). A mesma pergunta de antes: qual é a variável dependente e qual é a variável independente?

Pausa para pensar

A variável independente é a variável nominal RELAÇÃO, a qual pode ser sujeito ou objeto. A variável dependente é a variável TAMANHO, que pode assumir qualquer valor inteiro positivo. Imagine que você já formulou as quatro hipóteses (hipótese alternativa: textual e estatística; hipótese nula: textual e estatística) e analisou o pequeno corpus em (3). Neste caso, sua planilha não deve parecer com a Tabela 6.

Exemplo (3)

- a As crianças menores comeram um algodão doce grande.
- b Ele trancou a porta.
- c A rápida raposa marrom atingiu o cão preguiçoso.

Tabela 6 – Uma tabela não muito boa 2

Frase	Suj.	Obj.
As crianças menores comeram um algodão doce grande.	3	4
Ele trancou a porta.	1	2
A rápida raposa marrom atingiu o cão preguiçoso.	4	3

Ambas as Tabelas 5 e 6 não estão de acordo com as regras básicas mencionadas acima. Na Tabela 6, por exemplo, cada linha representa dois dados de uma vez (o tamanho do sujeito e o tamanho do objeto direto). Outra violação: nem todas as variáveis estão representadas por uma coluna única e exclusiva. Ela apresenta duas colunas que, na verdade, representam os níveis da variável independente. Antes de continuar a leitura, como você re-organizaria a Tabela 6 para que ela fique de acordo com as regras básicas mencionadas acima?

Uma maneira melhor para representar os dados pode ser vista na Tabela 7: cada dado tem sua própria linha e é caracterizado de acordo com as duas variáveis em suas respectivas colunas. Uma outra versão ainda mais completa dessa tabela poderia incluir uma coluna com apenas os sujeitos e objetos de

Pausa para pensar

forma que a localização de casos específicos seja mais fácil. Na primeira linha dessa nova versão, você encontraria *As crianças menores*, na segunda linha da mesma coluna você encontraria *um algodão doce grande* etc. A mesma lógica se aplica para a melhoria da Tabela 6. Veja a Tabela 7.

Tabela 7 – Uma representação melhor dos dados da Tabela 6

Caso	Frase#	Frase	Relação	Tamanho
1	1	As crianças menores comeram um algodão doce grande.	subj.	3
2	1	As crianças menores comeram um algodão doce grande.	obj.	4
3	2	Ele trancou a porta.	subj.	1
4	2	Ele trancou a porta.	obj.	2
5	3	A rápida raposa marrom atingiu o cão preguiçoso.	subj.	4
6	3	A rápida raposa marrom atingiu o cão preguiçoso.	obj.	3

Com pouquíssimas exceções, essa é a maneira como você sempre deve armazenar os seus dados.⁹ O ideal é que você armazene seus dados nesse formato em uma planilha e salve os dados (i) no arquivo no formato original do programa que utilizar (assim você preserva as cores e outras formatações específicas que você pode ter adicionado) e (ii) no arquivo no formato de texto (*tab-delimited*), pois é um arquivo mais facilmente importado para o R.

Tabela 8 – Uma representação melhor dos dados da Tabela 5

CASO	CONSTRUÇÃO	TAMANHO
1	VPO	2
2	VPO	2
3	VOP	2
4	VOP	2
5	VOP	2
6	VOP	2
7	VPO	3
8	VPO	3
9	VOP	3
10	VOP	3
11	VOP	3
...	...	...

Todos os passos que dizem respeito à coleta e armazenamento dos dados devem ser descritos na seção Métodos do seu texto escrito: qual é a sua população, como você selecionou sua amostra (idealmente) representativa e balanceada, para quais variáveis você coletou dados, etc.

1.3.4 A decisão

Uma vez que seus dados já foram coletados e armazenados de acordo com a Tabela 7 e a Tabela 8, você pode finalmente fazer o que estava querendo o tempo todo: a análise dos seus dados. (Por enquanto, não vou explicar como você decide qual teste estatístico escolher, mas vou voltar a esse tópico no início do Capítulo 4.) Como resultado de suas análises, você irá obter frequências, médias, dispersões, coeficientes de correlação ou distribuições. No entanto, um aspecto importante da sua análise é que, ao contrário do que muitos pensam, você não tenta apenas mostrar que sua hipótese alternativa está correta, mas

⁹ Existem técnicas estatísticas mais complexas que podem exigir formatos diferentes, mas na grande maioria dos casos, o formato padrão discutido acima (também chamado de padrão longo) é o que você irá precisar e que irá lhe permitir mudar facilmente para outro formato.

sim que a versão estatística da hipótese nula está errada. Uma vez que a hipótese nula é a contrapartida lógica da hipótese alternativa, mostrar que a H_0 está errada reforça a H_1 . A pergunta óbvia agora é: por que fazer dessa forma? A resposta a essa pergunta pode ser dada novamente fazendo-se referência ao exemplo de sujeitos e objetos.

Vamos supor que você formulou estas hipóteses:

H_0 : Sujeitos e objetos diretos em orações transitivas não têm tamanhos diferentes.

H_1 : Sujeitos e objetos diretos em orações transitivas têm tamanhos diferentes.

Agora considere as seguintes questões:

- quantos sujeitos e objetos diretos você precisa ter no seu estudo para mostrar que a hipótese H_1 está correta?
- quantos sujeitos e objetos diretos você precisa ter no seu estudo para mostrar que a hipótese H_0 está incorreta?

Pausa para pensar

Você provavelmente já deve ter percebido que a resposta para a primeira pergunta é “muitos”. Falando de uma maneira geral, você só pode mostrar que a hipótese alternativa está correta se analisar todo e qualquer sujeito e objeto direto que existe e não encontrar nenhum exemplo contrário à hipótese. Por outro lado, para mostrar que a hipótese nula está incorreta, basta analisar caso-a-caso e, assim que aparecer um exemplo contrário, a hipótese está incorreta. No entanto, especialmente em Ciências Humanas e Sociais, você geralmente não rejeita uma hipótese com base em apenas um exemplo contrário. Em vez disso, você usa o seguinte procedimento de quatro etapas, o qual é às vezes chamado de o paradigma do Teste de Significância da Hipótese Nula (*NULL Hypothesis Significance Testing* - NHST):

- Você define o chamado *nível de significância* $p_{\text{crítico}}$, que normalmente é definido como 0,05 (ou seja, 5 %) e representa o valor limite para a rejeição ou aceitação da H_0 ;
- Você analisa seus dados calculando algum efeito e usando a estatística em suas hipóteses estatísticas;
- Você calcula o chamado *probabilidade de erro* p , ou seja, a probabilidade de encontrar e ou algo que se desvia ainda mais de H_0 em sua amostra quando, na população, H_0 for verdadeira;
- você compara $p_{\text{crítico}}$ e p e decide: se $p < p_{\text{crítico}}$, então você pode rejeitar a H_0 e aceitar a H_1 – caso contrário, você deve ficar com a H_0 .

Por exemplo, se na sua amostra a diferença média de tamanho entre sujeitos e objetos diretos for de 1,4 sílabas, então você deve calcular a probabilidade de erro p para encontrar essa diferença de 1,4 sílabas ou até mesmo uma diferença maior quando você, na verdade, não espera essa diferença (porque é isso que a H_0 prevê). Então, existem duas possibilidades:

- se essa probabilidade p de uma diferença de 1,4 sílabas for menor que o $p_{\text{crítico}}$ de 5%, então você pode rejeitar a hipótese nula que diz que não há diferença entre o tamanho de sujeitos e objetos diretos na população. Na seção de resultados do seu artigo, você pode então escrever que encontrou uma diferença significativa entre as médias na sua amostra, e na seção de discussão, você discute as implicações desse achado, etc.

- se essa probabilidade p for igual ou maior que o $p_{\text{crítico}}$ de 5%, então você não pode rejeitar a hipótese nula que diz que não há diferença entre o tamanho médio de sujeitos e objetos diretos na população. Na seção de resultados do seu artigo, você deve admitir que não encontrou uma diferença significativa na sua amostra entre o tamanho médio do sujeito e do objeto direto. E na seção de discussão, você deve discutir as implicações desse achado e também especular ou cogitar sobre os motivos pelos quais você não encontrou uma diferença significativa – pode ter havido algum *outlier* nos dados do corpus ou no experimento (porque os participantes reagiram de forma estranha a um estímulo específico, por causa de erros na codificação, etc.).¹⁰

Dois aspectos dessa lógica são importantes. Primeiramente, o fato de que um efeito é significativo não quer dizer necessariamente que é importante, a despeito do que o uso comum da palavra “significativo” sugere. Em Estatística, a palavra significativo é utilizada como um termo técnico que significa que o efeito (aqui correspondendo à diferença) é grande o suficiente para que possamos assumir que, dado o tamanho da amostra, esse efeito provavelmente não ocorreu devido ao acaso. Além disso, só porque você aceitou a hipótese alternativa com base em um resultado significativo, isso não quer dizer que você *provou* que a hipótese alternativa está correta. Isso porque ainda assim existe uma probabilidade de erro p é baixa o suficiente para garantir a *aceitação* da hipótese alternativa, mas não para prová-la.

Essa linha de raciocínio pode parecer um pouco confusa no início, especialmente porque estamos falando de duas probabilidades distintas. Uma é a probabilidade de 5% (com a qual a outra probabilidade é comparada), e a outra é a probabilidade de obtenção dos dados observados dado que a hipótese nula está correta. A primeira, o nível de significância $p_{\text{crítico}}$, é *definida antes que os dados sejam obtidos*, enquanto que a segunda, a probabilidade de erro, é o chamado valor- p *que é calculado com base nos dados*. Por que essa probabilidade é chamada de probabilidade de erro? É porque - lembre-se do que foi dito cima. Às vezes, você observará que as pessoas usam palavras diferentes para diferentes valores- p :

- $p < 0,001$ é geralmente chamado de *altamente significativo* e é indicado por ***;
- $0,001 \leq p < 0,01$ é geralmente chamado de *muito significativo* e é indicado por **;
- $0,01 \leq p < 0,05$ é geralmente chamado de *significativo* e é indicado por *;
- $0,05 \leq p < 0,1$ é geralmente chamado de *marginalmente significativo* e é indicado por *ms* ou um ponto final, mas uma vez que esses valores- p são maiores que o padrão usual de 5%, chamar esses resultados de marginalmente significativos, é como se o pesquisador dissesse: “Olha, não encontrei os resultados significativos que estava procurando, mas mesmo assim os dados são bem interessantes, não acha?”, e é por isso que eu desencorajo o uso dessa expressão.

Aviso Importante

Você nunca deve modificar suas hipóteses *depois* que obteve seus resultados para dar a impressão de que seu estudo provou uma hipótese nova ou foi um estudo bem sucedido. Outra coisa: você nunca deve ficar procurando por algo significativo nos seus dados e, somente após encontrar, publicar os resultados como se tivesse formulado as hipóteses a priori. Você deve sim, explorar seus dados para encontrar padrões interessantes, mas nunca utilizar os dados para gerar hipóteses. Se isso acontecer, você deve testar as hipóteses utilizando um outro banco de dados.

Apesar de termos visto acima que a comparação entre as duas probabilidades contribui para a nossa decisão à favor ou contra a hipótese alternativa, ainda não sabemos como o valor- p é calculado.

¹⁰ *Outliers* são valores na amostra que são bastante atípicos considerando-se o resto da amostra.

1.3.4.1 Valores unicaudais de distribuições de probabilidades discretas

Imagine que decidimos tirar cara-ou-coroa 100 vezes. Se cair cara, você me paga R\$ 1,00 (um real). Se cair coroa, eu pago R\$ 1,00 para você. Antes de começar o jogo, vamos formular as seguintes hipóteses:

H_0 textual: Stefan não rouba: probabilidade de cair cara e coroa é 50% *versus* 50%.

H_1 textual: Stefan rouba: a probabilidade de cair cara é maior que 50%.

Esse cenário pode ser facilmente operacionalizado usando frequências:

H_0 estatística: Stefan irá vencer com a mesma frequência que o leitor irá vencer, ou seja 50 vezes.

H_1 estatística: Stefan vai vencer mais do que o leitor, ou seja, vai vencer mais de 50 vezes.

Agora perguntamos: depois que jogamos as 100 vezes, depois de qual dos resultados abaixo você desconfiaria que eu roubei?

Pausa para pensar

- Quando você perder 51 vezes (provavelmente não)?
- Quando você perder 55 vezes? Ou 60 vezes? (hmmm... talvez)?
- Quando você perder 80 vezes ou mais (alguma coisa tem...)?

Talvez sem perceber, você está fazendo uma espécie de teste de significância. Vamos tornar isso mais concreto (imaginando que você perdeu 60 vezes) e também parafrasear isso em termos dos quatro passos do paradigma de teste de significância da hipótese nula:

- i Vamos supor que você defini o nível de significância $p_{\text{crítico}}$ para o seu valor habitual de 0,05;
- ii Você observa o efeito, e perde 60 vezes;
- iii Você tenta calcular a chamada probabilidade de erro p , a probabilidade de perder 60 vezes ou mais na amostra (nosso jogo de 100 lances) quando H_0 é verdadeira e você deveria ter perdido 50 vezes. Por que "60 vezes ou mais"? Bem acima dissemos

Você calcula a chamada probabilidade de erro p , a probabilidade de encontrar e ou algo que se desvia de H_0 ainda mais em sua amostra quando, na população, H_0 for verdadeira;
- iv Se você pode calcular p , você compara o $p_{\text{crítico}}$ e p e decide no que acredita: se $p < p_{\text{crítico}}$, então você pode rejeitar a H_0 , aceitar a H_1 , e me acusar de trapaçar – caso contrário, você deve manter a H_0 e aceitar suas derrotas.

Assim, você deve se perguntar como e quanto o resultado observado se desvia do resultado esperado pela hipótese nula. Claramente o número de vezes que você perdeu é maior: $60 > 50$. Assim, os resultados que se desviam da hipótese nula são aqueles iguais ou maiores que 60: 60 vezes, 61 vezes, 62 vezes, ..., 99 vezes e 100 vezes. Utilizando um jargão mais técnico, você definiu seu nível de significância em 5% (0,05) e se pergunta “qual é a probabilidade de que Stefan não tenha roubado, mas ainda assim vencido 60 vezes, uma vez que ele só deveria ter vencido 50 vezes?” Essa é a lógica por trás de testes de significância.

É possível mostrar que a probabilidade p de se perder 60 vezes ou mais por um acaso – sem que eu esteja roubando – é 0,02844397, ou seja, 2.8%. Uma vez que esse valor- p é menor que o nosso limite estabelecido, 0,05 (ou 5%), você pode me acusar de ter roubado. Imagine no entanto que somos muito amigos e que, para não arriscar a amizade, você tenha estabelecido o nível de significância em 1%. Dessa

forma, você não poderia me acusar de estar roubando, uma vez que a probabilidade p observada é $0,02844397 > 0,01$.

Espero que o exemplo acima tenha esclarecido ainda mais a lógica por trás do teste de hipóteses, mas o que provavelmente ainda não está claro é como o valor- p foi calculado. Para ilustrar isso, vamos diminuir o nosso exemplo: ao invés de 100 jogadas, vamos fazer 3 jogadas. A Tabela 9 mostra todas as possibilidades de resultados para 3 jogadas de cara-ou-coroa e suas respectivas probabilidades, assumindo que a hipótese nula esteja correta, ou seja, 50% para cara e 50% para coroa. Na Tabela 9, as três primeiras colunas representam os resultados possíveis, a coluna 4 e 5 representam o número de caras e o número de coroas observados em cada uma das oito possibilidades, e a última coluna exibe as probabilidades para cada resultado. Note que as probabilidades são as mesmas para todas as oito possibilidades, 0,125. Será por que?

Existem duas explicações plausíveis para explicar isso, e ambas requerem que você entenda o conceito de *independência*.

Tabela 9 – Todos os resultados possíveis para as três jogadas de cara-ou-coroa e suas respectivas probabilidades (H_0 está correta)

Jogada 1	Jogada 2	Jogada 3	# cara	# coroa	$p_{\text{resultado}}$
cara	cara	cara	3	0	0,125
cara	cara	coroa	2	1	0,125
cara	coroa	cara	2	1	0,125
cara	coroa	coroa	1	2	0,125
coroa	cara	cara	2	1	0,125
coroa	cara	coroa	1	2	0,125
coroa	coroa	cara	1	2	0,125
coroa	coroa	coroa	0	3	0,125

A primeira envolve a compreensão de que a probabilidade de cair cara e a probabilidade de cair coroa são as mesmas em cada jogada e que cada jogada é independente uma da outra. Dizer que cada jogada é independente é o mesmo que dizer que o resultado de uma jogada (cara ou coroa) não tem nenhuma influência no resultado de nenhuma outra jogada). De maneira semelhante, as amostras (as oito possibilidades) são independentes umas das outras, uma vez que não há razão para que os valores de uma amostra sejam ligados a valores de uma outra amostra. Por exemplo, se você retirar de um corpus uma amostra aleatória de 100 orações transitivas e contar o tamanho (em sílabas) dos sujeitos dessas orações e depois retirar, do mesmo corpus, outras 100 orações transitivas e contar o tamanho (em sílabas) dos objetos diretos dessas orações, as duas amostras – 100 tamanhos de sujeito e 100 tamanhos de objetos diretos – serão independentes. Se por outro lado, você retirar do corpus, aleatoriamente, 100 orações transitivas e contar o tamanho dos sujeitos e objetos diretos de cada uma delas, as duas amostras – 100 tamanhos de sujeito e 100 tamanhos de objetos diretos – serão dependentes, uma vez que você pode combinar perfeitamente os 100 sujeitos aos 100 objetos diretos, já que ambos vieram da mesma frase. Da mesma forma, se você faz o mesmo experimento com um participante duas vezes, os resultados do primeiro e do segundo experimentos são dependentes. Essa distinção ficará mais clara adiante.

Vamos voltar às três jogadas de cara-ou-coroa: uma vez que há oito possibilidades distintas e cada possibilidade é independente uma da outra, e consequentemente igualmente prováveis, a probabilidade de cada resultado será $\frac{1}{8} = 0,125$.

A segunda maneira de entender a Tabela 9 envolve o cálculo de cada evento separadamente. Vamos ver a primeira linha: a probabilidade de cair cara na primeira jogada, na segunda e na terceira é sempre 0,5. Uma vez que as jogadas são independentes umas das outras, você consegue obter a probabilidade de conseguir três caras multiplicando as probabilidades individuais: $0,5 \times 0,5 \times 0,5 = 0,125$ (essa é a regra da multiplicação na Teoria da Probabilidade). Se fizer o mesmo para cada linha subsequente irá encontrar o mesmo resultado, 0,125. Portanto, podemos mostrar que a hipótese nula prediz que cada um de nós deve ganhar aproximadamente 1,5 vezes em média (isto é, se jogássemos o jogo de três jogadas 100 vezes).

Agora, imagine que você perca duas em três jogadas. Se você tivesse estabelecido o mesmo nível de significância de 5%, você me acusaria de estar roubando?

Pausa para pensar

Claro que não! O que você deve considerar para responder a essa pergunta? Você deve considerar o resultado obtido – você perdeu duas vezes – e você precisa considerar também os outros resultados que são incompatíveis com a hipótese nula de acordo com a direção prevista. Nesse exemplo, a única outra possibilidade incompatível com a hipótese nula é que você perca as três vezes. Vamos então calcular a soma das probabilidades desses eventos.

Se você observar na coluna 4, vai ver que existem três resultados em que você perde duas vezes em três lançamentos: cara-cara-coroa (linha 2), cara-coroa-cara (linha 3) e coroa-cara-cara (linha 5). Assim, a probabilidade de perder exatamente duas vezes é $0,125 + 0,125 + 0,125 = 0,375$. Essa probabilidade é muito maior que o nível de significância de 0,05. No entanto, você ainda deve adicionar a essa probabilidade, a probabilidade do outro evento incompatível com a hipótese nula que é 0,125. Se somar todas essas probabilidades, a probabilidade p de perder duas vezes ou mais em um total de três jogadas é 0,5. Uma vez que essa probabilidade é 10 vezes maior que o nível de significância, acusar-me de estar roubando nesse caso seria insano. Isso é dez vezes mais do que o nível de significância, então não há nenhuma chance de você poder me acusar de trapaça. Note que mesmo se você tivesse perdido todos as três jogadas, você ainda não poderia me acusar de trapaça, porque a probabilidade de que isso aconteça quando H_0 é verdade ainda é 0,125.

Podemos também representar esta lógica graficamente e ao mesmo tempo considerar um número maior de jogadas. A Figura 4 tem seis painéis, um para 3 jogadas, um para 6, um para 12, e depois para 25, 50 e 100. Em cada um, as probabilidades somadas para todos os números possíveis de caras, dado o número de jogadas feitas, estão representadas como barras, e o resultado mais extremo (eu sempre venço) é representado com uma barra cinzenta e uma seta que aponta para ela. Nos casos de 3 e 6 jogadas, eu também plotei as probabilidades desses eventos no topo das barras.

Assim, se você perdeu mais frequentemente do que deveria de acordo com a H_0 e quer determinar a probabilidade de perder tantas vezes e até mesmo mais frequentemente, você se distancia da expectativa de H_0 , a qual está no meio (ao longo do eixo x) do gráfico, longe do resultado observado (digamos, $x = 3$) e adiciona o comprimento dessa barra aos comprimentos de todas as outras barras que você encontrar se você continuar se movendo na mesma direção, onde há apenas uma barra $x = 3$, e então você está resolvido.

A Figura 4 também ilustra outro ponto muito importante. Primeiro, lembre-se que a distribuição básica subjacente a estes dados é uma distribuição de probabilidade discreta e não normal, ou seja, 0,5 (caras) vs. 0,5 (coroas). Em segundo lugar, na medida em que os números de jogadas em nossos jogos aumentam, as probabilidades dos resultados possíveis se parecem cada vez mais como a curva em forma de sino que conhecemos de distribuições normais. Assim, embora a distribuição subjacente não seja normal, uma vez que o tamanho da amostra se torna grande o suficiente, ainda obtemos uma curva em forma de sino. Isto também significa que, se os dados sob investigação estiverem distribuídos de forma suficientemente similar à distribuição normal (ou outra de várias funções de densidade de probabilidade amplamente utilizadas, como a distribuição F -, t -, ou a χ^2), então não é necessário computar, e somar, probabilidades exatas como fizemos acima, mas pode-se aproximar o valor- p de parâmetros de equações subjacentes às distribuições acima. Isto é frequentemente chamado de "uso de testes paramétricos". Esta aproximação de um valor- p com base em uma função pode ser apenas tão boa quanto o ajuste distribucional dos dados à função correspondente. Vamos rever isso abaixo.

1.3.4.2 Valores- p bicaudais de distribuições de probabilidades discretas

Agora, precisamos adicionar outra perspectiva. Na última seção, só lidamos com hipóteses alternativas uni-direcionais: a hipótese alternativa foi "Stefan rouba: a probabilidade de cair cara é maior do que

Figura 4 – Todas as possibilidades de resultados possíveis de 3, 6, 12, 25, 50, 100 lançamentos de moedas e suas probabilidades (quando H_0 é verdadeira, unicaudal)

50% [e não apenas diferente de 50%]”. O tipo de teste de significância nesse caso é chamado de teste uni-direcional, porque você estava apenas interessado em uma direção na qual o resultado observado se desvia do resultado esperado (digamos, porque você sabia com certeza que você não trapaceou). Assim, quando você somou os comprimentos de barras na Figura 6, você apenas se afastou da expectativa de H_0 em uma direção.

Porém, geralmente precisamos lidar com hipóteses alternativas não-direcionais. Nestes casos, devemos analisar os resultados que podem se desviar do resultado esperado nas duas direções. Pense novamente no jogo em que você e eu jogamos uma moeda três vezes, mas desta vez também temos um observador imparcial que não tem razão para suspeitar que só eu estaria trapaceando. Ele, portanto, formula as seguintes hipóteses (com um nível de significância de 0,05):

H_0 estatística: Stefan irá vencer na mesma proporção que o outro jogador, ou seja, 50 vezes (ou “Ambos os jogadores irão vencer o mesmo tanto”).

H_1 estatística: Stefan irá vencer numa proporção diferente do outro jogador - ou mais, ou menos (ou “Os jogadores não irão vencer o mesmo tanto”).

Imagine agora que você perdeu três vezes. O observador então se pergunta se um de nós pode ser acusado de ter roubado. Assim como antes, ele precisa determinar quais eventos considerar, e ele também usa uma tabela de todos os resultados possíveis para ajudá-lo a descobrir o que aconteceu. Considere, portanto, a Tabela 10.

Primeiro, ele precisa considerar o resultado observado: que a você perdeu três vezes, o que está listado na linha 1 e tem uma probabilidade de 0,125. Mas ele também deve considerar as probabilidades de outros resultados que são incompatíveis com a hipótese nula. Com uma hipótese alternativa uni-direcional, você se distanciou da hipótese nula em apenas uma direção, mas dessa vez, não há hipótese

Tabela 10 – Todos os resultados possíveis para as três jogadas de cara-ou-coroa e suas respectivas probabilidades (H_0 está correta)

Jogada 1	Jogada 2	Jogada 3	# cara	# coroa	$p_{\text{resultado}}$
cara	cara	cara	3	0	0,125
cara	cara	coroa	2	1	0,125
cara	coroa	cara	2	1	0,125
cara	coroa	coroa	1	2	0,125
coroa	cara	cara	2	1	0,125
coroa	cara	coroa	1	2	0,125
coroa	coroa	cara	1	2	0,125
coroa	coroa	coroa	0	3	0,125

uni-direcional de forma que o observador deve procurar desvios tão grandes ou maiores na outra direção da expectativa da hipótese nula. Como você pode ver na Tabela 10, há outro desvio da H_0 que é tão extremo quanto o outro, a dizer, que eu perco três vezes. Uma vez que o observador tem apenas uma hipótese não-direcional, ele inclui a probabilidade desse evento, também, chegando a uma probabilidade cumulativa de 0,25. Esta lógica é representada graficamente na Figura 5 da mesma maneira que acima.

Figura 5 – Todas as probabilidades de resultados possíveis de 3, 6, 12, 25, 50, 100 lançamentos de moedas e suas probabilidades (quando H_0 for correta, bicaudal)

Note que quando você testou a sua hipótese alternativa uni-direcional, você analisou o resultado “você perdeu três vezes”, mas quando o observador imparcial testou a hipótese dele, ele analisou o resultado “alguém perdeu três vezes”. A consequência disso é importante: quando você conhece o fenômeno que deseja investigar o suficiente a ponto de conseguir formular uma hipótese alternativa uni-direcional, o valor- p necessário para se ter um resultado significativo é bem menos extremo comparado ao valor- p necessário para um resultado significativo com uma hipótese bi-direcional. Na maioria dos casos, o valor- p que você encontra em estudos com hipóteses bi-direcionais é o dobro do valor- p

encontrado para hipóteses uni-direcionais. Ter conhecimento prévio do fenômeno de interesse é, de fato, um ganho importante para o estudo. Vamos ilustrar esse fato.

Vamos então voltar ao nosso exemplo com 100 jogadas. Vamos analisar novamente o jogo sob a sua hipótese uni-direcional, e, depois, sob a hipótese bi-direcional do observador imparcial. No entanto, desta vez você e o observador tentam determinar, antes do início do jogo, quais resultados serão considerados extremos a ponto de rejeitar a hipótese nula e aceitar a hipótese alternativa. Vamos começar com a sua perspectiva: Na Figura 6, você vê o gráfico das 100 jogadas com a frequência esperada para cara: 50 vezes. (O significado das linhas pretas será explicado em breve.)

Figura 6 – Todos os resultados possíveis para as 100 jogadas de cara-ou-coroa e suas respectivas probabilidades (H_0 está correta, H_1 uni-direcional)

Apresentamos anteriormente um resultado empírico onde estávamos interessados no seu valor- p . Para obtermos esse valor- p , nos distanciamos dos resultados esperado de acordo com a hipótese nula e movemos em direção aos valores mais extremos. Queremos agora determinar, mas não exceder, um valor- p antes de termos os resultados, e então temos que proceder da maneira inversa: do ponto extremo em direção ao resultado esperado de acordo com a hipótese nula. Por exemplo, para determinar quantas vezes você pode perder sem que a probabilidade acumulada ultrapasse 0,05, você começa do resultado mais extremo à direita – que você perde 100 vezes – e vai adicionando as probabilidades das barras (Claro, você calcularia isso, e não mediria comprimentos literalmente). A probabilidade de você perder todas as 100 jogadas é $7,8886 \times 10^{-31}$. Adicione a essa probabilidade a probabilidade de você perder 99 vezes, e depois a probabilidade de perder 98 vezes, etc. Quando você tiver somado todas as probabilidades até chegar ao resultado de você perder 59 vezes, você verá que a probabilidade acumulada é de 0,0443; todas essas probabilidades estão representadas em preto na Figura 6. Uma vez que a probabilidade de você perder 58 vezes é 0,0223, você não pode adicionar a probabilidade desse evento às outras probabilidades sem exceder o nível de significância de (0,05). Em outras palavras, se você não quer eliminar mais de 5% dos comprimentos de barras somados, você deve parar de adicionar probabilidades quando chegar no $x = 59$. Você então conclui: “se o Stefan ganhar 59 vezes ou mais, então eu irei acusá-lo de estar roubando, pois a probabilidade disso acontecer é, dentre as menores que 0.05, a mais alta de todas.

Observe agora a perspectiva do observador mostrada na Figura 7, que é muito semelhante, porém não completamente idêntica, à Figura 6. O observador também deve começar no resultado mais extremo: eu ganho todas as vezes: 100 caras $\approx 7,8886 \times 10^{-31}$. No entanto, como o observador está trabalhando com uma hipótese alternativa bi-direcional, ele também deve incluir o outro resultado extremo: nós tiramos caras 0 vezes. Para cada resultado adicional (99, 98, 97 ...), o observador deve incluir os respectivos valores do lado oposto (1, 2, 3 ...). Assim que o observador adicionar todas as probabilidades – 61 vezes caras / 39 vezes coroa e 39 vezes cara / 61 vezes coroa – então a probabilidade acumulada será 0,0352 (as barras pretas na Figura 7).

Figura 7 – Todos os resultados possíveis para as 100 jogadas de cara-ou-coroa e suas respectivas probabilidades (H_0 está correta, H_1 uni-direcional)

Uma vez que a probabilidade conjunta dos dois próximos eventos – 60 caras / 40 coroas e 40 caras / 60 coroas – é 0,0217, o observador não pode mais adicionar essa probabilidade sem que a probabilidade acumulada ultrapasse o nível de significância de 0,05. Em outras palavras, se o observador não quer ultrapassar a marca dos 5% nos dois lados, ele deve parar de adicionar probabilidades quando chegar no $x = 61$ (indo da direita para a esquerda) e $x = 39$ (indo da esquerda para direita). O observador então conclui: se o Stefan ou oponente dele ganhar 61 vezes ou mais, então alguém está roubando (mais provavelmente a pessoa que ganha mais frequentemente).

Novamente, observe que, na mesma situação, a pessoa com a hipótese alternativa uni-direcional precisou de um valor menos extremo para aceitar a hipótese como correta em comparação à pessoa com a hipótese alternativa bi-direcional: com o mesmo nível de significância, você poderia acusar-me de estar roubando quando você perdesse 59 vezes (apenas 9 vezes a mais que o resultado esperado). Já o observador imparcial poderia acusar alguém de estar roubando assim que alguém perdesse 61 vezes (11 vezes a mais que o valor esperado). Em outras palavras, se você perder 60 vezes, você pode acusar-me de estar roubando, já o observador imparcial não pode. Essa diferença é muito importante e vamos utilizá-la com frequência.

Enquanto lia as últimas páginas, você provavelmente estava se perguntando de onde vêm as probabilidades dos eventos: Como sabíamos que a probabilidade de tirar 100 caras em 100 jogadas é $7,8886 \times 10^{-31}$?

Basicamente, esses valores foram computados da mesma forma que fizemos para a Tabela 9 e para a Tabela 10, isto é, foram computados no programa R com base na chamada distribuição binomial. Você pode facilmente calcular a probabilidade de ocorrência de uma dentre duas possibilidades de eventos possíveis ocorrer x vezes, num total s de possibilidades, quando a probabilidade do evento é p no R utilizando a função `dbinom`¹¹. Os argumentos dessa função são:

- `x`: a frequência de ocorrência do evento (ex.: três vezes cara);
- `size`: o número total de tentativas em que o evento pode ocorrer (ex.: três jogadas);
- `prob`: a probabilidade de ocorrência do evento em cada tentativa (ex.: 50%).

Você sabe que probabilidade de tirar três caras em três jogadas – quando a probabilidade de tirar cara em cada jogada é 50% – é 12,5%. No R:

Código em R

```
> dbinom(3, 3, 0.5)¶  
[1] 0.125
```

Você também pode calcular a probabilidade de todas as quatro possibilidades de cair cara – 0, 1, 2 e 3 – utilizando apenas uma linha de comando (porque, como veremos mais adiante, sequências de números inteiros podem ser definidas com dois pontos):

Código em R

```
> dbinom(0:3, 3, 0.5)¶  
[1] 0.125 0.375 0.375 0.125
```

Da mesma forma, você consegue calcular a probabilidade de que cara vai cair duas ou três vezes somando as probabilidades relevantes:

Código em R

```
> sum(dbinom(2:3, 3, 0.5))¶  
[1] 0.5
```

Agora, você consegue fazer o mesmo para calcular a probabilidade de tirar 100 caras em 100 jogadas,

Código em R

```
> dbinom(100, 100, 0.5)¶  
[1] 7.888609e-31
```

a probabilidade de tirar 58 ou mais caras em 100 jogadas (a qual é maior que 5% e não permite que você aceite a H_1 unicaudal),

Código em R

```
> sum(dbinom(58:100, 100, 0.5))¶  
[1] 0.06660531
```

¹¹ Eu explicarei como instalar o R etc. no próximo capítulo. Não importa se você não instalou o R e/ou não é capaz de inserir ou entender o *input* acima ainda. Voltaremos a este ponto...

a probabilidade de tirar 59 caras ou mais em 100 (a qual é menor que 5% e não permite que você aceite a H_1 unicaudal):

Código em R

```
> sum(dbinom(59:100, 100, 0.5))  
[1] 0.04431304
```

De fato, você não precisaria fazer isso por tentativa e erro como o exposto acima pode sugerir. Você pode usar a função `qbinom` para obter o maior número de caras cuja probabilidade cumulativa com cada resultado ainda mais extremo não exceda a 0,05, e você pode ver que isso corresponde ao achado acima:

Código em R

```
> qbinom(0.05, 100, 0.5, lower.tail=FALSE)  
[1] 58
```

Para testes bi-direcionais, você pode fazer o mesmo, ou seja, computar a probabilidade de tirar cara 40 vezes ou menos, ou 60 vezes ou mais (uma probabilidade que é maior que 5% e não permite que você aceite uma hipótese alternativa bicaudal/não-direcional):

Código em R

```
> sum(dbinom(c(0:40, 60:100), 100, 0.5))  
[1] 0.05688793
```

Aqui está a probabilidade de tirar cara 39 vezes ou menos, ou 61 vezes ou mais (que é menor do que 0,05 e permite que você aceite uma H_1 bicaudal/não-direcional):

Código em R

```
> sum(dbinom(c(0:39, 61:100), 100, 0.5))  
[1] 0.0352002
```

Mais uma vez, não há necessidade de fazer isso por tentativa e erro de forma manual. Você pode usar novamente `qbinom` para obter o maior número de caras cuja probabilidade cumulativa com cada resultado ainda mais extremo não excede 0,05 - a única complicação é que, como você quer 'adicionar comprimentos de barras' nos dois lados, e os comprimentos de barra são idênticos em ambos os lados (porque as curvas na Figura 6 e na Figura 7 são simétricas), você deve obter o resultado que não exceda 0,05 quando você adicionar ambos os lados, ou seja, quando um lado não excede a 0,025. Então, você verá novamente que isso corresponde à nossa descoberta manual acima:

Código em R

```
> qbinom(0.05/2, 100, 0.5, lower.tail=FALSE)  
[1] 60
```

1.3.4.3 Extensão: distribuições de probabilidades contínuas

Nos exemplos acima, sempre lidamos com uma variável com dois níveis: Sorteio de cara versus coroa. Infelizmente, a vida não é tão simples assim. Por um lado, vimos que nossa variável categórica geralmente

envolve mais de dois níveis. Por outro lado, se as variáveis em questão forem variáveis proporcionais, então o cálculo das probabilidades em todos os estágios possíveis é impossível. Por exemplo, é impossível calcular, para um dado estímulo, todas as probabilidades de todos os tempos de reação possíveis. Por esse motivo, muitas técnicas matemáticas não calculam valores- p exatos, mas são baseadas no fato de que, com o aumento do tamanho da amostra, a distribuição das probabilidades de um evento se aproximam de distribuições matemáticas cujas funções e principais propriedades são bem conhecidas. Quatro distribuições matemáticas serão muito importantes para os testes que discutiremos nos Capítulos 4 e 5:

- a distribuição Normal padrão com escores- z (norm);
- a distribuição t (t);
- a distribuição F (f);
- a distribuição Qui-quadrado / χ^2 (chisq).

Para cada uma destas distribuições, assim como para binom visto acima, existe uma função que começa com a letra q e termina com os nomes das funções entre parênteses acima. (ex.: $qnorm$, qt , qf e $qchisq$), e uma função que começa com p e também termina com o nome das funções apresentadas acima (ex.: $pnorm$, pt , pf e $pchisq$). A primeira calcula as *funções quantile* dessas (quatro ou outras) distribuições de probabilidade, enquanto a última calcula o inverso, as chamadas *funções de distribuições cumulativas*. Podemos explicar isso com relativa facilidade com base na Figura 8, cujos painéis plotam a função de densidade da distribuição normal padrão.

Figura 8 – Função de densidade da distribuição normal padrão para $p_{uni-direcional}=0,05$

Na Figura 6, estávamos interessados em determinar o quanto um resultado pode desviar do resultado esperado de, no caso em questão, 50 cabeças e 50 caudas, sem ser significativo, onde "ser significativo" significava surgir com uma probabilidade cumulativa de menos de 0,05 do espaço de resultado total.

Nesse caso, somamos comprimentos das barras que compõem a curva da distribuição binomial (usando `dbinom`) ou identificamos diretamente o maior número de cabeças cuja probabilidade cumulativa com resultados mais extremos não excedeu a 0,05 (com `qbinom`).

Código em R

```
> sum(dbinom(58:100, 100, 0.5))  
[1] 0.06660531  
> qbinom(0.05, 100, 0.5, lower.tail=FALSE)  
[1] 58
```

Para as distribuições contínuas do tipo ilustrado na Figura 8, não há comprimentos de barras a serem adicionados, mas a noção correspondente é a área sob a curva, que é definida como 1 e da qual qualquer valor no eixo x pode cortar algo para a esquerda ou para a direita. Para tais cálculos, podemos usar novamente as funções com q e p . Por exemplo, se queremos saber qual valor de x corta 5 %, ou seja, 0,05, da área esquerda sob a curva, podemos calculá-lo das seguintes maneiras com `qnorm`:

Código em R

```
> qnorm(0.05, lower.tail=TRUE)  
[1] -1.644854  
> qnorm(1-0.95, lower.tail=TRUE)  
[1] -1.644854  
> qnorm(0.95, lower.tail=FALSE)  
[1] -1.644854  
> qnorm(1-0.05, lower.tail=FALSE)  
[1] -1.644854
```

Desta forma, a área cinza abaixo da curva no painel esquerdo da Figura 8 no intervalo $-\infty \leq x \leq -1,644854$ corresponde a 5% da área total abaixo da curva. Uma vez que a distribuição normal padrão é simétrica, o mesmo ocorre na área cinza da curva no painel direito no intervalo $1,644854 \leq x \leq \infty$

Código em R

```
> qnorm(0.95, lower.tail=TRUE)  
[1] 1.644854  
> qnorm(1-0.05, lower.tail=TRUE)  
[1] 1.644854  
> qnorm(0.05, lower.tail=FALSE)  
[1] 1.644854  
> qnorm(1-0.95, lower.tail=FALSE)  
[1] 1.644854
```

Esses são testes uni-direcionais porque examinam apenas um lado da curva, seja o esquerdo (quando `lower.tail=TRUE` no painel esquerdo) ou o direito (quando `lower.tail=FALSE` no painel direito). Para testes bi-direcionais correspondentes com o mesmo nível de significância de 0.05, você teria que proceder com `binom` e que levar em conta as duas áreas sob a curva (assim como na Figura 9), isto é, 2,5% em cada lado para chegar a 5% no total. Assim, para obter os valores do eixo x que *juntamente* cortam 5% abaixo da curva, é isto que você poderia digitar no R:

Código em R

```
> qnorm(0.025, lower.tail=TRUE)
[1] -1.959964
> qnorm(1-0.975, lower.tail=TRUE)
[1] -1.959964
> qnorm(0.975, lower.tail=FALSE)
[1] -1.959964
> qnorm(1-0.025, lower.tail=FALSE)
[1] -1.959964
```

Figura 9 – Função densidade da distribuição normal padrão para $p_{bi-direcional}=0,05$

Código em R

```
> qnorm(0.975, lower.tail=TRUE)
[1] 1.959964
> qnorm(1-0.025, lower.tail=TRUE)
[1] 1.959964
> qnorm(0.025, lower.tail=FALSE)
[1] 1.959964
> qnorm(1-0.975, lower.tail=FALSE)
[1] 1.959964
```

Novamente, você pode ver que com testes bicaudais não-direcionais você precisa de um resultado mais extremo para um resultado significativo: um valor de -1,7 é menor que -1,644854 e seria significativo em um teste unilateral (se você previu a direção negativa), mas esse mesmo valor é maior que -1,959964 e, portanto, não é pequeno o suficiente para um teste bicaudal significativo. Em suma, com as funções-q determinamos a estatística mínima unicaudal ou bicaudal que precisamos para obter um dado valor- p .

Para testes unilaterais, você normalmente usa $p = 0,05$; para testes bicaudais $p = \frac{0,05}{2} = 0,025$ de cada lado. As funções cujos nomes começam com p fazem o oposto daqueles que começam com q: com elas, você determina a que valor- p nossa estatística corresponde. As duas linhas a seguir geram os valores- p para testes unicaudais (ver Figura 8 novamente):

Código em R

```
> pnorm(-1.644854, lower.tail=TRUE)¶
[1] 0.04999996
> pnorm(1.644854, lower.tail=FALSE)¶
[1] 0.04999996
```

Para o teste bi-direcional, você naturalmente deve multiplicar a probabilidade por dois seja qual for a área sob a curva você obtiver, você deve o considerá-la em ambos os lados da curva. (Ver Figura 9 novamente):

Código em R

```
> 2*pnorm(-1.959964, lower.tail=TRUE)¶
[1] 0.05
> 2*pnorm(1.959964, lower.tail=FALSE)¶
[1] 0.05
```

As outras funções que começam com p/q funcionam da mesma maneira, mas necessitam de algumas informações adicionais: os chamados graus de liberdade. Não explicarei essa noção aqui em detalhes, mas citarei a regra de ouro de [Crawley \(2002, p. 94\)](#) que diz: “[g]raus de liberdade [gl] é igual ao tamanho da amostra, n , menos o número de parâmetros, p , [não relacionados aos outros ps acima, STG], estimados a partir dos dados”. Por exemplo, se você calcula a média de quatro números, então $gl = 3$, pois quando você quer se assegurar que você irá obter uma determinada média a partir de quatro valores, então você pode escolher livremente três valores, sendo o quarto valor pré-determinado. Imagine que você queira obter a média 8, então os primeiros três valores podem variar livremente e ser, por exemplo, 1, 2 e 3, mas então o último número tem que ser 26. Graus de liberdade são a forma pela qual os tamanhos das amostras e a quantidade de informação que você deseja obter da amostra são integrados no teste de significância.

Os testes paramétricos que são baseados nas distribuições mencionadas anteriormente são geralmente um pouco mais fáceis de calcular (embora geralmente esse não seja mais um ponto importante, dado o poder de computação dos atuais computadores de mesa) e são mais poderosos, mas apresentam o problema potencial aludido acima. Uma vez que esses testes são apenas estimativas dos valores- p reais calculados com base nas equações que definem os valores $z / t / F / \chi^2$, a precisão desses testes depende de quão bem essas equações representam a distribuição dos dados. No exemplo anterior, a distribuição Binomial da Figura 4 e Figura 5 e a distribuição normal da Figura 8 e Figura 9 são extremamente semelhantes, mas isso pode ser bem diferente em outras ocasiões. Assim, testes paramétricos fazem algumas pressuposições com relação à distribuição dos dados – a mais comum sendo o pressuposto da distribuição normal – para que você possa utilizar tais testes apenas se seus dados satisfizerem essas pressuposições. Caso contrário, você deve utilizar o que chamamos de testes não-paramétricos, ou utilizar algum teste exato (como fizemos com a moeda jogada acima) ou um teste de permutação ou outros métodos de reamostragem. Para quase todos os testes apresentados nos Capítulos 4 e 5 a seguir, listarei todas as pressuposições que você precisará testar antes de aplicar o teste, explicarei o teste propriamente dito com a computação do valor- p , e ilustrarei como você apresentaria os resultados na terceira parte (resultados) da versão escrita de seu estudo. Já posso adiantar que você deve sempre fornecer os tamanhos das amostras, o efeito obtido (tais como média, percentuais, diferença entre médias, etc), o

nome do teste que você utilizou, seus parâmetros estatísticos, o valor- p , e sua decisão (à favor ou contra a H_1). A interpretação desses achados deverá então ser discutida na quarta e última seção do seu estudo.

Recomendação para estudo/leitura

[Good e Hardin \(2012, Cap. 1, 2 e 3\)](#) para dicas interessantes e [Good e Hardin \(2012, Cap. 8\)](#) para ver as informações que você deve fornecer na seção de metodologia e resultados.

Aviso Importante

Não caia na tentação de utilizar testes paramétricos quando as pressuposições não são atendidas. O que você ganha em fazer o teste errado, publicar resultados errados e depois ser citado por erros e problemas metodológicos no seu estudo?

1.4 O desenho de um experimento fatorial: introdução

Nesta seção, vamos abordar algumas regras fundamentais para o planejamento de experimentos¹². Provavelmente, a ideia principal desta seção é noção de *token set* ([Coward \(1997\)](#), 1997). Vamos distinguir dois tipos de conjunto de exemplares, *token* esquemáticos e tokens concretos. Basicamente, um *token* esquemático é uma representação tabular de todas as condições experimentais. Para explicar isso de maneira mais clara, vamos retornar ao nosso exemplo de colocação da partícula.

Imagine que você queira investigar *particle placement* não só com base em estudos de corpora, mas também com base em dados experimentais. Por exemplo, você pode estar interessado em saber como que falantes nativos do Inglês julgam a aceitabilidade de frases (a variável dependente *Aceitabilidade*) que diferem no que diz respeito à escolha da construção (a primeira variável independente: Construção: VPO vs. VOP) da classe gramatical do núcleo do objeto direto (a segunda variável independente OBJPOS: PRONOMINAL vs. LEXICAL)¹³. Uma vez que há dois níveis para cada uma das variáveis independentes, haverá $2 \times 2 = 4$ condições experimentais. Esse conjunto de condições experimentais é o *token* esquemático, que é representado de duas maneiras distintas na Tabela 11 e na Tabela 12. Para um experimento de verdade, você deve criar estímulos – um *token* concreto que instancia os níveis das variáveis apresentadas no *token* esquemático.

Tabela 11 – *Token set* esquemático para as variáveis CONSTRUÇÃO $\times$ OBJPOS 1

	OBJPOS: pronominal	OBJPOS: lexical
CONSTRUÇÃO: VPO	V Part pron. SN _{obj. dir.}	V Part lexical SN _{obj. dir.}
CONSTRUÇÃO: VOP	V pron. SN _{obj. dir.}	Part V lexical SN _{obj. dir.} Part

Tabela 12 – *Token set* esquemático para as variáveis CONSTRUÇÃO $\times$ OBJPOS 2

Condição Experimental	CONSTRUÇÃO	OBJPOS 2
1	VPO	pronominal
2	VPO	lexical
3	VOP	pronominal
4	VOP	lexical

¹² Irei apenas considerar o tipo mais simples e conservador de *design* experimental, *designs* fatoriais, nos quais cada nível variável é combinado com todos os outros níveis variáveis

¹³ Para clareza na exposição, vamos assumir apenas dois níveis para OBJPOS.

No entanto, tanto a construção dos *token sets* concretos e a apresentação do estímulo propriamente são governadas por uma variedade de regras que têm a finalidade de diminuir ao máximo as fontes de ruído (ou perturbação) nos dados. Três fontes de ruído podem ser destacadas:

- *saber do que se trata o experimento*: é importante que os participantes não saibam do que o experimento se trata antes e durante a sua execução (depois de sua execução, você deve avisá-lo do que se tratava o experimento). Esse ponto é importante uma vez que as respostas do participante podem ser enviesadas, ou seja, podem ser as que ele acha que são socialmente aceitas, ou aquelas que irão ajudar o pesquisador.
- *efeitos experimentais indesejados*: você deve se certificar de que as respostas dos seus participantes não estão sendo influenciadas pela ordem de apresentação dos estímulos (um fenômeno conhecido como habituação). No campo de julgamento de aceitabilidade, por exemplo, Nagata (1987), Nagata (1989) mostrou que os julgamentos mudam depois da exposição prolongada a estímulos para julgamento. Você certamente não quer esse tipo de efeito.
- *avaliação dos resultados*: você deve se certificar de que as respostas dos participantes possam ser interpretadas sem qualquer tipo de ambiguidade. Mesmo que tenha um grande número de participantes competentes e bem dispostos, se seu *design* não for apropriado para a análise dos dados, seus dados serão inúteis.

Para evitar os problemas mencionados acima, você deve levar em consideração as regras de (4) a (12) apresentadas abaixo. Aqui está a primeira (4):

(4) Os estímulos para cada *token* concreto devem se diferenciar apenas com relação à combinação dos níveis das variáveis sob investigação (é importante que essa seja a única diferença entre os estímulos).

Considere a Tabela 13, por exemplo. Na Tabela 13, os estímulos se diferenciam apenas com relação às duas variáveis independentes. Se não fosse assim e você encontrasse uma diferença no nível de aceitabilidade de duas das frases (*John picked up it* e *John brought it back*, por exemplo), você não poderia atribuir essa diferença à diferença de construção (que é o objetivo principal do experimento) e sim à uma diferença no tipo de verbo (um achado interessante, mas que não é o foco do estudo). A regra (4) tem haver com o ponto sobre “avaliação dos resultados”.

Tabela 13 – Um *token* concreto para as variáveis CONSTRUÇÃO × OBJPOS 1

	ObjPOS: pronominal	ObjPOS: lexical
Construção: VPO	John picked up it.	John picked up the keys.
Construção: VOP	John picked it up.	John picked the keys up.

Quando for criar o seu *token* concreto, é importante que tente controlar as variáveis das quais você não tem interesse de investigar e as que dificultariam as suas análises posteriores com relação as variáveis que você tem interesse. No caso do nosso exemplo, a escolha dos verbos e dos objetos diretos pode ser importante. Sabe-se, por exemplo, que a colocação de partícula está relacionada com a concretude do referente contido no objeto direto. Existem várias formas de considerar tais fontes de variação na construção dos estímulos. Uma maneira, por exemplo, é certificar-se de que 50% dos objetos são abstratos e 50% são concretos para cada uma das condições experimentais no *token set* esquemático (como se você estivesse introduzindo uma outra variável independente). Uma outra forma é utilizar apenas objetos concretos ou apenas objetos abstratos, o que implicaria, obviamente, que qualquer que seja o resultado que você encontrar no seu experimento, ele poderia ser generalizado apenas para a classe de objetos que você escolher utilizar.

Recomendação para estudo/leitura

Good e Hardin (2012, p. 31) e Good (2005, Cap. 5)

(5) Você deve utilizar mais de um *token set* concreto. Na verdade, o ideal é que o número de conjuntos de *tokens* seja correspondente ao número de combinações dos níveis das suas variáveis (ou múltiplos desse número).

Um motivo para a regra (5) é que, se você utilizasse apenas um conjunto de *tokens* concretos da Tabela 13, então a partir de um ponto de vista conservador você poderia apenas generalizar seus resultados para frases que contêm o verbo frasal *pick up* e os objetos *it* e *the book*, o que não seria provavelmente um estudo muito interessante. Assim, o primeiro motivo para (5) novamente está relacionado à questão de 'avaliação dos resultados', e a solução é a criação de vários conjuntos de *token* concretos, com diferentes verbos e diferentes objetos tais como os apresentados nas Tabela 14 e Tabela 15, que também devem estar de acordo com a regra (4).

Tabela 14 – Um conjunto de *tokens* concretos para CONSTRUÇÃO × OBJPOS 2

	OBJPOS: PRONOMINAL	OBJPOS: LEXICAL
CONSTRUÇÃO: VPO	Mary brought back him.	Mary brought back his dad.
CONSTRUÇÃO: VOP	Mary brought him back.	Mary brought his dad back.

Tabela 15 – Um *token* concreto para as variáveis CONSTRUÇÃO × OBJPOS 3

	OBJPOS: PRONOMINAL	OBJPOS: LEXICAL
CONSTRUÇÃO: VPO	I eked out it.	I eked out my living.
CONSTRUÇÃO: VOP	I eked it out.	I eked my living out.

A segunda razão para (5) é que se você apenas usou o *token* concreto definido na Tabela 13, então os sujeitos provavelmente serão capazes de adivinhar o propósito do experimento imediatamente: uma vez que nosso conjunto de *tokens* tinha que se conformar com (4), o sujeito pode identificar as combinações de nível variável relevantes rapidamente porque essas são as únicas coisas que diferenciam as sentenças. Isso imediatamente nos leva à seguinte regra:

(6) Cada participante deve ver apenas um item de cada conjunto de *tokens* concretos.

Como eu acabei de mencionar, se você não seguir 0, os participantes podem descobrir o propósito do experimento, uma vez que estarão expostos à variação existente em um conjunto de *tokens* concreto: a única diferença entre *John picked up it* e *John picked it up* é a escolha da construção. Assim, quando um sujeito X vê a combinação de nível variável (CONSTRUÇÃO: VPO × OBJPOS: PRONOMINAL) na forma de *John picked up it*, então os outros itens experimentais da Tabela 13 devem ser apresentados a outros sujeitos. Nesse sentido, tanto (5) quanto (6) também preocupam-se com o fator 'conhecimento sobre o que é o experimento'.

(7) Cada participante deve ser exposto a todas as combinações dos níveis das variáveis.

A regra (7) está ligada aos fatores 'efeitos experimentais indesejados' e 'avaliação dos resultados'. Primeiramente, se todos os itens experimentais que você apresentar ao seu participante apenas representarem uma combinação de nível de variável, você pode atrapalhar seus resultados com o efeito de habituação. Isso poderia ser obviamente evitado se você adicionasse uma variável à sua análise que menciona, para cada apresentação de uma condição experimental, com que frequência a mesma já foi apresentada. Em segundo lugar, se você apresenta para um participante uma mesma combinação de nível de variável com mais frequência que outra combinação, os resultados que você encontrar podem ser fruto das diferentes frequências de apresentação, e não por causa dos efeitos das variáveis sob investigação.

(8) Cada participante deve ser exposto a todas as combinações mais de uma vez e com a mesma frequência.

(9) Cada item experimental deve ser apresentado para mais de um participante e para o mesmo número de participantes.

Essas regras são motivadas pelo fator 'avaliação dos resultados'. Você pode entender qual é o seu propósito se você pensar sobre o que acontece quando você tenta interpretar uma reação muito incomum

por um sujeito a um estímulo. Por um lado, essa reação poderia significar que o item em si é incomum de alguma forma no sentido de que cada sujeito reagiria inusitadamente a ele - mas você não pode testar isso se esse item também não for dado a outros sujeitos, e esta é a razão para a regra em (9). Por outro lado, a reação inusitada poderia significar que somente esse indivíduo em particular reage inusitadamente a essa combinação de nível variável no sentido de que o mesmo sujeito reagiria mais “normalmente” a outros itens instanciando a mesma combinação de nível variável - mas você não pode testar isso se esse sujeito não vê outros itens com a mesma combinação de nível variável, e esta é a razão para (8).

(10) Os itens experimentais devem ser misturados a itens distratores; o número de distratores deve ser, no mínimo, igual ao número de itens experimentais por participante. O ideal é que o número de distratores seja duas ou três vezes o número de itens experimentais por participante.

Essa regra (10) está ligada ao fator ‘saber do que se trata o experimento’: você não quer que seus participantes adivinhem do que se trata o experimento (ou até mesmo que eles achem que sabem do que se trata o experimento) para que seus resultados não sejam enviesados.¹⁴

Um fator adicional que pode enviesar seus resultados é a ordem de apresentação dos itens experimentais e distratores. Para minimizar esse efeito, você deve levar em consideração as duas regras finais:

(11) A ordem de apresentação dos itens experimentais e distratores deve ser pseudo-aleatória.

(12) A ordem de apresentação dos itens experimentais e distratores deve ser pseudo-aleatorizada de maneira distinta para cada participante.

A regra (11) requer que a ordem de apresentação dos itens (experimentais e distratores) seja aleatória, porém essa aleatorização deve ser corrigida – por isso pseudo-aleatória – de acordo com os itens abaixo:

- o primeiro item nunca deve ser um item experimental, mas sim um distrator;
- itens experimentais não podem ser apresentados consecutivamente;
- itens experimentais que representam a mesma combinação não devem ser apresentados consecutivamente. Por exemplo, após a apresentação de *John picked it up*, você não deve apresentar o item *Mary brought him back*.

A regra (12) quer dizer que a forma de aleatorização deve ser diferente para cada participante. Assim, os efeitos que você encontrar nunca poderão ser atribuídos a alguma ordem sistemática na apresentação dos itens: cada participante é exposto a uma ordem diferente dos itens (experimentais e distratores). Assim, tanto a regra (11) quanto a regra (12) está relacionada ao fator ‘avaliação dos resultados’. (Esta reordenação de estímulos pode ser bastante tediosa, especialmente quando o experimento envolve muitos itens de teste e assuntos, e é por isso que, uma vez que você for mais proficiente com o R, pode ser útil escrever uma função chamada, digamos, `stimulus.randomizer` para fazer isso para você, como eu já faço.)

Apenas depois que todos esses passos forem completados é que você está pronto para aplicar seus questionários ou recrutar participantes para participar do experimento. Vale a pena mencionar mais uma vez que você deve descrever, na seção metodologia, a maneira como preparou seu experimento ou organizou seu estudo. Vamos analisar um outro exemplo na seção seguinte, para deixar tudo mais claro.

Uma observação final sobre isso antes de examinarmos outro exemplo. Eu sei por experiência própria que a seção anterior pode ter um efeito um tanto quanto desanimador. Especialmente os novatos a lêem e pensam “como é que eu vou ser capaz de criar um experimento para o meu projeto se eu tiver que fazer tudo isso? (Eu nem conheço o meu software de planilha bem o suficiente ainda ...)” E é verdade: eu mesmo ainda preciso de um longo tempo antes que uma planilha para um experimento meu fique do jeito esperado. Mas se você não passar pelo que à primeira vista parece uma terrível provação, seus resultados podem muito bem ser, vamos dizer a verdade, uma porcaria! Pergunte a si mesmo o que é mais

¹⁴ Em alguns experimentos em psicologia, nem mesmo a pessoa que aplica os testes sabe do propósito do experimento. Esse procedimento assegura que nem mesmo o assistente de pesquisa, inconscientemente, facilite as respostas do participante. Uma outra forma de controlar isso é a produção de instruções padronizadas, em forma de vídeo, por exemplo.

desencorajador: gastar talvez vários dias para obter a planilha correta, ou gastar talvez várias semanas fazendo um experimento mais simples e com ter resultados inutilizáveis ...

Aviso Importante

Você deve estar sempre preparado para o fato de que nem todo participante responde a todas as perguntas que você fornece. Assim, você deve planejar seu experimento/estudo de forma que você recrute o máximo de participantes possível. Como mencionamos anteriormente, mesmo que o participante não responda a uma ou duas perguntas do experimento, você precisa incluir esse “dado” na sua planilha (basta marca-lo com a notação “NA”). É importante analisar se os dados faltantes formam algum padrão interessante – por exemplo, mostrar que 90% dos dados faltantes são de apenas uma das combinações experimentais, ou que 90% dos dados faltantes foram de dois ou três participantes apenas (em um total de 60 participantes, por exemplo)

1.5 O design de um experimento fatorial: um outro exemplo

Imagine que você esteja interessado em investigar quais variáveis determinam a quantidade de elementos aos quais o quantificador da língua inglesa *some* se refere. Considere as frases em (13):

Exemplo (13)

- a $[_{SN} \text{ some balls } [_{SP} \text{ in front of } [_{SN} \text{ the cat }]]]$
- b $[_{SN} \text{ some balls } [_{SP} \text{ in front of } [_{SN} \text{ the table }]]]$
- c $[_{SN} \text{ some cars } [_{SP} \text{ in front of } [_{SN} \text{ the building }]]]$

Assim, a pergunta que se faz é: será que a quantidade de bolas em *some some balls in front of the cat* é a mesma quantidade de bolas em *some balls in front of the table*? Ou: a quantidade de bolas a que a frase *some balls in front of the table* se refere é a mesma quantidade de carros em *some cars in front of the building*? O que – ou mais precisamente, qual a quantidade – o quantificador *some* do inglês expressa? Sua revisão da literatura relevante mostrou que pelo menos duas variáveis parecem influenciar a quantidade a que a palavra *some* se refere:

- OBJETO: o tamanho (físico) do objeto a que a palavra se refere: pequeno (ex.: bola) versus grande (ex.: carro);
- PONTOREF: o tamanho do objeto introduz o como ponto de referência pelo sintagma preposicional: pequeno (ex.: gato) versus grande (ex.: prédio).¹⁵

Obviamente, o estudo da palavra *some* com essas duas variáveis resulta em um conjunto de *tokens* esquemático com 4 combinações de níveis variáveis, conforme apresentado na Tabela 16.

Tabela 16 – *Token sets* (esquemáticos + concretos) para as variáveis OBJETO × PONTOREF

	PONTOREF: PEQUENO	PONTOREF: GRANDE
OBJETO:PEQUENO	pequeno + pequeno: <i>some dogs next to a cat</i>	pequeno + grande: <i>some dogs next to a car</i>
OBJETO:GRANDE	grande + pequeno: <i>some cars next to a cat</i>	grande + grande: <i>some cars next to a fence</i>

As hipóteses não-direcional para esse estudo são:

¹⁵ Não iremos discutir aqui o que caracteriza grande ou pequeno. No estudo original, essa definição foi baseada em um estudo piloto anterior ao estudo principal.

H_0 : A estimativa média da quantidade de elementos a que a palavra “some” se refere não depende do tamanho dos elementos (OBJETO: PEQUENO VERSUS GRANDE) e não depende do tamanho do ponto de referência (PONTOREF: PEQUENO VERSUS GRANDE) nas frases em que os participantes tiverem que estimar a quantidade de elementos:

$$média_{pequeno+pequeno} = média_{pequeno+grande} = média_{grande+pequeno} = média_{grande+grande}.$$

H_1 : A estimativa média da quantidade de elementos que a palavra some caracteriza depende do tamanho dos elementos (OBJETO: PEQUENO VERSUS GRANDE) e/ou depende do tamanho do ponto de referência (PONTOREF: PEQUENO VERSUS GRANDE) e/ou de algum efeito conjunto dos dois: há pelo menos um sinal de diferença ($\neq$) na equação apresentada acima.

Agora, imagine que você quer testar essa hipótese utilizando um questionário: os participantes receberão as frases da Tabela 16 e deverão responder quantos elementos eles acham que o falante das frases se referia com a palavra “some” – por exemplo, na frase “some dogs next to the car”, o participante deveria dizer quantos cachorros haviam na cena. Uma vez que existe quatro combinações, você irá precisar de pelo menos quatro *token set* concretos – regra (5). Esses *token set* concretos devem ser criados de acordo com a regra (4). De acordo com as regras (6) e (7), você vai precisar de pelo menos quatro participantes: você não pode ter menos que isso, por que senão um participante teria que ver mais de estímulo de um dos *token set* concretos. Você pode então apresentar os estímulos para os participantes de maneira rotativa. Você pode ver o resultado disso no documento <Fase 1> do arquivo <_inputfiles01-5_ExperimentalDesign.ods>. Assim como todos os arquivos, este foi encontrado no site parceiro (ver início do Capítulo 2). Os estímulos experimentais reais são representados apenas esquematicamente como uma combinação de identificação única do número do conjunto de *tokens* concretos e dos níveis variáveis das duas variáveis independentes (na coluna E).

Como você pode ver facilmente na tabela à direita, a rotação garante que cada participante veja cada uma das combinações apenas uma vez e cada uma delas com um item de um *token set* concreto distinto. No entanto, você sabe que vai precisar de mais do que isso, pois na <Fase 1> cada participante vê cada uma das combinações apenas uma vez – o que viola a regra (8) e cada item experimental é visto por apenas um participante – o que viola a regra (9). Logo, você deve re-utilizar os itens da <Fase 1>, mas precisa colocá-los em uma ordem diferente, de maneira que itens experimentais não co-ocorram com os mesmos itens experimentais (você pode fazer isso fazendo uma rotação diferente com os participantes). Um resultado possível pode ser visto na planilha <Fase 2>.

A organização na <Fase 2> ainda não está em conformidade com a regra (8). Você deve apresentar alguns itens experimentais para o participante 1, por exemplo, mas não pode utilizar os itens experimentais que já existem sem violar a regra (6). Para isso, você precisa de mais quatro *token set* concretos, que serão distribuídos entre os participantes, assim como foi feito anteriormente. O resultado pode ser visto na <Fase 3>. Como pode ser visto na tabela do lado direito, cada item experimental é visto por dois participantes (o total da linha), e nas colunas você pode ver que cada participante vê cada uma das combinações duas vezes com itens experimentais diferentes.

Agora, como cada participante vê oito itens experimentais, você deve criar o número suficiente de distratores. Para o nosso exemplo, vamos utilizar dois distratores para cada item experimental, totalizando 16 distratores (note que os distratores serão apresentados para todos os participantes, não sendo necessária a criação de $8 \times 16 = 128$ distratores. Veja na Fase 4 da planilha, onde os distratores foram incluídos na parte inferior da tabela.

Em seguida, você deve re-ordenar todos os itens - experimentais e distratores - para cada um dos participantes. Para isso, você pode adicionar uma coluna chamada “RND” (de aleatorização), que irá conter números aleatórios entre 0 e 1 (você pode fazer isso no R ou digitando na planilha “=RAND()” (sem as aspas) em uma célula do LibreOffice Calc e, em seguida, clicar duas vezes no pequeno quadrado preto no canto inferior direito que você vê quando você clica nessa célula uma vez, o que irá preencher todas as células abaixo com números aleatórios.

Como próximo passo, você deve classificar toda a planilha (i) de acordo com a coluna “SUBJ” e depois (ii) de acordo com a coluna “RAND”. No entanto, há um detalhe importante primeiro: realce essa coluna

inteira, copie o conteúdo para a área de transferência, vá para Editar: Colar especial ... e escolha colar apenas o texto e os números. Isto garantirá que os números aleatórios não são re-calculados depois de qualquer coisa que você fizer com a planilha. Em seguida, classifique como mencionado de forma que todos os itens de um assunto sejam colocados juntos, e que dentro de cada assunto a ordem dos itens seja aleatória. Isto é requerido por (12) e representado em <Fase 5>.

Se você observar os itens na <Fase 5>, você irá notar que alguns elementos ainda precisam ser trocados de ordem: as setas vermelhas na coluna H mostram as sequências que podem ser problemáticas. Para resolver isso, você pode escolher, arbitrariamente, um dos distratores e trocar um item experimental da sequência problemática por esse distrator. Os resultados dessas trocas podem ser vistos na <Fase 6>, onde as setas verdes mostram as correções. Se tivéssemos usado estímulos reais, você poderia criar uma folha de rosto com instruções para os sujeitos e alguns exemplos (que, no caso de, digamos, julgamentos idealmente cobririam os extremos dos julgamentos possíveis!), colar os estímulos experimentais na(s) página(s) seguinte(s), e distribuir os questionários. Em seguida, quando você recebesse as respostas, você as inseriria em <Fase 7> e passaria para a análise estatística dos dados. Por exemplo, para avaliar esse experimento, você teria que computar uma variedade de médias:

- as médias para os dois níveis da variável Objeto ($média_{objeto: pequeno}$ e $média_{objeto: grande}$);
- as médias para os dois níveis da variável PontoRef ($média_{PontoRef: pequeno}$ e $média_{PontoRef: grande}$);
- as quatro médias das quatro combinações/interações entre as variáveis Objeto and PontoRef.

Discutiremos o método que é usado para testar essas médias em relação a diferenças significativas - um modelo linear - na Seção 5.2. Agora você deve fazer os exercícios referentes ao Capítulo 1 (os quais você pode encontrar no site)

Fundamentos Básicos do R

*Quando dizemos que um historiador ou um linguista é 'inumerado'
Queremos dizer que ele não pode sequer começar a entender sobre o que
cientistas e matemáticos estão falando
Oxford English Dictionary, 2a ed., 1989, s.v. numeracy. (Citado de Keen (2010, p. 4))*

2.1 Introdução e instalação

Neste capítulo, iremos apresentar os princípios básicos do R que nos permitem carregar, processar e armazenar os dados, bem como executar algumas análises estatísticas simples. Assim, o presente capítulo é uma preparação para a aplicação do software nos capítulos seguintes. Vamos começar com o primeiro passo: a instalação do R.

1. O site principal do programa R é <<http://www.r-project.org/>>. A partir desse site, você pode acessar o CRAN <<http://cran.r-project.org/mirrors.html>>. Clique no mirror Austria e então no link do seu sistema operacional;
2. para o Windows, você deverá clicar no link “base” e então no link para fazer o download do programa de instalação; para o sistema operacional Mac OS X, você será diretamente direcionado à página que contém o link para o arquivo .pkg; para o sistema operacional Linux, você deve escolher sua distribuição, talvez a versão da distribuição, e então o arquivo relevante ou, mais convenientemente, você pode conseguir instalar o R, e muitos pacotes usados frequentemente, usando um gerenciador de pacotes, como Synaptic ou Muon;
3. então, você deve executar o instalador;
4. clique duas vezes no ícone do R na área de trabalho, ou no ícone no menu Iniciar ou no atalho na barra de ferramentas.

Você já pode agora começar a utilizar o programa R. No entanto, o R tem muito mais a oferecer. Uma vez que o R é um programa gratuito com sua fonte aberta ao público, muitas pessoas têm desenvolvido o que chamamos de pacotes para o R. Esses pacotes funcionam como se fossem adicionais que você incorpora ao programa, de forma que você passa a ter acesso a comandos (ou funções) que não fazem parte da configuração inicial/padrão do R.

5. Para instalar um pacote, assim que abrir o programa R, digite o seguinte comando na janela principal: `install.packages()` e escolha um mirror; recomendamos sempre utilizar o *mirror* Austria;

6. Escolha todos os pacotes que acha que vai precisar; se você tiver uma boa conexão de internet – banda larga de preferência, você pode escolher todos os pacotes. Recomendamos, inicialmente, os seguintes pacotes: `amap`, `aod`, `car`, `cluster`, `effects`, `Hmisc`, `lattice`, `qcc`, `plotrix`, `rms`, `rpart` `vcd`. (Você pode também digitar, digamos, `install.packages("car")` na tela de console para instalar o referido pacote e idealmente fazer isso com direitos de administrador; isto é, em Ubuntu, por exemplo, inicie o R com `sudo R`. Em sistemas Linux, você às vezes também precisará de arquivos adicionais tais como `gfortran`, que você pode precisar instalar separadamente)

O próximo passo agora é baixar os arquivos com exemplos de arquivos, todos os códigos e exercícios com chave de resposta para o seu disco rígido. Idealmente, você deve criar no seu computador uma pasta para todos os arquivos do livro, como, por exemplo, `<_sflwr/>` no seu disco rígido (for `statistics for linguists with R`). Em seguida, faça o download de todos os arquivos disponíveis no site parceiro desta edição do livro (`<http://tinyurl.com/StatForLingWithR>`) e os salve/descompacte em:

- `<_sflwr/_inputfiles>`: nessa pasta você irá salvar todos os arquivos de input: textos, dados, planilhas, exercícios, etc. Para descompactar os arquivos, você irá precisar da senha “hamster_R2”;
- `<_sflwr/_outputfiles>`: nessa pasta você irá salvar todos os arquivos de output dos Capítulos 2 ao 5. Para descompactar os arquivos, você irá precisar da senha “squi_R2rel”;
- `<_sflwr/_scripts>`: nessa pasta você irá salvar os arquivos com códigos, exercícios e chave de resposta. Para descompactar os arquivos, você irá precisar da senha “otte_R2”.

(A propósito, estamos utilizando barras regulares, pois você poderá utilizá-las no R, também são mais fáceis do que barras invertidas). O site do livro irá conter um arquivo chamado Errata. E por último, recomendamos que todos utilizem um editor de texto que realce a sintaxe dos comandos do R ou um ambiente de desenvolvimento integrado (IDE em inglês) Se você usar um editor de texto, eu recomendo Notepad++ e Geany para usuários Windows ou use o Notepad++ com o Wine para usuários Linux. Provavelmente a melhor opção, contudo deve ser o RStudio (`<http://www.rstudio.org/>`), verdadeiramente uma IDE de código livre para o R, no qual pode-se editar facilmente os códigos do R, enviando o código da janela do editor para o console usando a combinação das teclas Ctrl+ENTER, plotar histogramas, e muitas outras coisas. Você deve assistir ao screencast no site RStudio.

Depois disso, você poderá ver todos os scripts na pasta `<_scripts>` com a sintaxe realçada, o que facilita a compreensão dos comandos. Recomendamos também que você escreva todos os scripts que forem maior que 2 ou 3 linhas utilizando esses programas e só depois cole-os na janela principal do R. O realce da sintaxe irá ajudá-lo e evitar que cometa erros que inviabilizam a execução das funções e comandos.

O R não é apenas um programa estatístico – ele é também uma linguagem de programação e um ambiente que tem pelo menos algumas semelhanças com as linguagens Perl, Python ou Julia. A gama de aplicações do programa é enorme uma vez que o programa oferece as funcionalidades de uma planilha eletrônica, de um programa estatístico, de uma linguagem de programação, de funções de banco de dados, etc. Esta introdução à estatística, no entanto, é focada principalmente:

- nas funções que geram e processam estruturas de dados simples no R, e
- nas funções aplicadas distribuição de probabilidades, testes estatísticos e avaliação gráfica.

Logo, infelizmente não poderemos lidar com estruturas de dados mais complexas e com os outros aspectos interessantes que o R, enquanto linguagem de programação pode oferecer. Além disto, não utilizarei sempre a maneira mais elegante e simples de executar uma tarefa específica, mas sim a maneira mais útil do ponto de vista pedagógico e metodológico (assim podemos destacar as principais semelhanças entre funções e abordagens distintas). Portanto, o presente volume não oferece uma introdução geral ao R, e eu sugiro que você veja as recomendações para estudos adicionais e a seção de referência para livros introdutórios sobre o R.

Vamos agora falar de algumas convenções tipográficas e outras convenções. Toda vez que nos referimos a uma pasta, arquivo ou página da internet, iremos utilizar “<” e “>” (ex.: <_inputfiles/04-1-1-1_tense-aspect.csv>). Para o nome do arquivo, o número que se encontra antes do *underscore* refere-se à seção do livro onde o arquivo é utilizado. Textos que você deve digitar diretamente no R será formatado da seguinte maneira `mean(c(1, 2, 3))`. O caracter “`¶`” instrui você a pressionar ENTER (eu mostro esses caracteres aqui porque eles podem ser importantes para mostrar a estrutura exata de uma linha e porque o espaço em branco faz uma grande diferença em sequências de caracteres; os arquivos de código, claro, não os incluem de forma visível, a menos que você defina o seu editor de texto para exibi-los). O código será geralmente dado em blocos cinzentos de várias linhas como esta:

Código em R

```
> a=c(1, 2, 3)¶
> mean(a)¶
[1] 2
```

Outro ponto importante: não digite os dois caracteres `>`. Eles servem apenas para diferenciar o seu input do output do R. Você também verá algumas vezes linhas que começam com o símbolo “+”, o qual você também não deve digitar, indicando que o R ainda está esperando mais um *input*¹ antes de começar a executar uma função. Por exemplo, se você digitar `2-¶`, é assim que a interface do programa R aparecerá:

Código em R

```
> 2-¶
+
```

O R está esperando você completar a subtração. Quando você digitar o número que deseja subtrair e pressionar ENTER, a função será executada corretamente.

Código em R

```
+3¶
[1] -1
```

Um outro exemplo: imagine que você queira carregar no R o pacote `corpora` para que tenha acesso às funções que os linguistas computacionais Marco Baroni e Stefan Evert contribuíram para a comunidade. Para carregar esse pacote – ou seja, colocá-lo disponível para utilização no R – você deve utilizar o comando `library(corpora)`. (Importante: você deve instalar o pacote antes de carregar o pacote, como eu expliquei acima). No entanto, se você esquecer de fechar os parênteses do comando, o R irá esperar que você complete o input:

Código em R

```
>library(corpora¶
+ )¶
>
```

Infelizmente, nem sempre o R será bonzinho assim. A propósito, se você comete um erro no R, você usualmente precisa corrigir apenas uma coisa na linha de comando. Portanto, ao invés de digitar toda a linha de comando novamente, pressione a seta do teclado (a seta para cima) para voltar à linha que você

¹ o resto de um comando

deseja alterar e execute o comando novamente; Além disso, você não precisa mover o cursor até o final da linha para então pressionar ENTER.

Os arquivos de corpus ou tabelas/estruturas de dado serão representados como na Figura 10, onde os símbolos “→” e “¶” representam “teclar TAB” e “teclar ENTER”, respectivamente. Menus, sub-menus e comandos em sub-menus dentro de aplicações serão dados em itálico e em aspas duplas, e níveis hierárquicos dentro de menus de aplicações serão indicados com dois pontos. Por exemplo, para abrir um documento no LibreOffice Writer, você deve clicar nos menus Arquivo: Abrir...

POS	→	TOKENFREQ	→	TYPEFREQ	→	CLASS¶
adj	→	421	→	271	→	open¶
adv	→	337	→	103	→	open¶
n	→	1411	→	735	→	open¶
conj	→	458	→	18	→	closed¶
prep	→	455	→	37	→	closed¶

Figura 10 – Formato representacional de arquivos de corpora e *data frame*

2.2 Funções e seus argumentos

Como você deve se lembrar das suas aulas de matemática na escola, geralmente utilizamos letras, não números, para representar variáveis que contêm números. Nas aulas de Álgebra, por exemplo, você deve se lembrar que, nas lições sobre equações, você tinha que descobrir os valores que as variáveis a e b representam nas seguintes equações (aqui $a = \frac{23}{7}$ e $b = \frac{20}{7}$):

$$a + 2b = 9 \text{ e}$$

$$3a - b = 7$$

No R, você consegue resolver tais equações, também, mas uma vez que o R é uma ferramenta muito poderosa, variáveis como a e b podem representar elementos multidimensionais enormes, chamados aqui de *estruturas de dados*. Neste capítulo, vamos lidar com as estruturas de dados que são mais importantes para análises estatísticas. Tais estruturas de dados podem ser digitadas diretamente no R, ou, como ocorre na maioria das vezes, ser importadas à partir de outros arquivos. Apresentarei os dois modos de inserção de dados, mas a maioria dos exemplos abaixo pressupõe que os dados estejam disponíveis em na forma de um arquivo de texto delimitado por tabulações que tem a estrutura discutida no capítulo anterior e que foi criado em um editor de texto ou em um editor de planilha como o LibreOffice Calc. Nas seções a seguir, eu vou explicar

- como criar estruturas de dados no R;
- como carregar estruturas de dados no R e salvá-las no programa R;
- como editar estruturas de dados no R.

Um dos aspectos mais importantes sobre o R é saber como dizer ao programa para fazer alguma outra coisa além dos cálculos simples que executamos acima. Um comando no R basicamente consiste de dois elementos: uma *função* e, entre parênteses, os *argumentos* dessa função. A função pode ser compreendida como uma instrução para executar alguma tarefa, e os argumentos da função representam (i) a que elementos a função deve ser aplicada e (ii) como a instrução deve ser aplicada. Vamos ver dois exemplos simples que você aprendeu na escola. Imagine que você queira calcular a raiz quadrada de 5 - sem simplesmente digitar a instrução $5^{0.5}$, isto é - você precisa saber o nome da função, bem como quantos e quais argumentos ela requer. Bem, o nome da função que calcula a raiz quadrada é `sqrt` (do inglês *square root*) e ela precisa de apenas um argumento, chamado no R de x , ou seja, o número do qual pretende saber a raiz quadrada. O comando no R é:

Código em R

```
> sqrt(x=5)
[1] 2.236068
```

Note que o R apenas mostra o resultado da função que você pediu para executar, mas não o armazena. Se você deseja armazenar o resultado de uma função (na verdade, isso funciona para qualquer estrutura de dados), você deve utilizar o símbolo `<-` (um sinal de menor-que e um sinal de subtração). Você pode também utilizar o sinal de igual `=` para o mesmo propósito. No nosso exemplo, vamos armazenar o exemplo e chamá-lo de `a`. Note que no R, caracteres maiúsculos e minúsculos são diferentes. Outro ponto importante sobre os nomes dados para sua estrutura de dados é: você pode utilizar letras, números e pontos, mas o nome deve necessariamente começar com uma letra ou um ponto final (ex.: `meu.resultado` ou `meu_resultado` ou ...):

Código em R

```
> a<-sqrt(x=5)
```

O R não retorna nada, mas o resultado de `sqrt(5)` foi agora atribuído a uma estrutura de dados, chamada de vetor, que recebeu o nome de `a`. Você pode testar se a atribuição foi bem sucedida verificando o conteúdo de `a`. Uma função utilizada para fazer isso é a `print`, e seu argumento minimamente necessário é a estrutura de dados cujo conteúdo você deseja ver, mas na maioria das vezes, basta simplesmente digitar o nome da estrutura de dados em questão:

Código em R

```
> print(a)
[1] 2.236068
> a
[1] 2.236068
```

Três comentários finais fazem-se necessários antes da nossa discussão mais detalhada sobre várias estruturas de dados. Primeiramente, o R ignora tudo aquilo que é escrito em uma linha após o símbolo `#` (jogo da velha), o qual você pode utilizar para escrever comentários em cada linha (para lembrar o que aquela linha está fazendo). Em segundo lugar, o operador de atribuição pode também ser usado para atribuir um novo valor para uma estrutura de dados já existente. Por exemplo:

Código em R

```
> a=sqrt(x=9) # armazena o resultado da função
> a # mostra o resultado armazenado
[1] 3
> a=a+2 # substitui a estrutura anterior pela nova
> a # mostra o novo resultado da estrutura chamada a
[1] 5
```

Se você quiser deletar ou limpar uma estrutura de dados, você pode utilizar a função `rm` (do inglês *remove*). Você pode eliminar uma estrutura de dados específica ou todas as que estão armazenadas. Essas opções são definidas na parte de argumentos da função.

Código em R

```
> rm(a) # apaga a estrutura de dados chamada de a
> rm(list=ls(all=TRUE)) # apaga todos os dados armazenados
```

Em terceiro lugar, mais tarde será muito importante saber que as funções têm ordens padrão de seus argumentos e que muitas funções têm configurações padrão para seus argumentos. A informação da sentença anterior significa que, se você fornecer argumentos em sua ordem padrão, você não precisa nomeá-los. Isto é, em vez de `sqrt(x = 9)`, você poderia simplesmente escrever `sqrt(9)` porque o (único) argumento `x` está na sua “posição padrão”. A segunda informação significa que se você usar uma função sem especificar todos os argumentos necessários, então R usará as configurações padrão, se elas forem fornecidas por essa função. Vamos explorar isso com base em uma função muito útil chamada `sample`. Essa função gera amostras de elementos aleatórias ou pseudo-aleatórias e pode requerer até quatro argumentos:

- `x`: uma estrutura de dados - normalmente um vetor - que define os elementos do qual você quer uma amostra;
- `size`: um número positivo que define o tamanho da amostra;
- `replace`: que pode ter o valor verdadeiro (`replace=TRUE`) ou falso (`replace=FALSE`). Se definido como verdadeiro, quando um elemento for escolhido da estrutura de dados, ele volta e pode ser escolhido novamente. Se for definido como falso, cada elemento só pode ser escolhido uma única vez. A definição padrão é FALSO;
- `prob`: um vetor com as probabilidades de cada elemento ser escolhido. O parâmetro desse argumento é vazio (NULL) na sua especificação padrão, ou seja, todos os elementos têm a mesma probabilidade de serem escolhidos.

Vamos executar alguns exemplos que farão mais uso de ordens padrão e configurações de argumentos padrão. Primeiro, você gera um vetor contendo os números de 1 a 10 utilizando a função `c` (do inglês *concatenate*); os dois pontos entre os números 1 e 10 geram uma sequência de números inteiros contidos entre os dois valores:

Código em R

```
> some.data=c(1:10)
```

Agora, imagine que você queira uma amostra de cinco elementos contidos nesse vetor, de forma que cada elemento tenha a mesma probabilidade de ser escolhido, e possa ser escolhido mais de uma vez. O comando que você deve utilizar é o seguinte:²

Código em R

```
> sample(x=some.data, size=5, replace=TRUE, prob=NULL)
[1] 5 9 9 9 2
```

Porém, se você listar os argumentos da função em sua ordem padrão (como fazemos aqui), você pode escolher não colocar os nomes dos argumentos:

² Os seus resultados serão obviamente diferentes, pois se trata de uma amostra aleatória.

Código em R

```
> sample(some.data,5,TRUE,NULL)¶  
[1] 3 8 4 1 7
```

Além disso, como `prob=NULL` é padrão, você pode deixar esse argumento também de fora:

Código em R

```
> sample(some.data,5,TRUE)¶  
[1] 2 1 9 9 10
```

Para o mesmo comando, mas fazendo com que cada elemento só possa ser escolhido UMA ÚNICA VEZ, temos:

Código em R

```
> sample(some.data,5,FALSE)¶  
[1] 1 10 6 3 8
```

Uma vez que `replace=FALSE` é a definição padrão, podemos deixá-lo também de fora da função:

Código em R

```
> sample(some.data,5)¶  
[1] 10 5 9 3 6
```

Algumas vezes, você pode deixar também de fora o argumento `size`. Se você fizer isso, o R irá entender que TODOS os elementos devem ser escolhidos, o que te dará, no final das contas, um vetor com TODOS os elementos em uma ordem aleatória:

Código em R

```
some.data¶  
[1] 1 2 3 4 5 6 7 8 9 10  
> sample(some.data)¶  
[1] 2 4 3 10 9 8 1 6 5 7
```

Finalmente, imagine que você apenas deseja os números de 1 a 10 em uma ordem aleatória. Para isso, você nem precisa de um vetor contendo os números de 1 a 10. Basta que você utilize a função `sample` contendo apenas um argumento (o número 10):

Código em R

```
> sample(10)¶  
[1] 5 10 2 6 1 3 4 9 7 8
```

Em casos extremos, algumas funções podem até ser digitadas sem nenhum argumento. Considere a função `q` (do inglês *quit*). Essa função serve para fechar o R. Os argumentos dessa função são:

- `save`: uma linha de caracteres dizendo se você quer ou não salvar o que você fez na área de trabalho. O padrão é: o programa irá te perguntar antes de fechar;

- `status`: um número que indica o *status* de erro do sistema operacional. O valor padrão é 0 e significa ‘conclusão bem-sucedida’;
- `runLast`: um valor lógico (TRUE or FALSE), indicando se uma função chamada `Last` deve ser executada antes do R ser fechado; o padrão é TRUE.

Assim, se você quiser fechar o R com essas configurações, basta digitar:

Código em R

```
> q()
```

O R irá então perguntá-lo se você deseja salvar a área de trabalho ou não. Quando você tiver respondido essa pergunta, o R irá executar a função `Last` (se você definiu alguma), fechar o programa e enviar “0” para o sistema operacional.

Como você já deve ter percebido, os valores padrão são bem úteis para minimizar a quantidade de material a ser digitado para executar uma função. No entanto, especialmente no início, como uma boa maneira de aprender o R, é interessante balancear um pouco no uso dos *defaults* e no uso do código completo. Embora isso acabe sendo uma questão de preferência pessoal, recomendo usar códigos mais explícitos no início para que você se torne maximamente consciente das opções que seu código R usa. Você poderá encurtar seus códigos na medida em que você se tornar mais proficiente.

Recomendação para estudo/leitura

As funções `? ou help` te mostram arquivos de ajuda para algumas funções. Tente usar agora, por exemplo, `?sample` ou `help(sample)`. A função `formals` te mostra os argumentos que uma função exige, seus argumentos padrão e a ordem em que eles devem ser digitados. Tente agora `formals(sample)` ou `args(sample)`.

2.3 Vetores

2.3.1 Como criar vetores no R

A estrutura de dados mais básica do R é um vetor. Vetores são sequências de elementos (números ou palavras/sequências de letras) que possuem apenas uma dimensão. É importante entender bem a estrutura de um vetor, uma vez que todas as outras estruturas de dados que iremos trabalhar no R podem ser entendidas em termos de vetores. Sem perceber, já utilizamos um vetor anteriormente quando calculamos a raiz quadrada de 5:

Código em R

```
> sqrt(5)
[1] 2.236068
```

O “[1]” antes do resultado indica que o primeiro (e aqui, único) elemento retornado como resultado da função `sqrt(5)` é o elemento número 1, que é 2,236068. Para testar se esse resultado é mesmo um vetor, podemos utilizar o próprio R. Primeiramente, devemos dar um nome ao resultado da função `sqrt(5)`. Vamos chamar esse resultado de `a`.

Código em R

```
> a=sqrt(5)¶
```

A função `is.vector` testa se o seu argumento é ou não um vetor. O R responde com `TRUE` para “sim” e `FALSE` para “não”:

Código em R

```
> is.vector(a)¶  
[1] TRUE
```

E a função `length` (do inglês “comprimento”) determina o número de elementos do vetor informado no argumento da função:

Código em R

```
> length(a)¶  
[1] 1
```

É claro que você também pode criar argumentos que contêm sequências de letras – a única diferença é que a sequência de letras deve ser colocada entre aspas:

Código em R

```
> a.name="John"; a.name¶  
[1] "John"
```

Neste livro, iremos lidar apenas com vetores lógicos, numéricos e de caracteres. Vetores usualmente se tornam interessantes apenas quando contêm mais de um elemento. Você já conhece a função que cria tais vetores – a função `c`. Os argumentos dessa função são os elementos que devem ser concatenados. Os elementos devem ser separados por vírgula. Por exemplo:

Código em R

```
> numbers=c(1, 2, 3); numbers¶  
[1] 1 2 3
```

ou

Código em R

```
> same.names=c("al", "bill", "chris"); same.names¶  
[1] "all" "bill" "chris"
```

Note que, uma vez que números e sequências de letras individuais também são vetores (justamente vetores de comprimento 1), a função `c` pode também combinar vetores com dois ou mais elementos:

Código em R

```
> numbers1=c(1,2,3);numbers2=c(4,5,6)# para criar dois vetores¶
> numbers1.e.numbers2=c(numbers1,numbers2)# combinação dos dois vetores¶
> numbers1.e.numbers2¶
[1] 1 2 3 4 5 6
```

Uma outra função com resultado semelhante é `append` (do inglês anexar). Essa função deve ter pelo menos dois argumentos e no máximo três (e como sempre, os argumentos devem ser separados por vírgula):

- `x`: um vetor ao qual o elemento anexado;
- `values`: o vetor a ser anexado;
- `after`: o local no primeiro argumento onde os elementos do segundo argumento devem ser anexados – no começo ou no final do vetor definido em `x`; o valor padrão é no final.

Assim, utilizando `append`, o exemplo anterior deve ficar desta forma:

Código em R

```
> numbers1.e.numbers2=append(numbers1,numbers2)# combinação dos dois vetores¶
> numbers1.e.numbers2¶
[1] 1 2 3 4 5 6
```

Um outro exemplo típico de como a função `append` pode ser utilizada é dado a seguir, quando queremos modificar um vetor já existente:

Código em R

```
> evenmore=c(7, 8)¶
> numbers=append(numbers1.e.numbers2, evenmore)¶
> numbers¶
> [1] 1 2 3 4 5 6 7 8
```

É importante ressaltar que, diferentemente de vetores em Perl, no R, vetores só podem conter elementos de um único tipo. Por exemplo, um vetor pode conter números *ou* sequências de caracteres, mas não os dois: se você tentar colocar os dois tipos de elementos juntos em um único vetor, o R irá mudar o tipo de dados de um dos elementos para homogeneizar os tipos de elementos do vetor (ou todos os elementos serão letras, ou serão números), e uma vez que você pode interpretar números como caracteres, mas não vice-versa, o R transforma os números em sequências de caracteres e depois as combina em um vetor de sequências de caracteres:

Código em R

```
> mixture=c("al", 2, "chris"); mixture¶
[1] "al"      "2"       "chris"
```

Código em R

```
> numbers.num=c(1, 2, 3); numbers.char=c("four", "five", "six")
> nums.and.chars=c(numbers.num, numbers.char)
> nums.and.chars
[1] "1" "2" "3" "four" "five" "six"
```

As aspas nos números 1, 2 e 3 indicam que esses números estão na verdade sendo tratados como caracteres, ou seja, eles não podem ser utilizados para cálculos matemáticos. Para verificar o tipo de vetor (ou melhor, identificar o tipo de dado contido no vetor), usamos a função `str` (do inglês *structure*). O argumento dessa função é o nome da estrutura de dados:

Código em R

```
> str(numbers.num)
num [1:3] 1 2 3
> str(nums.and.chars)
chr [1:6] "1" "2" "3" "four" "five" "six"
```

O primeiro vetor consiste em três elementos numéricos (1, 2 e 3). O segundo vetor consiste em seis sequências de caracteres (*character*) que são exibidos.

Como veremos mais adiante, precisaremos por algumas vezes criar vetores longos em que elementos ou sequências de elementos precisarão ser repetidas. Ao invés de digitar esses elementos, poderemos utilizar duas funções bastante úteis para esse fim. As funções são `rep` e `seq`. A função `rep` (do inglês *repetition*) pede dois argumentos: o elemento ou elementos que deseja repetir e o número de vezes que esse elemento precisa ser repetido. Para criar, por exemplo, um vetor `x` que repita 4 vezes a sequência numérica 1, 2 e 3, você deve digitar o comando:

Código em R

```
> numbers=c(1, 2, 3)
> x=rep(numbers, 4)
```

ou

Código em R

```
> x=rep(c(1, 2, 3), 4); x
[1] 1 2 3 1 2 3 1 2 3 1 2 3
```

Para criar um vetor em que os números de 1 a 3 são repetidos individualmente 4 vezes – não em sequência – utilizamos o argumento `each`:

Código em R

```
> x=rep(c(1, 2, 3), each=4); x
[1] 1 1 1 1 2 2 2 2 3 3 3 3
```

O mesmo pode ser feito com vetores contendo sequências de caracteres. Para sequências de números inteiros, você pode também utilizar : (dois pontos) para definir o intervalo da sequência:

Código em R

```
> x=rep(c(1:3), 4)¶
```

A função `seq` (do inglês *sequence*) é utilizada de uma maneira um pouco diferente, com três argumentos:

- `from`: define o ponto inicial da sequência;
- `to`: define o ponto final da sequência;
- `by`: o incremento da sequência.

Assim, ao invés de utilizar o comando `numbers=c(1:3)¶`, você pode utilizar:

Código em R

```
> numbers=seq(1, 3, 1)¶
```

Uma vez que o incremento padrão é 1, seria suficiente fazer assim:

Código em R

```
> numbers=seq(1, 3)¶
```

Na verdade, você pode até escrever apenas isto:

Código em R

```
> numbers=seq(3)¶
```

Se a sequência numérica criada não utiliza o incremento 1, você pode definir qualquer outro valor para esse argumento. Por exemplo, para criar um vetor `x` em que os elementos pares contidos entre 1 e 10 sejam repetidos seis vezes em sequência, utilizamos as seguintes linhas:

Código em R

```
> numbers=seq(2, 10, 2)¶  
> x=rep(numbers, 6)¶
```

ou

Código em R

```
> x=rep(seq(2, 10, 2), 6)¶
```

Finalmente, em vez de fornecer o incremento, você também pode deixar R descobri-lo para você, como quando você sabe a extensão que sua sequência deve ter e apenas deseja incrementos iguais em todos os lugares. Você pode então usar o argumento `length.out`. O seguinte comando gera uma sequência de 7 elementos de 1 a 10 com incrementos iguais e a atribui a `numbers`:

Código em R

```
> numbers=seq(1, 10, length.out=7); numbers
[1] 1.0 2.5 4.0 5.5 7.0 8.5 10.0
```

Com as funções `c`, `append`, `rep`, e `seq`, podemos criar vetores longos e complexos com muita facilidade. Um outro ponto interessante a destacar é que você pode nomear tanto vetores completos, como também seus elementos individuais:

Código em R

```
numbers=c(1, 2, 3); names(numbers)=c("one", "two", "three")
> numbers
one two three
1    2     3
```

Antes de falarmos sobre como carregar e salvar vetores no R, vamos falar um pouco sobre uma forma interativa de criar vetores no R. Se você utilizar a função `scan()` (para vetores numéricos) ou `scan(what=character())` (para vetores com sequências de caracteres), você consegue digitar cada número ou sequência de caracteres separadamente pressionando a tecla ENTER para passar para o próximo elemento. Quando terminar de digitar os elementos, pressione a tecla ENTER duas vezes:

Código em R

```
> x=scan()
1: 1
2: 2
3: 3
4: 
Read 3 items
> x
[1] 1 2 3
```

Recomendação para estudo/leitura

As funções `as.numeric` e `as.character` transformam vetores em numéricos e em sequência de caracteres, respectivamente.

2.3.2 Como carregar e salvar um vetor no R

Uma vez que grande parte das estruturas de dados que utilizamos no R para análises estatísticas não são digitadas manualmente no R, vamos aprender agora como importar dados de outros arquivos para o R. Primeiro um comentário geral: O R consegue “ler” arquivos de vários formatos distintos, mas vamos discutir aqui apenas os arquivos salvos em formato de texto, ou seja arquivos com a extensão `<.txt>` ou `<.csv>`. Assim, se o seu arquivo de dados não foi criado por um editor de textos, mas sim por uma planilha do tipo LibreOffice Calc, você deve primeiro salvar esses dados no formato de texto (no menu Arquivo: Salvar como ... e Salvar como tipo: *Texto CSV (.csv)*).

Uma função poderosa para entrar com os dados no R é a função `scan`, a qual já utilizamos para digitar os dados manualmente. Os argumentos dessa função são vários (sugerimos que liste os argumentos

pelos seus nomes). Os argumentos mais importantes da função `scan` para os nossos objetivos com as suas respectivas configurações são:

- `file=`: define o caminho do arquivo que deseja carregar no R; por exemplo, `"_inputfiles/02-3-2_vector1.txt"`, mas na maioria das vezes é provavelmente mais fácil usar apenas a função `file.choose()`, a qual irá solicitar que você escolha o arquivo de interesse diretamente; note que o argumento `file` também pode ser `"clipboard"`;
- `what=`: o tipo de elemento que será lido pela função `scan`. As definições mais importantes são `what=double(0)` (para elementos numéricos e também a definição padrão) e `what=character(0)` (para sequências de caracteres)
- `sep=`: define o caracter que separa os elementos do arquivo. A definição padrão é, `sep=" "`, que significa que cada espaço em branco irá separar os elementos. Para utilizar tabs, a definição é `sep="\t"`, e para cada elemento em uma linha separada, a definição é `sep="\n"`;
- `dec=`: o caracter de ponto decimal; `dec="."` é o padrão definido; se você desejar usar uma vírgula no lugar do ponto, basta executar `dec=","`.

Para carregar o arquivo `<_inputfiles/02-3-2_vector1.txt>`, que contém os elementos mostrados na Figura 14, e chamá-lo de `x`, devemos digitar o comando:

```
1¶
2¶
3¶
4¶
5¶
```

Figura 11 – Exemplo de um arquivo com dados numéricos

Código em R

```
> x=scan(file=file.choose(), sep="\n")¶
Read 5 items
```

Para ver o conteúdo de `x`, basta:

Código em R

```
> x¶
[1] 1 2 3 4 5
```

Para carregar um arquivo de sequência de caracteres, como o mostrado na Figura 12, o procedimento é o mesmo. Você apenas tem que dizer ao R que está carregando um arquivo de sequências de caracteres que são separadas por espaços:

```
alpha·bravo·charly·delta·echo¶
```

Figura 12 – Exemplo de um arquivo com dados não-numéricos

Código em R

```
> x=scan(file.choose(), what=character(0), sep=" ")
```

Você obtém:

Código em R

```
> x  
[1] "alpha" "bravo" "charly" "delta" "echo"
```

Agora vamos ver como fazer para salvar um vetor em um arquivo de texto. A função que iremos utilizar – que basicamente é o oposto da função `scan` – é a função `cat`, a qual requer argumentos muito semelhantes:

- o vetor, ou vetores, que pretende salvar em formato de arquivo de texto;
- `file=`: o nome do arquivo que será salvo (ou, no sistema operacional Windows, `choose.files()`);
- `sep=`: define o caracter que separa os elementos do arquivo: `sep=` para espaços (padrão), `sep="\t"`, para tabs e `sep="\n"` para salvar cada elemento em sua própria linha;
- `append=TRUE` ou `append=FALSE` (definição padrão): esse argumento serve para você escolher se quer salvar um ou mais vetores em um arquivo já existente. Se escolher `append=TRUE`, o vetor será anexado à um arquivo já existente (definido por você).

Assim, para anexar dois nomes ao vetor `x` e então salvá-lo com um outro nome, você pode utilizar o comando a seguir:

Código em R

```
> x=append(x, c("foxtrot", "golf"))  
> cat(x, file=choose.files())
```

Recomendação para estudo/leitura

Leia sobre a função `write`, `save` e `dput` para salvar vetores (ou outras estruturas)

2.3.3 Como editar vetores no R

Agora que você já sabe como criar, carregar e salvar vetores, vamos aprender como editá-los. As funções que vamos aprender aqui te permitem acessar uma parte de um vetor, utilizar essas partes em algumas outras funções e também permite alterar essas partes. Vamos começar vendo algumas funções que utilizamos para editar vetores numéricos. Uma dessas funções é `round` (do inglês arredondar). O primeiro argumento dessa função é o vetor que pretendemos arredondar, e o segundo é o número de casas decimais que queremos. (Note que o R arredonda os números de acordo com o padrão IEEE: 3,55 é arredondado para 3,5 e não 3,6)

Código em R

```
> a=seq(3.4, 3.6, 0.05); a
[1] 3.40 3.45 3.50 3.55 3.60
> round(a, 1)
[1] 3.4 3.4 3.5 3.5 3.6
```

A função `floor` arredonda para baixo, já a função `ceiling` arredonda para cima. A função `trunc` simplesmente reduz os elementos em direção a zero.

Código em R

```
> floor(c(-1.8, 1.8))
[1] -2 1
> ceiling(c(-1.8, 1.8))
[1] -1 2
> trunc(c(-1.8, 1.8))
[1] -1 1
```

A forma mais importante de acessar partes de um vetor (ou de qualquer outra estrutura de dados) envolve escolher partes do vetor (sub-conjuntos) através do uso de colchetes. De uma maneira bem simples, é assim que você seleciona elementos de um vetor:

Código em R

```
> x=c("a", "b", "c", "d", "e")
> x[3]
[1] "c"
```

Como o R é um programa bem flexível, o conjunto de códigos abaixo não deve ser visto com surpresa por você:

Código em R

```
> y=3; x[y]
[1] "c"
> z=c(1, 3); x[z]
[1] "ac"
> z=c(1:3); x[z]
[1] "abc"
```

Se utilizar números negativos, você seleciona todo o vetor MENOS o elemento que o número representa:

Código em R

```
> x[-2]
[1] "a" "c" "d" "e"
```

No entanto, existem formas mais poderosas de acessar partes de um vetor. Por exemplo, você pode pedir ao R que selecione apenas elementos que satisfaçam uma certa condição. Uma forma de estabelecer condições no R é através da utilização de *marcadores lógicos*:

Código em R

```
> x=="d"
[1] FALSE FALSE FALSE TRUE FALSE
```

O comando acima pede ao R que cheque, para cada elemento do vetor `x`, se ele é “d” ou não e retorna a resposta para você. Note que o marcador lógico para “igual” utiliza dois sinais de igual. Isso é para diferenciá-lo do sinal de igual que utilizamos para atribuir nomes às estruturas de dados (`file=`). Outros marcadores lógicos que nos serão úteis são:

<code>&</code>	e	<code> </code>	ou
<code>></code>	maior que	<code><</code>	menor que
<code>>=</code>	maior ou igual que	<code><=</code>	menor ou igual que
<code>!</code>	não	<code>!=</code>	diferente

Pausa para pensar

Veja alguns exemplos:

Código em R

```
> x=c(10:1) # cria um vetor com numeros de 10 a 1
> x
[1] 10 9 8 7 6 5 4 3 2 1
> x==4 # quais elementos de x sao 4?
[1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
> x<=7 # quais elementos de x sao <= 7?
[1] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
> x!=8 # quais elementos de x são diferentes de 8?
[1] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
> (x>8 | x<3) # quais elementos de x são > 8 ou < 3?
[1] TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
```

Uma vez que no R `TRUE` e `FALSE` correspondem à 1 e 0 respectivamente, é fácil determinar com que frequência uma expressão lógica em particular é verdadeira em um vetor:

Código em R

```
> sum(x==4)
[1] 1
> sum(x>8 | x<3)
[1] 4
```

A função `table` conta quantas vezes certos elementos (ou combinações de elementos) ocorre em um vetor. Por exemplo, com a função `table` conseguimos determinar imediatamente quantos elementos do vetor `x` são maiores que 8 ou menores que 3. (Note que a função `table` ignora valores faltosos, ou seja, se quiser contar também o número de valores não preenchidos, você deve digitar `table(..., exclude=NULL)`)

Código em R

```
> table(x>8 | x<3)
FALSE  TRUE
6      4
```

Parece óbvio, no entanto, que os comandos acima não são maneiras muito elegantes de identificar as posições dos elementos em um vetor. Mesmo que muitos elementos de *x* satisfaçam uma condição, você sempre tem como resultado 10 respostas e deve então localizar os TRUES um por um. Imagine se você tivesse um vetor com 10.000 elementos? No entanto, o R tem uma função que localiza para você os elementos que satisfazem uma certa função. A função é *which* e seus argumentos são os mesmos utilizados anteriormente:

Código em R

```
> which(x==4) # quais elementos de x são 4?
[1] 7
```

Como você pode notar, essa função funciona como se você estivesse conversando com o programa. Você pergunta ao programa “Qual elemento de *x* é 4?”, ele te responde dizendo que o sétimo elemento em *x* é um 4. Veja os outros exemplos de como a função *which* é utilizada:

Código em R

```
> which(x<=7) # quais elementos de x são <= 7?
[1] 4 5 6 7 8 9 10
> which(x!=8) # quais elementos de x não são 8?
[1] 1 2 4 5 6 7 8 9 10
> which(x>8 | x<3) # quais elementos de x são > 8 ou < 3?
[1] 1 2 9 10
```

Você pode facilmente criar uma outra estrutura de dados com os resultados acima, ou seja, você pode criar um vetor:

Código em R

```
> y=which(x>8 | x<3)
> y
[1] 1 2 9 10
```

Importante: não confunda a *posição de um elemento* no vetor com o próprio *elemento* do vetor. O comando *which(x==4)* não te dá como resposta o elemento 4, mas sim a posição do elemento 4 no vetor *x*, que é 7; o mesmo ocorre para os outros exemplos. Mas como obter os próprios elementos e não apenas suas posições? Lembre-se que o R utiliza vetores. A estrutura de dados que acabou de criar, *y*, é um vetor:

Código em R

```
> is.vector(y)
[1] TRUE
```

Vimos anteriormente que podemos utilizar colchetes para acessar elementos específicos em um vetor. Assim, imagine que você tenha o vetor *x* e quer saber, não só as posições dos elementos maiores

que 8 ou menores que 3, mas também quer saber quais são esses elementos. Para isso, basta que coloque a função `which` dentro dos colchetes:

Código em R

```
> y=which(x>8 | x<3)¶  
> x[y]¶  
[1] 10 9 2 1  
> x[which(x>8 | x<3)]¶  
[1] 10 9 2 1
```

Ou você usa a seguinte opção, que se baseia no fato de que, quando você seleciona um subconjunto com um vetor lógico de TRUEs e FALSEs, o R retorna os elementos subcategorizados por TRUEs:

Código em R

```
> x[x>8 | x<3]¶  
[1] 10 9 2 1
```

Você utiliza uma abordagem semelhante para acessar a frequência com a qual uma expressão é verdadeira:

Código em R

```
> length(which(x>8 | x<3))¶  
[1] 4
```

Às vezes você precisa verificar se vários elementos satisfazem alguma condição de uma só vez (por exemplo, os números 1, 6 e 11), o que não pode ser feito com a função `which`, mas sim com o operador muito útil `%in%`:

Código em R

```
> c(1, 6, 11) %in% x¶  
[1] TRUE TRUE FALSE
```

O resultado do operador `%in%` é um vetor que contém a resposta que diz se os elementos antes do operador `%in%` ocorrem no vetor depois do operador `%in%`. Se você quiser saber a posição exata da primeira ocorrência de cada um dos elementos, você pode utilizar o `match`:

Código em R

```
> match(c(1, 6, 11), x)¶  
[1] 10 5 NA
```

O resultado significa o seguinte: o primeiro elemento do primeiro vetor – o 1 – ocorre pela primeira (e única vez) na décima posição do vetor `x`; o segundo elemento do primeiro vetor – o 6 – ocorre pela primeira (e única vez) na quinta posição do vetor `x`; o último elemento do primeiro vetor – o 11 – não ocorre no vetor `x`.

Espero que esteja ficando claro para você que muitas das operações do R ocorrem em termos de vetores. Essa é uma das grandes potencialidades do R. Quase tudo que fizemos até agora foi baseado na

ideia de vetores (geralmente com 1 elemento). Assim, você pode sempre utilizar as funções de maneira flexível, até mesmo colocando uma função dentro de outra. Por exemplo, agora que você já sabe como acessar partes de um vetor, você já pode também modificar essas partes. Imagine que você queira mudar todos os elementos maiores que 8 no vetor `x` para 12:

Código em R

```
> x # mostra x novamente
[1] 10 9 8 7 6 5 4 3 2 1
> y=which(x>8)
> x[y]=12
> x
[1] 12 12 8 7 6 5 4 3 2 1
```

Como você deve ter notado, uma vez que você quer substituir mais de um elemento de `x` mas apenas forneceu um elemento substitutivo (12), o R recicla o valor que você forneceu até que todos os elementos de R sejam substituído (vamos falar sobre essa característica mais adiante). Uma maneira mais curta de fazer a mesma coisa é:

Código em R

```
> x=10:1
> x[which(x>8)]=12
> x
[1] 12 12 8 7 6 5 4 3 2 1
```

Uma maneira ainda mais curta:

Código em R

```
> x=10:1
> x[x>8]=12
> x
[1] 12 12 8 7 6 5 4 3 2 1
```

O R também oferece várias funções relacionadas à Teoria dos Conjuntos – `setdiff`, `intersect`, e `union`. Essas funções requerem dois vetores como argumentos. A função `setdiff` retorna os elementos do primeiro vetor que não está contido no segundo vetor. A função `intersect` retorna os elementos do primeiro vetor que estão presentes também no segundo vetor. A função `union` retorna todos os elementos que ocorrem em pelo menos um dos dois vetores.

Código em R

```
> x=c(10:1); y=c(2, 5, 9, 12)
> setdiff(x, y)
[1] 10 8 7 6 4 3 1
> setdiff(y, x)
[1] 12
> intersect(x, y)
[1] 9 5 2
```



```
> intersect(y, x)
[1] 2 5 9
> union(x, y)
[1] 10 9 8 7 6 5 4 3 2 1 12
> union(y, x)
[1] 2 5 9 12 10 8 7 6 4 3 1
```

Uma outra função bem útil é a função `unique`, que pode ser facilmente explicada para linguistas: `unique` lê todos os elementos de um vetor (como se fosse uma frequência de ocorrência) e retorna todos os elementos que ocorrem pelo menos uma vez (como se fosse uma frequência de tipo).

Código em R

```
> x=c(1, 2, 3, 2, 3, 4, 3, 4, 5)
> unique(x)
[1] 1 2 3 4 5
```

Outra característica interessante do R é que você pode facilmente realizar algumas funções matemáticas a um conjunto de elementos de um vetor numérico. Se você aplicar uma função matemática a um vetor, essa função é aplicada a todos os elementos do vetor:

Código em R

```
> x=c(10:1)
> x
[1] 10 9 8 7 6 5 4 3 2 1
> y=x+2
> y
[1] 12 11 10 9 8 7 6 5 4 3
```

Se você somar dois vetores (ou multiplicar um pelo outro, ou dividir, ou ...) três resultados podem ser obtidos. Primeiro, se os vetores têm o mesmo tamanho, a operação é executada para cada um dos pares dos elementos contidos nos vetores correspondentes:

Código em R

```
> x=c(2, 3, 4); y=c(5, 6, 7)
> x*y
[1] 10 18 28
```

Segundo, os vetores não têm tamanhos iguais, e o tamanho do vetor maior é múltiplo do tamanho do vetor menor sem que haja resto. Então, os elementos do vetor menor serão reciclados quantas vezes forem necessárias até que todos os elementos do vetor maior tenham sido utilizados. Como você pode ver acima, o tamanho do vetor menor é geralmente 1.

Código em R

```
> x=c(2, 3, 4, 5, 6, 7); y=c(8, 9)
> x*y
[1] 16 27 32 45 48 63
```

Terceiro, os vetores não têm o mesmo tamanho e o vetor maior não é divisível pelo vetor menor. Neste caso, o R irá reciclar o vetor menor quantas vezes for necessário, mas irá te retornar uma mensagem de aviso, dizendo que o vetor maior não é divisível pelo vetor menor.

Código em R

```
> x=c(2, 3, 4, 5, 6); y=c(8, 9)¶
> x*y¶
[1] 16 27 32 45 48
Warning message:
In x * y : longer object length is not a multiple of shorter
object length
```

Finalmente, vamos ver duas funções utilizadas para mudar a ordem dos elementos de um vetor. A primeira é chamada `sort`, e o mais importante argumento dessa função é obviamente o vetor que se pretende ordenar; um outro argumento importante dessa função define o tipo de ordenação: do menor valor para o maior – `decreasing=FALSE` (definição padrão) – ou do maior valor para o menor – `decreasing=TRUE`.

Código em R

```
> x=c(1, 3, 5, 7, 9, 2, 4, 6, 8, 10)¶
> y=sort(x)¶
> z=sort(x, decreasing=TRUE)¶
> y; z¶
[1] 1 2 3 4 5 6 7 8 9 10
[1] 10 9 8 7 6 5 4 3 2 1
```

A segunda função com o mesmo propósito de ordenar é `order`. Ela tem como argumento um ou mais vetores, e também o argumento `decreasing=...` que define o tipo de ordenação (crescente ou decrescente) – mas tem como resultado algo que pode, a primeira vista, não ser muito óbvio. Você consegue ver o que a função `order` faz?

Código em R

```
> z=c("a", "c", "e", "d", "b")¶
> order(z, decreasing=FALSE)¶
[1] 1 5 2 4 3
```

Pausa para pensar

O output de `order` quando aplicado a um vetor `z` é um vetor que diz em que ordem você deve colocar os elementos de `z` para classificá-los como especificado. Vamos explicar melhor esta caracterização não tão simples de entender: se você deseja classificar os valores de `z` em ordem crescente, você primeiro tem que considerar o primeiro valor de `z`. Assim, o primeiro Valor de `order(z, decreasing=FALSE)¶` é 1. O próximo valor que você tem que considerar é o quinto valor de `z`. O próximo valor é o segundo valor de `z`, etc. (Se você fornecer mais de um vetor para a função `order`, vetores adicionais são usados para quebrar os laços.) Como veremos abaixo, esta função será útil quando aplicada a estrutura de dados.

Recomendação para estudo/leitura

- as funções `any` e `all` para verificar se todos ou nenhum elemento de um vetor satisfazem uma função específica
- a função `abs` para obter os valores absolutos de um vetor numérico
- as funções `min` e `max` para obter os valores mínimos e máximos de vetores numéricos

2.4 Fatores

À primeira vista, a estrutura de dados conhecida como fatores é muito semelhante a um vetor de caracteres (ex.: palavras). Fatores serão úteis quando importarmos estruturas de dados e quisermos que o R reconheça algumas de suas colunas como variáveis nominais ou categóricas.

2.4.1 A criação de fatores no R

Como mencionado antes, fatores têm a função principal de codificar variáveis nominais e categóricas, ou seja, situações em que as variáveis têm dois ou mais (mas não demais) níveis diferentes. A maneira mais simples de criar um fator é criar um vetor e depois alterá-lo para um fator utilizando a função `factor`, o qual pede dois argumentos. O primeiro argumento é o vetor que você deseja transformar em fator, e o segundo é `levels=...`, o qual será discutido em detalhes na Seção 2.4.3 abaixo.

Código em R

```
> rm(list=ls(all=TRUE))
> x=c(rep("male", 5), rep("female", 5))
> y=factor(x); y
[1] male male male male male female female female female female
Levels: female male
> is.factor(y)
[1] TRUE
```

Quando você gera um fator, você pode ver uma diferença entre os fatores e vetores porque a saída (*output*) inclui uma lista por padrão em ordem alfabética de todos os níveis desse fator.

Uma outra maneira muito útil de gerar fatores é baseada na função `cut`. Em sua implementação mais simples, essa função toma um vetor numérico como seu primeiro argumento (`x`) e um número de intervalos como seu segundo argumento (`breaks` (quebra)), e então divide `x` em intervalos de quebra:

Código em R

```
> cut(1:9, 3)
[1] (0.992,3.66] (0.992,3.66] (0.992,3.66] (3.66,6.34]
(3.66,6.34] (3.66,6.34] (6.34,9.01] (6.34,9.01]
(6.34,9.01]
Levels: (0.992,3.66] (3.66,6.34] (6.34,9.01]
```

Como você pode ver, o vetor com os números de 1 a 9 agora foi gravado como um fator com três níveis que fornecem os intervalos usados pelo R para cortar o vetor numérico.

- $0.992 < \text{interval/level } 1 \leq 3.66$;
- $3.66 < \text{interval/level } 1 \leq 6.34$;
- $6.34 < \text{interval/level } 1 \leq 9.01$.

Esta função tem outra forma de usar breaks e outros argumentos muito úteis, então você deve explorar essas possibilidades com mais detalhes: `?cut`.

2.4.2 Como carregar e salvar fatores no R

A forma como carregamos/importamos fatores para o R é a mesma que utilizamos para vetores. Basta que você transforme o vetor em fator logo após a importação. Para salvar um fator, no entanto, o processo é um pouco diferente. Suponhamos que você tenha o seguinte fator `a`.

Código em R

```
> a=factor(c("alpha", "charly", "bravo")); a
[1] alpha charly bravo
Levels: alpha charly bravo
```

Se você tentar salvar esse fator como um arquivo de texto – assim como fizemos para os vetores, seu resultado será parecido com o da Figura 13.

Código em R

```
> cat(a, sep="\n", file=file.choose())
```

```
1
2
3
```

Figura 13 – Um outro arquivo de exemplo

Isso acontece porque o R representa os fatores internamente em formato numérico. Logo, o R também apenas salva esses números no arquivo de texto. Mas, uma vez que você quer as palavras, você deve forçar o R a tratar o fator como se fosse um vetor, o que produzirá o resultado desejado:

Código em R

```
> cat(as.vector(a), sep="\n", file=file.choose())
```

2.4.3 Como editar fatores no R

A maneira de editar fatores no R também é muito semelhante à maneira como editamos vetores, porém um pouco mais trabalhosa quando você deseja introduzir novos níveis. Vamos criar um fator `x`:

Código em R

```
> x=factor(rep(c("long", "intmed", "short"), 1:3)); x
[1] long intmed intmed short short short
Levels: intmed long short
```

Observe como a organização em ordem alfabética dos níveis não é particularmente útil, uma vez que ela não coincide com uma ordem ascendente ou decrescente do significado dos níveis. A coisa mais fácil que você pode fazer é mudar o primeiro nível do primeiro nível segundo a ordem alfabética para um outro nível (o que será importante nos Capítulos 4 e 5). Por exemplo, você pode tornar `short` o primeiro nível. Para isso, você pode usar a função `relevel`, a qual requer que o fator seja alterado e o novo nível de referência:

Código em R

```
> x=relevel(x, "short"); x
[1] long intmed intmed short short short
Levels: short intmed long
```

Como você pode ver, o conteúdo do fator em si não mudou, apenas a ordem na qual os níveis estão listados, e agora temos uma boa ordenação dos níveis.

Se você quiser alterar a ordem dos níveis de forma mais substancialmente - por exemplo, invertendo a sua ordem - você pode usar a função `factor` novamente e atribuir os níveis da maneira desejada. Novamente, o conteúdo do fator é o mesmo, mas a ordem dos níveis é agora invertida.

Código em R

```
> x=factor(x, levels=levels(x)[3:1]); x
[1] long intmed intmed short short short
Levels: long intmed short
```

Agora, que tal mudar o conteúdo dos fatores? Você pode querer apenas alterar o nome de um nível nesse fator, e, para fazer isso, basta definir um novo nível, por exemplo, mudando `intmed` para `intermed`:

Código em R

```
> levels(x)[2]="intermed"; x
[1] long intermed intermed short shor short
Levels: long intermed short
```

Depois, você pode querer passar um determinado elemento para algum outro nível que já é reconhecido no fator. Nesse caso, você pode tratar fatores como você faria com vetores:

Código em R

```
> x[3]="short"; x
[1] long intermed short short short short
Levels: long intermed short
```

Uma dificuldade surge quando você deseja atribuir um nível completamente novo:

Código em R

```
> x[6]="supershort"
Warning message:
In '[<-.factor'('*tmp*', 6, value = "supershort") :
invalid factor level, NAs generated
> x
[1] long intermed short short short <NA>
Levels: long intermed short
```

Assim, se você deseja atribuir um novo nível, você deve proceder de forma diferente. Vamos recriar `x` e, em seguida, definir o novo (quarto) nível com `levels`:

Código em R

```
x=factor(rep(c("long", "intermed", "short"), c(1, 1, 4)),
levels=c("long", "intermed", "short"))
> x=factor(x, levels=c(levels(x), "supershort")); x
[1] long intermed short short short short
Levels: long intermed short supershort
```

Importante: o conteúdo do fator ainda não mudou, você só tem mais um nível do que antes. Isso também mostra que um fator pode ter níveis que não são reconhecidos em seu conteúdo. No entanto, agora que `x` tem todos os níveis que você precisa, você pode proceder como acima e atribuir o novo valor como você faria com um vetor:

Código em R

```
> x[6]="supershort"; x
[1] long intermed short short short supershort
Levels: long intermed short supershort
```

Agora, vamos supor que você mudou de idéia e mudou o sexto dado novamente para `short`:

Código em R

```
> x[6]="short"; x
[1] long intermed short short short short
Levels: long intermed short supershort
```

Obviamente, não é bom ter o nível `super short` listado nos níveis se ele não é realmente reconhecido, especialmente porque mais tarde vamos usar funções que retornariam resultados para esses níveis mesmo se eles não forem reconhecidos. Portanto, vamos nos livrar desse nível. Felizmente, isso é fácil: você pode apenas utilizar a função `factor` novamente, a qual eliminará os níveis não utilizados:

Código em R

```
> x=factor(x); x # also see ?droplevels
[1] long intermed short short short short
Levels: long intermed short
```

Às vezes, deseja-se combinar os níveis de um fator, para, por exemplo, testar se todos os níveis de um fator para os quais os dados do corpora foram anotados são realmente necessários. Vamos supor que você decidiu que você realmente só quer distinguir 'short' de 'not short'. Para fazer isso, você altera os níveis e o fator substituindo os dois primeiros níveis pelo novo nível.

Código em R

```
> levels(x)=c("not_short", "not_short", "short"); x
[1] not_short not_short short short short short
Levels: not_short short
```

Finalmente, como eu mencionei acima, o R armazena fatores como números, e há situações (especialmente no contexto de plotagem, ver Capítulo 5) nas quais é útil ter acesso a esses números, o que pode ser feito por meio da função `as.numeric`:

Código em R

```
> as.numeric(x)
[1] 1 1 2 2 2 2
```

Recomendação para estudo/leitura

- a função `is.factor` para testar se a estrutura de dados é um fator
- a função `gl` e `reorder` para criar fatores e reordenar níveis

2.5 Estruturas de dados

A estrutura de dados que é a mais relevante para quase todos os testes estatísticos que iremos fazer nesse livro é conhecido como *data frame*. Um *data frame*, que é basicamente uma tabela, é na verdade um tipo específico de uma outra estrutura de dados conhecida como lista. Uma vez que o *data frame* é a estrutura de dados mais utilizada, tanto no R quanto em outros pacotes estatísticos, vamos concentrar nos *data frames* e ignorar as listas.

2.5.1 Como criar um *data frame* no R

Um *data frame* pode ser facilmente criado a partir de vetores (e fatores). Suponhamos que você coletou três tipos de informação distintas com relação a cinco classes de palavras e queira gerar o *data frame* da figura 14:

- a variável Frequência Ocorrência (`FreqCorr`), ou seja, a frequência de palavras em uma determinada classe de palavra em um corpus X;
- a variável Frequência Tipo (`FreqTipo`), ou seja, o número de palavras distintas de uma determinada classe de palavra no corpus X;
- a variável Classe (`Classe`), que representa se a classe de palavra é de um grupo de classe-fechada ou classe-aberta.

POS	→	TOKENFREQ	→	TYPEFREQ	→	CLASS
adj	→	421	→	271	→	open
adv	→	337	→	103	→	open
n	→	1411	→	735	→	open
conj	→	458	→	18	→	closed
prep	→	455	→	37	→	closed

Figura 14 – Outro exemplo de um *data frame*

Primeiro passo: você deve criar quatro vetores, um para cada coluna da tabela:

Código em R

```
> rm(list=ls(all=TRUE))
> POS=c("adj", "adv", "n", "conj", "prep")
> TOKENFREQ=c(421, 337, 1411, 458, 455)
> TYPEFREQ=c(271, 103, 735, 18, 37)
> CLASS=c("open", "open", "open", "closed", "closed")
```

Segundo passo: A primeira linha da nossa tabela não contém dados, mas sim os nomes das colunas. Você deve definir se a primeira linha contém dados ou apenas os nomes das linhas. No primeiro caso, você precisa apenas criar um *data frame* com a função `data.frame`, a qual requer como argumentos os vetores relevantes. A ordem dos vetores define a ordem das colunas. Agora, você já pode ver as características de uma *data frame*:

Código em R

```
> x=data.frame(POS, TOKENFREQ, TYPEFREQ, CLASS)
```

```
> x
  POS TOKENFREQ TYPEFREQ CLASS
1 adj         421       271  open
2 adv         337       103  open
3 n          1411       735  open
4 conj         458        18 closed
5 prep         455        37 closed
> str(x)
'data.frame': 5 obs. of 4 variables:
 $ POS      : Factor w/ 5 levels "adj","adv",...: 1 2 4 3 5
 $ TOKENFREQ: num 421 337 1411 458 455
 $ TYPEFREQ  : num 271 103 735 18 37
 $ CLASS     : Factor w/ 2 levels "closed","open": 2 2 2 1 1
```

Dentro da estrutura de dados, o R alterou os vetores de sequência de caracteres para fatores e os representa internamente com números (por exemplo, `closed` é 1 e `open` é 2). É muito importante, neste caso, que o R apenas altere as variáveis em fatores quando elas contêm sequências de caracteres (e não apenas números). Se você tem uma estrutura de dados na qual as variáveis nominais ou categóricas são codificadas com números, então o R não saberá nem adivinhará que essas são fatores e tratará as variáveis como numéricas e, portanto, como variáveis de intervalo/razão nas análises estatísticas.

Portanto, você deve ou usar desde o início sequências de caracteres significativas como níveis de fator (como recomendado no Capítulo 1) ou deve caracterizar as variáveis relevantes como fatores no momento em que você cria a estrutura de dados: `factor(vetorname)`. Além disso, você não definiu os nomes das linhas, portanto o R automaticamente as enumera. Se você quiser usar as classes das palavras como nomes das linhas, você precisa dizer isso explicitamente:

Código em R

```
> x=data.frame(TOKENFREQ, TYPEFREQ, CLASS, row.names=POS) > x
  TOKENFREQ TYPEFREQ CLASS
adj         421      271  open
adv         337      103  open
n          1411      735  open
conj        458       18 closed
prep        455       37 closed
> str(x)
'data.frame': 5 obs. of 3 variables:
 $ TOKENFREQ: num 421 337 1411 458 455
 $ TYPEFREQ : num 271 103 735 18 37
 $ CLASS    : Factor w/ 2 levels "closed","open": 2 2 2 1 1
```

Como você pode ver, existem agora apenas três variáveis, pois POS agora funciona como nomes de linhas. Observe que isso só é possível quando a coluna com os nomes de linha não contém nenhum elemento ocorrendo duas vezes.

Uma segunda maneira de criar *data frames* que é muito menos flexível, mas extremamente importante para o Capítulo 5, envolve a função `expand.grid`. Em seu uso mais simples, a função possui como argumentos vários vetores ou fatores e retorna um *data frame* cujas linhas contêm todas as combinações possíveis de elementos de vetores e níveis de fatores. Parece complicado, mas é muito fácil de entendê-la a partir deste exemplo e vamos usá-la muitas vezes:

Código em R

```
> expand.grid(COLUMN1=c("a", "b"), COLUMN2=1:3)
  COLUMN1 COLUMN2
1        a      1
2        b      1
3        a      2
4        b      2
5        a      3
6        b      3
```

2.6 Como carregar e salvar *data frame* no R

Mesmo sabendo que podemos criar um *data frame* diretamente no R, essa não é a forma mais comum de entrar seus dados no R. Geralmente, vamos importar um arquivo que foi criado por um software de planilhas. Se você criar uma tabela no LibreOffice Calc e quiser importá-lo no R, você deve primeiramente salvar o arquivo no formato “texto separado por vírgulas”. Há duas maneiras de fazer isso. Você pode copiar o arquivo todo e colá-lo em um editor de texto (geany ou Notepad++, por exemplo), e então salvá-lo como um arquivo de texto separado por tabulações. Você pode também, diretamente no programa onde criou a planilha, salvar o arquivo com extensão CSV (*Arquivo: Salvar Como...* e *Salvar como tipo*

Texto CSV (.csv); então você escolhe a tabulação como o delimitador de campo e não escolhe nenhum delimitador de texto. Para carregar o arquivo no R, você deve utilizar a função `read.table` e alguns de seus argumentos:

- `file="..."`: o caminho para o arquivo com a tabela (com o Windows, você pode usar `choose.files()` para isso também; se o arquivo estiver ainda na área de transferência, você pode também digitar `file="clipboard"`);
- `header=T`: indica se a primeira linha contém o nome das colunas (o que sempre deve conter) ou `header=F` (definição padrão);
- `sep=`: define o delimitador das colunas; o padrão é `sep=` que significa espaço ou tabulação, mas geralmente você deve definir esse argumento utilizando `sep="\t"` de forma que você pode utilizar espaços nas células da sua tabela;
- `dec="."` ou `dec=", "`: o separador para casas decimais;
- `row.names=...` o número da coluna que contém o nome das linhas;
- `quote=...`: o padrão é que as aspas sejam marcadas com aspas simples ou duplas, mas você deve quase sempre definir `quote=`;
- `comment.char=...`: define o caracter que separa os comentários. O caracter padrão é "#", mas vamos sempre definir assim: `comment.char=`.

Assim, se quiser importar a tabela acima para o programa R que está em `<_inputfiles/02-5-2_dataframe1.txt>` – uma vez sem o nome das linhas e uma vez com o nome das linhas – é este o comando que você deve digitar no R:

Código em R

```
> a1=read.table(choose.files(), header=T, sep="\t", quote= , comment.char=
) # sem nome das linhas
```

ou

Código em R

```
> a2=read.table(choose.files(), header=T, sep="\t", quote= , comment.char=
, row.names=1) # com nome das linhas
```

Se digitar `a1` ou `str(a1)` (o mesmo com `a2`), você consegue ver se o *data frame* foi importado de maneira correta.

Embora a forma acima seja a maneira mais explícita e geral de carregar todos os tipos de estruturas de dados diferentes, quando você tiver configurado seus dados como recomendado acima, você pode usar uma versão mais curta com `read.delim`, a qual tem `header=TRUE` e `sep="\t"` como padrão e deve, portanto, funcionar na maioria das vezes:

Código em R

```
> a3=read.delim(file.choose())
```

Se você quiser salvar uma estrutura de dados criada no programa R, você pode utilizar o comando `write.table`. Os argumentos mais importantes para essa função são:

- `x`: um *data frame* que deseja salvar;
- `file`: o caminho para o diretório onde deseja salvar o arquivo. No sistema operacional Windows, você pode utilizar `choose.files()`;
- `append=F` (definição padrão) ou `append=T`: o primeiro comando substitui um arquivo já existente, o segundo agrega o conteúdo a um arquivo já existente;
- `quote=T` (definição padrão) ou `quote=F`: o primeiro comando mostra o nome dos fatores entre aspas; o segundo mostra o nome dos fatores sem aspas;
- `sep=`: você deve colocar entre as aspas o caractere que irá ser utilizado como separador das colunas; a definição padrão é " " que significa que um espaço irá separar as colunas. O que você deve utilizar é "\t", ou seja, tabulações;
- `eol3="\n"`: entre as aspas, você deve colocar o caracter que irá separar uma linha da outra; a definição padrão é "\n" que significa nova linha;
- `dec="."` (definição padrão): define o separador de casas decimais;
- `row.names=T` (definição padrão) ou `row.names=F`: define se o nome das linhas é incluído ou não;
- `col.names=T` (definição padrão) ou `col.names=F`: define se nome das colunas é incluído ou não.

Considerando as configurações padrão acima, e partindo do pressuposto de que seu sistema operacional usa um um arquivo de configuração em inglês, você pode salvar o *data frame* com o seguinte comando:

```
write.table(a1, choose.files(), quote=F, sep="\t"),col.names=NA¶
```

2.6.1 Como editar uma *data frame* no R

Nessa seção, vamos aprender como acessar partes de um *data frame*, como editar e como modificar um *data frame*.

Mais adiante, vamos ver vários exemplos de como acessar colunas específicas em um *data frame*. Existem várias maneiras de fazer isso. A primeira delas você pode inferir pela forma como o R mostra as colunas de um *data frame*. Quando você utiliza a função `str` para acessar a estrutura de uma estrutura de dados, note que o nome das colunas é precedido pelo símbolo "\$". Você pode utilizar a mesma sintaxe para acessar as colunas de um *data frame*, como pode ser visto no exemplo utilizando o arquivo `<_inputfiles/02-5-3_dataframe.csv>`.

³ eol de end of line

Código em R

```
> rm(list=ls(all=T)) # limpa memória
> a=read.table(choose.files())
> a
  ClaPal FreqOcor FreqTipo Classe
1   ADJ      421      271 aberta
2   ADV      337      103 aberta
3  SUBS     1411      735 aberta
4  CONJ      458       18 fechada
5  PREP      455       37 fechada
> a$FreqOcor
[1] 421 337 1411 458 455
> a$Classe
[1] aberta aberta aberta fechada fechada
Levels: aberta fechada
```

Você pode agora utilizar essas colunas como faria com qualquer outro vetor ou fator. Por exemplo, o comando a seguir calcula a razão entre a frequência de ocorrência e a frequência de tipo das classes gramaticais:

Código em R

```
> razao=a$FreqOcor/a$FreqTipo;razao
[1] 1.553506 3.271845 1.919728 25.444444 12.297297
```

Você pode também utilizar colchetes para acessar partes de um *data frame*. Discutimos anteriormente que se colocarmos a posição do elemento no vetor entre colchetes, acessamos esse elemento. Vetores e fatores são estruturas uni-dimensionais. Com estruturas bi-dimensionais (como um *data frame*), você pode também utilizar os colchetes, mas nesse caso é necessário informar a posição do elementos em relação às duas dimensões. É um procedimento simples. A única informação que você precisará memorizar é a ordem das posições – primeiro a linha, depois a coluna. As coordenadas devem ser separadas por vírgula. Vamos ver alguns exemplos:

Código em R

```
> a[2,3]
[1] 103
> a[2,]
  ClaPal FreqOcor FreqTipo Classe
2   ADV      337      103 aberta
> a[,3]
[1] 271 103 735 18 37
> a[2:3,4]
[1] aberta aberta
Levels: aberta fechada
> a[2:3,3:4]
  FreqTipo Classe
2      103 aberta
3      735 aberta
```

Note que os comandos acima não levam em consideração os nomes das colunas e das linhas. É importante também notar que todas as funções que aprendemos até agora que podem ser aplicadas à vetores podem também ser aplicadas às colunas extraídas de um *data frame*:

Código em R

```
> which(a[,2]>450)
[1] 3 4 5
> a[,3][which(a[,3]>100)]
[1] 271 103 735
> a[,3][ a[,3]>100]
[1] 271 103 735
```

Porém, a maneira mais prática para acessar colunas de um *data frame* é através da utilização da função `attach` (que pode ser desfeita com a função `detach`). Eu não vou entrar no debate ideológico sobre se é melhor usar `attach` ou `with`, etc, - se você está interessado nessa discussão, vá para a lista R-Help ou leia `?with`. . . Você não obtém nenhum resultado, mas você pode agora acesse qualquer coluna com seu nome:

Código em R

```
> attach(a)
> CLASS
[1] "open" "open" "open" "closed" "closed"
Levels: closed open
```

Observe duas coisas importantes. Primeiro, se você anexar um *data frame* que tenha um ou mais nomes que já tenham sido definidos como estruturas de dados ou como colunas de *data frames* anexados anteriormente, você receberá um aviso. Em tais casos, verifique se você está realmente lidando com as estruturas de dados ou colunas desejadas e considere usar `detach` para desanexar o *data frame* mais antigo. Segundo, quando você usa `attach`, você está, a rigor, usando 'cópias' dessas variáveis. Você pode alterá-las, mas essas alterações não afetam o *data frame* de onde elas provêm.

Código em R

```
> CLASS[4]=NA;CLASS
[1] "open" "open" "open" NA "close"
> a
> a
POS TOKENFREQ TYPEFREQ CLASS
1 ADJ 421 271 open
2 ADV 337 103 open
3 SUBS 1411 735 open
4 CONJ 458 18 closed
5 PREP 455 37 closed
```

Vamos modificar Class para o que estava anteriormente:

Código em R

```
> CLASS[4]<-"closed"
```

Se você deseja modificar o *data frame* “a”, você deve executar as mudanças diretamente na *data frame*, utilizando `a$TOKENFREQ[2]=338` ou `a$CLASS[4]=NA`, por exemplo. No entanto, como vimos na seção 2.4.3, isso só é fácil com vetores e fatores aos quais você não adiciona novos níveis – caso deseje incluir um novo nível, você deve primeiro defini-lo.

Às vezes, você vai querer investigar apenas parte de um *data frame* – talvez um grupo específico de linhas, ou de colunas, ou até mesmo uma matriz específica. Pode haver casos também em que o *data frame* é tão grande, que você vai querer manter apenas parte dela na memória do seu computador. Como sempre, existem várias formas de acessar parte de um *data frame*. Uma forma é através da utilização de índices entre colchetes acompanhados de funções lógicas, ou diretamente com a função `which`. Se você tiver utilizado a função `attach`, você poderá utilizar os nomes das colunas diretamente

Código em R

```
b=a[CLASS=="open",]; b
POS TOKENFREQ TYPEFREQ CLASS
1    ADJ      421      271 open
2    ADV      337      103 open
3    SUBS     1411      735 open
> b=a[a[,4]=="open",]; b
POS TOKENFREQ TYPEFREQ CLASS
1    ADJ      421      271 open
2    ADV      337      103 open
3    SUBS     1411      735 open
```

(Obviamente, você também pode digitar `b=a[a$CLASS=="open",]`). Em outras palavras, você determina todos os elementos da coluna chamada “CLASS” / a quarta coluna que está “open” e então utiliza essa informação para acessar as linhas que deseja (por isso a vírgula antes do último colchete). Existe uma maneira mais elegante de acessar partes de um *data frame*: através da utilização do comando `subset`. Essa função pede dois argumentos: o *data frame* que deseja acessar parte dela e as condições que o subconjunto do *data frame* deve cumprir. Então a próxima linha cria a mesma estrutura de b:

Código em R

```
> b=subset(a, CLASS=="open")
```

Você pode também utilizar várias condições ao mesmo tempo.

Código em R

```
> b=subset(a, CLASS=="open" & TOKENFREQ<1000); b
POS TOKENFREQ TYPEFREQ CLASS
1    ADJ      421      271 open
2    ADV      337      103 open
> b=subset(a, POS %in% c("ADJ", "ADV")); b
POS TOKENFREQ TYPEFREQ CLASS
1    ADJ      421      271 open
2    ADV      337      103 open
```

Como mencionei anteriormente, geralmente você irá utilizar um editor de texto ou um software de planilhas para editar seu *data frame*. Mas vale a pena mencionar que o programa R permite que você modifique o seu *data frame* como se ele estivesse em uma planilha. A função `fix` pede como principal

argumento um *data frame* e então abre um editor de planilhas com o qual você pode modificar o *data frame*; você pode até adicionar novos níveis para os fatores já existentes sem a necessidade de defini-los anteriormente (o programa R os define para você).

Finalmente, vamos agora analisar como podemos ordenar um *data frame*. Você lembra que a função `order` cria um vetor com as posições dos elementos e que esse vetor pode ser utilizado para ordenação. Suponhamos que você queira ordenar o *data frame* a de acordo com a coluna CLASS (em ordem alfabética crescente), e dentre CLASS ordenar de acordo com a TOKENFREQ (em ordem decrescente). Como fazer isso?

Pausa para pensar

Um problema aqui é que os dois estilos de ordenação são diferentes: Um é `decreasing=FALSE`, enquanto o outro é `decreasing=TRUE`. Nesse caso, você não aplica a função `order` ao TOKENFREQ, mas apenas para os valores negativos de TOKENFREQ.

Código em R

```
> order.index=order(CLASS, -TOKENFREQ); order.index
[1] 3 1 2 4 5
```

Uma vez que isso é feito, você pode utilizar o vetor `order.index` para ordenar a *data frame*:

Código em R

```
> a[order.index,]
POS TOKENFREQ TYPEFREQ CLASS
POS TOKENFREQ TYPEFREQ CLASS
4 conj      458      18 closed
5 prep      455      37 closed
3  n      1411     735 open
1  adj      421     271 open
2  adv      337     103 open
```

Naturalmente, você pode fazer isso de uma só vez:⁴

Código em R

```
> a[order(CLASS, -TOKENFREQ), ]
```

Você pode agora utilizar também a função `sample` para ordenar as linhas de um *data frame* de maneira aleatória (por exemplo, como fizemos anteriormente para aleatorizar uma tabela com itens experimentais). Primeiro, você determina o número de linhas que quer aleatorizar (com a função `dim`) e então combina as funções `sample` e `order`:

⁴ Observe que o R é superior a muitos outros programas neste sentido, porque o número de parâmetros de ordenação é em princípio ilimitado.

Código em R

```

> no.rows=nrow(a)
> order.index = sample(no.rows); order.index
[1] 5 3 1 2 4
> a[order.index,]
> a[order.index,]¶
POS TOKENFREQ TYPEFREQ CLASS
3 n 1411 735 open
4 conj 458 18 closed
1 adj 421 271 open
2 adv 337 103 open
5 prep 455 37 closed
> a[sample(nrow(a)),] # como na linha um¶

```

Mas como fazer se quisermos ordenar um *data frame* de acordo com vários fatores – alguns em ordem crescente e outros em ordem decrescentes? Neste caso, você não pode utilizar valores negativos de níveis de fatores – não faria sentido digitar -“open”, por exemplo. Assim, podemos utilizar a função `rank`, a qual primeiro coloca os níveis de fatores em ordem, e depois utilizar os valores negativos dessa classificação:

Código em R

```

> order.index<-order(-rank(CLASS), -rank(POS))¶
> a[order.index,]¶
POS TOKENFREQ TYPEFREQ CLASS
3 n 1411 735 open
2 adv 337 103 open
1 adj 421 271 open
5 prep 455 37 closed
4 conj 458 18 closed

```

Recomendação para estudo/leitura

- a função `is.data.frame` testa se a estrutura de dados é um *data frame*
- a função `dim` fornece o número de linhas e colunas de um *data frame*
- a função `read.csv` e `read.csv2` permitem ler arquivos tabulados
- a função `save` que salva estrutura de dados em fomato binário compactado
- a função `with` para acessar colunas de dados sem utilizar `attach`
- as funções `cbind` e `rbind` para combinar vetores e fatores em colunas ou em linhas
- a função `merge` que combina *data frames* diferentes
- a função `complete.cases` para verificar quais linhas do *data frame* possuem dados faltantes / NA

2.7 Um pouco sobre programação: condicionais e loops

Até agora, focamos em funções simples e existentes, mas quase não exploramos a linguagem de programação do R. Esta seção introduzirá algumas noções muito poderosas que permitem que você faça o R decidir qual de duas ou mais coisas especificadas por você o programa deve fazer e/ou se deve fazer algo mais de uma vez. Na Seção 2.7.1, exploraremos a primeira possibilidade, e na Seção 2.7.2 discutiremos a segunda, mas o tratamento aqui só pode ser muito breve e eu recomendo que você explore algumas das sugestões de leitura para mais detalhes.

2.7.1 Expressões condicionais

Futuramente, você vai muitas vezes se ver em situações nas quais você quer explorar uma das várias opções possíveis em uma análise estatística. Em um gráfico, por exemplo, os pontos de dados para os indivíduos do sexo masculino devem ser traçados em azul e os pontos de dados para as mulheres devem ser traçados em rosa. Ou, digamos que você realmente só quer que o R gere um gráfico quando o resultado é significativo. Em geral, você pode naturalmente sempre fazer estas coisas passo a passo: você poderia decidir-se para cada análise se ela é significativa e então gerar um gráfico quando ela for. No entanto, uma maneira mais elegante é escrever um código do R que tome decisões para você, que você pode aplicar a qualquer conjunto de dados, e que, portanto, permite que você recicle o código de uma análise para a próxima. Expressões condicionais são uma das maneiras - outras estão disponíveis e às vezes são mais elegantes - para fazer o R decidir o que fazer. É assim que a sintaxe em uma notação muitas vezes referida como pseudo-código pode parecer (assim, não há necessidade de digitar isso no R!):

Código em R

```
if (alguma expressão lógica testando uma condição ) {  
  o que fazer se a expressão lógica avaliada é verdadeira (TRUE)  
  (aqui pode-se ter mais de uma linha)  
} else if (outra expressão lógica) {  
  o que fazer se a outra expressão lógica avaliada for verdadeira (TRUE)  
  (aqui pode-se ter mais de uma linha)  
} else {  
  o que fazer se todas as expressões lógicas acima avaliadas forem falsas (FALSE) }
```

É isso, e a parte depois do primeiro } é opcional. Abaixo segue um exemplo com um código real (lembre-se de que "\n" significa 'uma nova linha'):

Código em R

```
> pvalue=0.06  
> if (pvalue>=0.05) {  
+ cat("Not significant, p =", pvalue, "\n")  
+ } else {  
+ cat("Significant, p =", pvalue, "\n")  
+ }  
Not significant, p = 0.06
```

A primeira linha define um valor-p, o qual você obterá mais tarde por meio de um teste estatístico. A próxima linha testa se esse valor-p é maior ou igual a 0,05. Ele é, e é por isso que o código após o primeiro { é executado e o R nunca consegue ver a parte depois do else.

Se você agora definir o valor-p para 0,04 e executar a expressão condicional (`if`) novamente, então é isso que acontece: a linha 2 acima testa se 0,04 é maior ou igual a 0,05. Como não é, o bloco de código entre `{ e }` antes do `else` é ignorado e o segundo bloco do código é executado. Tente você mesmo.

Uma versão curta desse mesmo procedimento pode ser extremamente útil quando você tem muitos testes para fazer, mas apenas uma instrução para todos eles quando um teste retorna `TRUE` ou `FALSE`. Essa versão usa a função `ifelse`, representada esquematicamente abaixo:

Código em R

```
ifelse(expressão lógica, que fazer quando TRUE, que fazer quando FALSE)
```

Uma aplicação dessa versão:

Código em R

```
> pvalues=c(0.02, 0.00096, 0.092, 0.4)¶
> decisions==ifelse (pvalues<0.05, "*", "ns")¶
> decisions¶
[1] "*" "*" "ns" "ns"
```

Como você pode observar, o `ifelse` testou todos os quatro valores-p em relação ao limite de 0.05 e colocou os valores requeridos correspondentes no novo vetor *decisions*. Vamos usar muito esse procedimento para personalizar gráficos.

2.7.2 Loops

Loops são poderosos para fazer com que o R execute uma ou muitas funções múltiplas vezes. Assim como muitas outras linguagens de programação, o R tem diferentes tipos de *loops*, mas eu apenas discutirei *for-loops* aqui. Esta é a sintaxe geral em pseudo-código:

Código em R

```
for (para cada elemento de uma sequência) {
  o que fazer com cada elemento da sequência
  (aqui pode-se ter mais de uma linha)
}
```

Vamos examinar isso passo a passo. A estrutura de dados `some.name` representa qualquer nome que você possa querer atribuir a uma estrutura de dados que é processada no *loop*, e `a.sequence` significa qualquer coisa que pode ser interpretada como uma sequência de valores, mais tipicamente um vetor de extensão 1 ou mais. Isso soa mais enigmático do que realmente é, aqui vai um exemplo muito fácil:

Código em R

```
> for (counter in 1:3) {¶
+ cat("This is iteration number", counter, "\n")¶
+ }¶
This is iteration number 1
This is iteration number 2
This is iteration number 3
```

Quando o R executa o *for-loop*, ele atribui a variável `counter` o primeiro valor da sequência 1:3, isto é, 1. Depois, na única linha do *loop*, o R imprime uma frase e a termina com o valor atual de `counter`, 1 e uma quebra de linha. Em seguida, o R chega ao `}` e, como a variável `counter` ainda não foi executado por completo sobre todos os valores da variável `a.sequence`, o programa repete a iteração, o que significa que ele volta ao início do *loop*, desta vez atribuindo a variável `counter` o próximo valor de `a.sequence`, ou seja, 2, e assim por diante. Uma vez que o R imprimiu a terceira linha, ele sai do loop porque a variável `counter` já foi executado em todos os elementos da variável `a.sequence`.

Aqui está um exemplo mais avançado, mas que é típico do uso que vamos fazer de *loops* mais tarde. Você é capaz de ver o que esse exemplo faz com base no código abaixo?

Código em R

```
> some.numbers=1:100
> collector=vector(length=10)
> for (i in 1:10) {
+ collector[i]=mean(sample(some.numbers, 50))
+ }
> collector
[1] 50.78 51.14 45.04 48.04 55.30 45.90 53.02 48.40 50.38 49.88
```

Pausa para pensar

A primeira linha gera um vetor `some.numbers` com valores de 1 a 100. A segunda linha gera um vetor chamado `collector`, o qual tem 10 elementos e será usado para coletar os resultados do *looping*. A linha 3 começa um *loop* de 10 iterações, usando um vetor chamado `i` como o contador. A linha 4 é a crucial agora: nela, o R seleciona 50 números aleatoriamente sem substituição do vetor `some.numbers`, calcula a média desses 50 números, e depois armazena essa média na `i`-ésima posição do vetor `collector`. Na primeira iteração, `i` é, naturalmente, 1, de modo que a primeira média é armazenada na primeira posição do vetor `collector`. Em seguida, o R itera, `i` se torna 2, o R gera uma segunda amostra aleatória, calcula sua média e a armazena na - agora - segunda posição do vetor `collector`, e assim por diante, até que o R tenha feito o processo de amostragem, cálculo de média e armazenado 10 valores e sai do loop. Em seguida, o vetor `collector` é impresso na tela.

No Capítulo 4, usaremos uma abordagem como esta para nos ajudar a explorar dados que violam alguns dos pressupostos de testes estatísticos comuns. No entanto, já vale a pena mencionar que *loops* muitas vezes não são a melhor maneira de fazer coisas como acima no R. Diferentemente de algumas outras linguagens de programação, o R é projetado de tal forma que ele é muitas vezes muito mais rápido e apresenta uma memória mais eficiente para fazer as coisas com os membros da família de funções `apply` (os quais você vai conhecer um pouco mais tarde), não com *loops*. Mesmo assim, ser capaz de escrever rapidamente um *loop* e testar algo é muitas vezes uma habilidade muito útil.

Recomendação para estudo/leitura

- Recomendações de estudo As funções `next` e `break` para controlar o comportamento de/em *loops*

2.8 Escrevendo suas próprias funções

O fato de que o R não é apenas um software de estatística, mas sim uma linguagem de programação completa, é algo que dificilmente pode ser supercitado o suficiente. Isso significa que quase tudo é possível: o limite do que você pode fazer com R não é definido pelo que os programadores de algum software pensaram que você pode querer fazer - o limite é definido apenas pelas suas habilidades e talvez pela sua memória RAM e processador (e essa é uma razão pela qual eu recomendo usar R para análises da Linguística de Corpus, ver [Gries \(2009a\)](#)). Um aspecto que torna isso particularmente óbvio é como você pode muito facilmente escrever suas próprias funções para facilitar e/ou automatizar tarefas tediosas e/ou frequentes. Nesta seção, vou dar alguns exemplos muito simples da lógica de como escrever suas próprias funções, principalmente porque ainda não lidamos com nenhuma função estatística. Não se desespere se você não entender essas questões de programação imediatamente - para a maior parte deste livro, você não vai precisar delas, mas esses recursos podem ser muito úteis quando você começar a lidar com dados mais complexos. Além disso, no Capítulo 3 e no Capítulo 4, eu vou voltar a este tópico para que você ganhe mais prática nisso, e terminar com uma lista de funções úteis para a sua própria pesquisa.

O primeiro exemplo que eu quero usar envolve examinar uma parte de uma estrutura de dados. Por exemplo, vamos supor que você carregou um vetor muito longo (digamos, com 10,000 elementos) e deseja verificar se você o importou para o R corretamente.

Imprimir simplesmente isso na tela pode ser um pouco tedioso, já que você não poderá ler todos os 10.000 itens (muito menos na velocidade com que eles são exibidos), nem geralmente precisa de todos os 10.000 itens - os primeiros `n` são geralmente o suficiente para ver se sua importação de dados foi bem-sucedida. O mesmo é verdadeiro para estruturas de dados longas: você não precisa ver todas as 1.600 linhas para verificar se a importação foi bem-sucedida, talvez as primeiras 5 ou 6 são suficientes. Vamos escrever uma função `peek` que por padrão mostra os primeiros 6 elementos de cada uma das estruturas de dados que você conhece: vetores unidimensionais ou fatores e estruturas de dados bidimensionais.

Uma boa maneira de lidar com a escrita de funções é considerar primeiro como você resolveria essa questão para apenas uma estrutura de dados particular, por exemplo, fora do contexto da escrita de funções e, em seguida, tornar qualquer código que você escreveu geral o suficiente para cobrir não apenas a estrutura de dados que você acabou de abordar, mas muito além dela. Para essa finalidade, vamos primeiro carregar uma estrutura de dados para este pequeno exemplo (a partir de `<_inputfiles/02-7_dataframe1.csv>`):

Código em R

```
> into.causatives = read.delim(file.choose())
> str(into.causatives)
'data.frame': 1600 obs. of 5 variables:
 $ BNC: Factor w/ 929 levels "A06","A08","A0C",...: 1 2 3 4 ...
 $ TAG_ING: Factor w/ 10 levels "AJ0-NN1","AJ0-VVG",...:
10 7 10 ...
 $ ING: Factor w/ 422 levels "abandoning","abdicating",...: 354 49 382 ...
 $ VERB_LEMMA: Factor w/ 208 levels "activate","aggravate",...: 76 126 186 ...
 $ ING_LEMMA : Factor w/ 417 levels "abandon","abdicate",...: 349 41 377 ...
```

Agora, você deseja trabalhar com vetores, fatores e estruturas de dados unidimensionais e bidimensionais. Como você obteria os primeiros seis elementos de cada um deles? Isso você já sabe. Para vetores ou fatores você escreveria:

Código em R

```
vector.or.factor[1:6]
```

e para estruturas de dados você escreveria:

Código em R

```
data.frame[1:6,]
```

Portanto, você precisa basicamente decidir qual é a estrutura de dados da qual o R deve exibir os primeiros n elementos (por padrão 6) e, em seguida, você faz a seleção com `cl[1:6]` ou `[1:6,]`. Uma vez que, em última análise, a idéia é fazer com que o R - não você - decida sobre o caminho certo de fazer essa seleção (dependendo da estrutura de dados), você usa uma expressão condicional:

Código em R

```
> if (is.data.frame(into.causatives)) {
> into.causatives[1:6,]
> } else {
into.causatives[1:6]
> }
```

	BNC	TAG_ING	ING	VERB_LEMMA	ING_LEMMA
1	A06	VVG	speaking	force	speak
2	A08	VBG	being	nudge	be
3	A0C	VVG	taking	talk	tak
4	A0F	VVG	taking	bully	take
5	A0H	VVG	trying	influence	try
6	A0H	VVG	thinking	delude	think

Para transformar isso em uma função, você envolve uma definição de função (nomeando o função `peek`) em torno deste pedaço de código. No entanto, se você usar o código acima como está, então esta

função usará o nome `into.causatives` na definição da função, o que não é exatamente muito geral. Como você viu, muitas funções do R usam `x` para a variável obrigatória principal. Seguindo esta tradição, você poderia escrever o seguinte:

Código em R

```
> peek = function (x) {  
>   if(is.data.frame(x)) {  
>     x[1:6,]  
>   } else {  
>     x[1:6]  
>   }  
> }  
> peek(into.causatives)
```

Ou seja, o R define uma função chamada `peek` que requer um argumento, chamado de `x` dentro da função. Quando você chamar `peek` com algum argumento – por exemplo, `into.causatives` – o R tomará o conteúdo dessa estrutura de dados e, para a duração da execução da função, atribuirá esse conteúdo a `x`. Em seguida, dentro da função, o R realizará todas as `peek` com `x` e retornará o resultado, que corresponde às primeiras 6 linhas de `into.causatives`.

Parece que chegamos ao final. Entretanto, faltam ainda algumas coisas. Quando você escreve uma função, é crucial que você tenha certeza de que ela cobre todos os tipos de possibilidades ou dados que você possa querer usar com ela. Afinal, você está escrevendo uma função que torna a sua vida mais fácil, para permitir que você não tenha que se preocupar com as mesmas coisas novamente depois de já ter pensado sobre isso uma vez, ou seja, quando você escreveu a função. Há três maneiras pelas quais o código acima deve ser melhorado:

- E se a estrutura de dados com a qual você aplica a função `peek` não for um vetor, um fator ou uma estrutura de dados?
- E se você quiser ser capaz de ver não 6, mas `n` elementos?
- E se a estrutura de dados com a qual você aplicar a função `peek` tiver menos que `n` elementos ou fileiras?

Para abordar a primeira possibilidade, precisamos apenas adicionar outra expressão condicional. Até agora nós apenas testamos se o que usamos com a função `peek` é um *data frame* - agora nós também precisamos verificar se, caso não seja um *data frame*, isso é um vetor ou fator, e, idelamente, retornamos algum alerta se a estrutura de dados não for nenhum dos três.

Para abordar a segunda possibilidade, precisamos ser capazes de dizer para a função, de forma flexível, quantas partes de `x` queremos ver, e a forma como dizemos isso a uma função é, naturalmente, por meio de seus argumentos. Assim, adicionamos um argumento (vamos chamá-lo de `n`) que diz o quanto queremos ver de `x`, e definimos 6 como padrão.

Para abordar a possibilidade final, temos que garantir que o R sabe quantos elementos `x` tem: se ele tem mais que `n`, o R deve mostrar `n`, mas se tem menos que `n`, o R deve mostrar todos que pode, isto é, todos eles.

A versão da função `peek` abaixo aborda todas estas questões:

Código em R

```
> peek = function (x, n=6) {  
>   if (is.data.frame(x)) {  
>     return(x[1:min(nrow(x), n),])  
>   } else if (is.vector(x) | is.factor(x)) {  
>     return(x[1:min(length(x), n)])  
>   } else {  
>     cat("Not defined for other data structures ...\n")  
>   }  
> }
```

A questão número um é endereçada pela adição de uma segunda condicional com o `else if` test - lembre-se do uso de `|` para significar 'ou' - e retorno de uma mensagem se `X` não for nem um vetor, fator ou uma estrutura de dados.

A questão número dois é endereçada pela adição de uma argumento `n` à definição da função e utilização de `n` no corpo da função. O padrão do argumento `n` é definido como 6, portanto, se o usuário não especificar `n`, 6 é usado, mas o usuário também pode substituir esse padrão por outro número.

A questão final é endereçada pelo ajuste do subconjunto: em vez de usar apenas `n`, usamos 1: o mínimo de `n` ou o número de elementos que `x` tem. Portanto, se `x` tiver mais do que `n` elementos, então `n` será o mínimo e veremos `n` elementos. Se `x` tiver menos que `n` elementos, então esse número de elementos será o mínimo e conseguiremos vê-los todos.

Finalmente, observe também que agora estou usando a função `return` para especificar exatamente o que a função `peek` deve retornar e apresentar como *output* para o usuário quando tiver terminado. Experimente executar as seguintes linhas (o *output* não é mostrado aqui, veja os comentários no arquivo de código) para veja se funciona:

Código em R

```
> peek(into.causatives)  
> peek(into.causatives, 3)  
> peek(into.causatives, 9)  
> peek(21:50, 10)  
> peek(into.causatives$BNC, 12)  
> peek(as.matrix(into.causatives))
```

Embora tudo isso possa não parecer fácil e que vale o esforço, veremos mais tarde que ser capaz de escrever suas próprias funções irá facilitar algumas análises estatísticas. Deixe-me também mencionar que o exemplo acima foi irônico: já existe na verdade uma função no R que faz o que a função `peek` faz (e mais, porque pode lidar com mais estruturas de dados) - pesquise sobre `head` e `tail` ;-).

Agora você deve fazer os exercícios relativos ao Capítulo 2...

Recomendação para estudo/leitura

- As funções `NA`, `is.na`, `NaN`, `is.nan`, `na.action`, `na.omit`, e `na.fail` para como lidar com dados que estão faltando
- [Ligges \(2005\)](#), [Crawley \(2007\)](#), [Braun e Murdoch \(2008\)](#), [Spector \(2008\)](#), [Gentleman \(2009\)](#), e [Gries \(2009a\)](#) para mais informações sobre o R: [Ligges \(2005\)](#), [Braun e Murdoch \(2008\)](#), e [Gentleman \(2009\)](#) sobre o R como linguagem de programação estatística, [Crawley](#) como uma

síntese completa, [Spector \(2008\)](#) sobre manipulação de dados no R, e [Gries \(2009a\)](#) sobre métodos da Linguística de Corpus com o R

Estatística descritiva

Qualquer linguista do século 21 terá que ler e compreender modelos matemáticos, bem como compreender métodos estatísticos de análise. Se você está interessado na métrica de Shakespeare, na percepção sociolinguística de identidade, em violações de concordâncias verbais do hindu, ou na percepção da duração da vogal, o uso da matemática como uma ferramenta de análise já está aqui e sua prevalência só vai crescer ao longo das próximas décadas. Se você não estiver preparado para ler artigos que envolvam o termo Bayesiano, ou ($p < 0,01$), médias-k clustering, intervalo de confiança, análise semântica latente, distribuições bimodais e unimodais, n-gramas, etc, então você será apenas um convidado tímido na festa da linguística.
(<http://thelouslylinguist.blogspot.com/2010/01/why-linguists-should-study-math.html>)

Nesse capítulo, vamos aprender a obter resultados descritivos. Na seção 3.1, vamos falar sobre estatística univariada, ou seja, a estatística que resume a distribuição de uma variável, de um vetor e de um fator. Na Seção 3.2, vamos discutir estatística bivariada, ou seja, a estatística que caracteriza a relação entre duas variáveis, dois vetores e dois fatores. As duas seções irão também apresentar formas de representar os dados graficamente; outros gráficos adicionais serão apresentados no Capítulo 4 e 5.

3.1 Estatística Univariada

3.1.1 Dados de frequência

A maneira mais simples de representar a distribuição de seus dados é provavelmente através de tabelas de frequência, ou seja, uma lista que diz com qual frequência cada um dos dados ocorreu. Fazer uma tabela de frequência no R é extremamente fácil. Vamos ver um exemplo. Suponhamos que você extraiu todas as ocorrências das partículas ‘uh’, ‘uhm’, e ‘silence’ e anotou para cada uma delas, se (1) foi produzida por um homem ou por uma mulher, (2) se ela ocorreu em um monólogo ou um diálogo e (3) o tempo que ela durou – medido em milésimos de segundos. Primeiramente, vamos importar esses dados para o programa R. Eles estão no arquivo `<_inputfiles/03-1_uh(m).txt>`.

Código em R

```
> UHM<-read.delim(file.choose())
> str(UHM)
'data.frame':
1000 obs. of 5 variables:
 $ CASE : int 1 2 3 4 5 6 7 8 9 10 ...
 $ SEX
 : Factor w/ 2 levels "female","male": 2 1 1 1 2 ...
 $ FILLER: Factor w/ 3 levels "silence","uh",...: 3 1 1 3 ...
 $ GENRE : Factor w/ 2 levels "dialog","monolog": 2 2 1 1 ...
 $ LENGTH: int 1014 1188 889 265 465 1278 671 1079 643 ...
> attach(UHM)
```

Para verificar com qual frequência cada disfluência ocorre, basta utilizar a função `table`, que cria uma lista de frequência de um fator ou vetor:

Código em R

```
>table(FILLER)
FILLER
silence      uh      uhm
332         394      274
```

Para saber o percentual (ou frequência relativa) de ocorrência de cada uma das disfluências, você pode fazer isso manualmente ou utilizar o comando `prop.table`, cujo argumento é uma tabela gerada pela função `table` e que retorna a percentagem das frequências da tabela (verifique abaixo).

Código em R

```
> table(FILLER)/length(FILLER)
FILLER
silence uh uhm
0.332 0.394 0.274
> prop.table(table(FILLER))
FILLER
silence uh uhm
0.332 0.394 0.274
```

Muitas vezes é útil gerar uma tabela com as frequências acumuladas, ou os percentuais acumulados. O R tem uma função `cumsum` que adiciona de maneira sucessiva os valores de um vetor e retorna todas as somas. Veja o exemplo a seguir:

Código em R

```
> 1:5
[1] 1 2 3 4 5
> cumsum(1:5)
[1] 1 3 6 10 15
```

Assim, você pode utilizar a função `cumsum` para as tabelas de frequência:

Código em R

```
> cumsum(table(PARTICULA))  
silence      uh      uhm  
332      726      1000  
> cumsum(prop.table(table(PARTICULA)))  
silencio      uh      uhm  
0.332      0.726      1.000
```

Você vai frequentemente precisar representar a distribuição dos seus dados de maneira gráfica, e as próximas seções apresentam algumas maneiras de fazer isso. Por razões de espaço, apenas podemos discutir algumas maneiras de produzir gráficos, mas você pode ler a ajuda das funções apresentadas (usando ? seguida pelo nome da função) e [Murrell \(2011\)](#) para maiores informações.

3.1.1.1 Gráficos de dispersão e gráficos de linha

Antes de discutirmos como representar vetores e fatores graficamente, vamos ver como os elementos de um vetor são representados individualmente em um gráfico. A maneira mais simples utiliza a função `plot`. Essa é uma função bem versátil que, a depender dos argumentos que você utiliza, vários gráficos diferentes podem ser produzidos. (O que vamos apresentar pode parecer um pouco confuso, mas você verá que é uma forma muito prática de produção de gráficos). Se você utilizar como argumento dessa função apenas um vetor numérico, então o R produz um gráfico de dispersão, ou seja, um gráfico bi-dimensional onde os valores do vetor são representados no eixo-y e a ordem com a qual eles aparecem é representada no eixo-x. Exemplo:

Código em R

```
> a=c(1, 3, 5, 2, 4); b=1:5  
> plot(a)·#·gráfico da esquerda na Figura 15
```

Figura 15 – Gráficos de dispersão simples

Mas se você fornecer dois vetores como argumentos da função, os valores do primeiro vetor serão representados pelo eixo-x e os valores do segundo vetor serão representados pelo eixo-y (os nomes dos vetores serão utilizados para nomear os eixos):

Código em R

```
> plot(a, b) # gráfico da direita na Figura 15¶
```

Com o argumento `type=...`, você determina o tipo de gráfico que deseja produzir. A definição padrão (utilizada nos comandos acima porque não especificamos outra coisa) é `type="p"` (para pontos). Se você utilizar `type="b"` (para o inglês *both* [ambos]), você obtém um gráfico com pontos e linhas conectando os pontos; se você utilizar `type="l"` (para linhas), você obtém um gráfico de linha; veja a Figura 16. (Com a definição `type="n"`, você produz apenas as coordenadas, sem nenhum dado.)

Código em R

```
> plot(b, a, type="b") # gráfico da esquerda na Figura 16¶
> plot(b, a, type="l") # gráfico da direita na Figura 16¶
```

Figura 16 – Gráficos de linhas simples

Você também pode definir os nomes dos eixos (`xlab="..."` e `ylab="..."`), um título para o gráfico (`main="..."`), o intervalo de valores de cada eixo (`xlim=...` e `ylim=...`), e a adição de uma matriz (`grid()`). Com o argumento `col="..."`, você consegue definir a cor dos elementos representados no gráfico, como você verá mais frequentemente abaixo.

Código em R

```
> plot(b, a, xlab="VETOR b", ylab="VETOR a", xlim=c(0, 8), ylim=c(0, 8),
type="b"); grid() # Figura 17¶
```

Um ponto importante que devemos ressaltar é que os limites dos eixos devem ser escolhidos de forma que a representação dos dados seja feita da maneira mais significativa possível. É sempre importante incluir a coordenada (0, 0) para certificar-se de que, em caso de comparações entre gráficos, os intervalos são os mesmos. Por exemplo, imagine que você queira comparar os intervalos de dois vetores *x* e *y* em dois gráficos distintos. Nesse caso, você não deve deixar que o R decida o intervalo desses vetores. Considere os dois gráficos superiores na Figura 18.

A nuvem de pontos parece bem semelhante nos dois gráficos. No entanto, quando você olha o intervalo do eixo-y, você nota que há uma diferença grande. O gráfico da esquerda superior apresenta intervalo de 0 a 2, enquanto o da direita superior apresenta intervalo de 0 a 6. Essa diferença fica mais

Figura 17 – Gráfico de dispersão exemplificando algumas definições extras

óbvia quando se manualmente alterarmos os limites do eixo-y através do argumento `ylim=...`. Observe os dois gráficos inferiores na Figura 18.

Nota importante: toda vez que você utiliza a função `plot`, o comportamento padrão do R é que um novo gráfico seja produzido e o anterior perdido (No RStudio, você pode voltar a gráficos anteriores com a tecla de seta ou pelo menu *Plots: ...*). Se você quiser produzir duas linhas em um único gráfico, você deve primeiro criar um gráfico com a função `plot` (e `type="l"` ou `type="b"`) e então adicionar o segundo gráfico com a função `points` (ou `lines`; às vezes você poderá também utilizar o argumento `add=TRUE`). Isso quer dizer que você deve definir o intervalo dos eixos do primeiro gráfico de forma que o segundo gráfico possa também ser representado nele.

Vamos ver um exemplo para deixar essa ideia mais clara. Suponhamos que queira produzir um gráfico com os vetores `m` e `n`, e então representar no mesmo gráfico os elementos dos vetores `x` e `y`, mas isso não funciona, como você pode observar no lado esquerdo da Figura 19.

Código em R

```
> m=1:5; n=5:1
> x=6:10; y=6:10
> plot(m, n, type="b"); points(x, y, type="b"); grid()
```

O gráfico da esquerda na Figura 19 mostra os pontos do vetor `m` e `n`, mas não os pontos do vetor `x` e `y`, pois os intervalos que o R utilizou para os vetores `m` e `n` são muito pequenos para os valores de `x` e `y`. Assim, você deve criá-los manualmente quando estiver criando o primeiro sistema de coordenadas. Uma forma de fazer isso é através da utilização da função `max`, que retorna o maior valor de um vetor (e `min` que retorna o menor valor). O gráfico da direita na Figura 19 mostra como os quatro vetores podem ser representados em um mesmo gráfico. (No comando, colocamos o valor mínimo como 0 manualmente – poderíamos ter utilizado os comandos `min(m, x)` e `min(n, y)`, mas queríamos incluir (0, 0) no gráfico.)

Figura 18 – Gráficos de dispersão e a importância da definição correta dos intervalos

Figura 19 – Gráficos de dispersão e a importância da definição correta dos intervalos

Código em R

```
> plot(m, n, type="b", xlim=c(0, max(m, x)), ylim=c(0, max(n, y)), xlab="Vetores m  
e x", ylab="Vetores n e y"); grid()¶  
> points(x, y, type="b")¶
```

Recomendação para estudo/leitura

As funções `pmin` e `pmax` que determina o mínimo e o máximo em cada posição de vetores diferentes. (Exemplo `pmin(c(1, 5, 3), c(2, 4, 6))`)¶

3.1.1.2 Gráficos de pizza

A função que cria gráficos de pizza é `pie`. O argumento mais importante dessa função é a tabela gerada pela função `table`. Você pode modificar os nomes das categorias se quiser (com o argumento `labels=...`) ou modificar a cor com o argumento `col=...`, etc.:

Código em R

```
> pie(table(FILLER), col=c("grey20","grey50","grey80"))¶
```

Figura 20 – Um gráfico de pizza com a frequência de ocorrência das disfluências

Para usar cores diferentes com `col=...` como acima, você tem que saber quantas cores existem e atribuir nomes a elas, o que pode se tornar trabalhoso com muitas cores e/ou gráficos. Para situações como estas, a função `rainbow` pode ser muito útil. Em seu uso mais simples, essa função requer apenas um argumento, o número de cores diferentes que você deseja. Assim, como você reescreveria a linha acima para o gráfico de pizza de tal forma que você deixe o R descobrir quantas cores são necessárias em vez de dizer `col=rainbow(3)`?

Pausa para pensar

Deixe o R usar tantas cores quanto o número de elementos presentes na tabela para a qual você está gerando o gráfico:

Código em R

```
> pie(table(FILLER), col=rainbow(length(table(FILLER))))¶
```

Note que os gráficos de pizza não são geralmente uma boa maneira de resumir dados, porque os seres humanos não são muito bons em inferir quantidades a partir de ângulos. Portanto, *pie* não é uma função que você deve usar com muita frequência - a função *rainbow*, por outro lado, é um que você deve definitivamente ter em mente.

3.1.1.3 Gráficos de barra

Para criar um gráfico de barra, você deve utilizar a função `barplot`. Novamente, o argumento mais importante dessa função é uma tabela criada a partir da função `table`. Assim, como no gráfico anterior, você pode utilizar as definições padrão ou modificá-las. Para definir seus próprios nomes para as categorias, você deve utilizar o argumento `names.arg=...`, e não `labels=...` (veja Figura 21).

Código em R

```
> barplot(table(FILLER)) # gráfico da esquerda na Figura 21¶
> barplot(table(FILLER), col=c("grey20", "grey40", "grey60")) # gráfico da direita na Figura 21
```

Figura 21 – Um gráfico de barra com a frequência de ocorrência das partículas

Uma maneira interessante de configurar um gráfico de barra é utilizar o argumento `space=0`, o qual produz as barras umas próximas das outras. Isso facilita caso queira incorporar mais dados ao seu gráfico. Você pode também fazer com que as frequências observadas apareçam dentro das barras através da função `text`. O primeiro argumento da função `text` é um vetor com as coordenadas no eixo-x do texto que deve ser impresso (com `space=0`, 0.5 para a primeira barra, 1.5 para a segunda barra e 2.5 para a terceira barra), o segundo argumento é um vetor com as coordenadas no eixo-y do texto a ser impresso (metade de cada frequência observada de forma que o texto fique dentro das barras), e `labels=...` define o texto que deve ser impresso; cf. o gráfico à esquerda da Figura 22.

Código em R

```
> barplot(table(FILLER), col=c("grey40", "grey60", "grey80"),
names.arg=c("Silencio", "Uh", "Uhm"), space=0)¶
> text(c(0.5, 1.5, 2.5), table(FILLER)/2, labels=table(FILLER))¶
```

A segunda maneira de criar um gráfico semelhante - cf. o gráfico à direita da Figura 22 - envolve algumas alterações úteis:

Código em R

```
> mids<-barplot(table(FILLER), col=c("grey40", "grey60","grey80"))
> text(mids, table(FILLER), labels=table(FILLER), pos=1)
```

Figura 22 – Gráficos de barra com as frequências impressas

A primeira linha agora não apenas gera o gráfico de barras, mas também indica o que R manda para uma estrutura de dados chamada *mids*, a qual contém as coordenadas *x* dos pontos médios das barras, que podemos então usar para *texting* (Veja *mids*.) Em segundo lugar, a segunda linha agora usa *mids* para as coordenadas *x* do texto a ser impresso e usa *pos=1* para fazer o R imprimir o texto um pouco abaixo das coordenadas especificadas; *pos=2*, *pos=3* e *pos=4* imprimiriam o texto um pouco para a esquerda, acima, e à direita das coordenadas especificadas, respectivamente.

As funções *plot* e *text* permitem que você crie um gráfico poderoso. Primeiro você cria um gráfico que contenha apenas os eixos e seus nomes (utilize *type="n"*), e depois com *text* você coloca nomes ou números. Tente isto para uma ilustração de um tipo de gráfico que você verá com mais frequência abaixo:

Código em R

```
> tab<-table(FILLER)
> plot(tab, type="n", xlab="Disfluencies", ylab="Observed
frequencies", xlim=c(0, 4), ylim=c(0, 500)); grid()
> text(seq(tab), tab, labels=tab)
```

Recomendação para estudo/leitura

a função *dotchart* para o gráfico de pontos e os parâmetros *cex*, *srt*, *col*, e *font* para ajustar os gráficos: *?par*.

3.1.1.4 Gráficos de Pareto

Uma outra maneira de representar as frequências das partículas é através do gráfico de Pareto. Nesse tipo de gráfico, as frequências das categorias observadas são representadas como um gráfico de barras, mas as barras são colocadas primeiramente em ordem decrescente de frequência e depois uma linha representando as frequências relativas acumuladas é produzida juntamente com as barras. A função *pareto.chart* está disponível no pacote *qcc* que você deve instalar e/ou carregar antes de utilizá-lo. Veja a Figura 47.

Código em R

```
> library(qcc)
> pareto.chart(table(FILLER),main=)
Pareto chart analysis for table(FILLER)
Frequency Cum.Freq. Percentage Cum.Percent.
uh          394      394      39.4      39.4
silence     332      726      33.2      72.6
uhm         274     1000      27.4     100.0
```

Figura 23 – Gráfico de Pareto representando as frequências das partículas

3.1.1.5 Histogramas

Enquanto gráficos de barra são provavelmente os tipos de gráficos mais utilizados para representar frequências de dados nominais/categoriais, histogramas são mais amplamente utilizados para intervalos/razão de variáveis. No R, histogramas são facilmente criados a partir da função `hist`, a qual requer como argumento apenas o vetor em questão.

Código em R

```
> hist(LENGTH)
```

Para fazer com o gráfico fique mais apresentável, como na Figura 48), cujo gráfico da esquerda contém um histograma da variável `LENGTH` com nomes dos eixos e barras cinza.

Código em R

```
> hist(LENGTH, main=, xlab="Length in ms", ylab=
"Frequency", xlim=c(0,2000), ylim=c(0, 100),
col="grey80")
```

O gráfico da direita na Figura 48 contém um histograma com as densidades de probabilidade (criadas com o argumento `freq=FALSE`) e sua respectiva curva (gerada por `lines`):

Código em R

```
> hist(LENGTH, main=, xlab="Length IN MS", ylab="Density", freq=FALSE, xlim=c(0,
2000), col="grey50")¶
> lines(density(LENGTH))¶
```

Figura 24 – Histogramas para as frequências de durações de disfluências

Com o argumento `breaks=...`, você pode definir no R o número de barras específicas que deseja no seu histograma. Você pode definir com um número inteiro ou então fornecer um vetor com os intervalos de cada barra. Para definir o número de barras, no entanto, precisamos saber qual número é apropriado para um dado conjunto de dados. Em geral, as pessoas não escolhem mais que 20 barras. Você pode também utilizar a fórmula proposta em (14) (conferir Keen 2010:143–160 para discussão) para auxiliá-lo na definição de quantas barras você irá precisar no seu histograma. O mais importante é que a escolha não represente mal a distribuição dos seus dados.

(15) Números de barras para um histograma com n dados = $1 + 3,32 \log_{10} n$

3.1.2 Distribuições cumulativas empíricas

Uma visualização muito útil de dados numéricos é o gráfico de distribuição acumulação empírica (função, abreviada como `ecdf`¹), um exemplo do qual você já viu como parte do gráfico de Pareto na Seção 3.1.1.4. Sobre o eixo x de um gráfico `ecdf`, você encontrará o intervalo da variável que é visualizada, no eixo y você encontrará uma escala de porcentagem de 0 a 1 (=100%), e os pontos no sistema de coordenadas mostram quanto um valor de variável e todos os outros valores menores à esquerda desse valor representam em porcentagem relativa a todos os dados. A figura 25 mostra tal gráfico para `LENGTH` e você pode ver que aproximadamente 18% de todos os comprimentos são menores do que 500 ms.

Esse gráfico é muito útil porque ele não perde informações descartando dados: cada ponto de dados é representado no gráfico, razão pela qual gráficos `ecdf` podem ser muito reveladoras mesmo para dados que a maioria dos outros gráficos não podem ilustrar bem. Vamos ver se você entendeu esse gráfico: como ficam os gráficos `fdce` de dados normalmente distribuídos e de dados uniformemente distribuídos?

Pausa para pensar

Você encontrará a resposta no arquivo de código (com gráficos); certifique-se de que você entende o porquê para você poder usar este tipo de gráfico muito útil.

¹ empirical cumulative distribution

Figura 25 – Gráfico ecdf das extensões das disfluências

Recomendação para estudo/leitura

- as funções `dotchart` e `stripchart` (com o `method="jitter"`) para representar a distribuição de pontos específicos
- a função `scatterplot` (do pacote `library(car)`) para gráficos de dispersão mais sofisticados
- as funções `plot3d` e `scatterplot3d` (dos pacotes `rgl` e `scatterplot3d`) para gráficos de dispersão em três dimensões

3.1.3 Medidas de tendência central

Medias de tendência central são provavelmente as estatísticas (estimativas) mais utilizadas em análise de dados. Elas são valores que têm como objetivo resumir o comportamento de uma variável. Em outras palavras, imagine que você deva utilizar apenas um número para representar um conjunto de dados. Que número você usaria? As medidas de tendência central fornecem esse número. A escolha de qual medida utilizar vai depender da natureza dos seus dados. Para dados categoriais/nominais, a melhor medida de tendência central é a moda, para dados ordinais, a melhor medida é a mediana. Para dados intervalares e razões, a melhor medida é a média aritmética.

3.1.3.1 Moda

A moda de uma variável ou distribuição é o valor que foi observado com a maior frequência. Até onde eu saiba, não existe uma função pré-determinada para calcular a moda no R, mas você pode facilmente encontrar a moda com as seguintes funções. Por exemplo, a moda de `FILLER` é `uh`:

Código em R

```
> which.max(table(FILLER))  
uh  
2  
> max(table(FILLER))  
[1] 394
```

Cuidado quando há mais de um nível que exibe o número máximo de observações - a tabulação é geralmente mais segura.

3.1.3.2 Mediana

A melhor medida de tendência central para dados ordinais é a mediana, que é o valor que se obtém quando todos os dados são colocados em ordem de tamanho e você escolhe aquele que divide essa distribuição ao meio. Por exemplo, a mediana da sequência 1, 2, 3, 4, 5 é 3. Se o intervalo consistir de um número ímpar, a mediana será a média entre os dois valores que dividem a distribuição ao meio.

Código em R

```
> median(LENGTH)¶  
[1] 897
```

3.1.3.3 Média aritmética

A medida de tendência central mais conhecida é a média aritmética. A média é calculada da seguinte maneira: soma-se todos os valores de uma distribuição ou vetor e divide-se a soma pela quantidade de valores presentes na soma. No R, é claro, existe uma função que calcula a média automaticamente:

Código em R

```
> sum(LENGTH)/length(LENGTH)¶  
[1] 915.043  
> mean(LENGTH)¶  
[1] 915.043
```

Uma desvantagem da média aritmética é que ela é muito sensível a *outliers* (ou seja, valores extremos). Veja o exemplo:

Código em R

```
> a=1:10; a¶  
[1] 1 2 3 4 5 6 7 8 9 10  
> b=c(1:9, 1000); b¶  
[1] 1 2 3 4 5 6 7 8 9 1000  
> mean(a)¶  
[1] 5.5  
> mean(b)¶  
[1] 104.5
```

Apesar dos vetores a e b terem praticamente os mesmos elementos (exceto por um), a média do vetor b é muito maior que a média do vetor a, simplesmente por causa de um valor extremo (*outlier*). Na verdade, a média de b não representa os valores entre 1 e 9 e nem mesmo o *outlier* 1000, ou seja, a média nesse caso não é uma boa medida de tendência central. Existem duas maneiras de resolver problemas como esse. Primeiro: você pode utilizar o argumento `trim=...` na função `mean`: nesse argumento, você define o percentual de elementos no começo e no final da distribuição que devem ser descartados antes do cálculo da média. Os comandos a seguir calculam a média dos vetores a e b após a eliminação dos maiores e dos menores valores da distribuição:

Código em R

```
> mean(a, trim=0.1)
[1] 5.5
> mean(b, trim=0.1)
[1] 5.5
```

Segundo: você pode simplesmente utilizar a mediana como medida de tendência central, o que é uma boa alternativa quando a distribuição dos dados para a qual se quer calcular a média não está de acordo com a curva normal.

Código em R

```
> median(a); median(b)
[1] 5.5
[1] 5.5
```

Conselho

Não é porque o R ou qualquer outro editor de planilhas retorna uma média (ou mediana) com várias casas decimais que você precisa reportar todas elas. Utilize o número de casas decimais que fizer sentido para o seus dados.

3.1.3.4 Média geométrica

A média geométrica é utilizada para calcular as médias de fatores ou proporções (enquanto que a média aritmética é calculada para obter a média das somas). Suponhamos que você grave seis ocasiões de uma criança: aos 2;1 (dois anos e 1 mês), 2;2, 2;3, 2;4, 2;5, e 2;6. Em cada ocasião, você computou a frequência cumulativa de palavras diferentes que essa criança produziu (frequência de tipo). Você representa essas frequências em um vetor de nome lexico:

Código em R

```
> lexicon=c(132, 158, 169, 188, 221, 240)
> names(lexicon)=c("2;1", "2;2", "2;3", "2;4", "2;5", "2;6")
```

Você agora quer calcular a média de crescimento do léxico dessa criança. Primeiramente você deve calcular o crescimento de uma idade para a outra:

Código em R

```
> increases=lexicon[2:6]/lexicon[1:5]; increases
2;2 2;3 2;4 2;5 2;6
1.196970 1.069620 1.112426 1.175532 1.085973
```

O resultado acima quer dizer o seguinte: aos dois anos e dois meses, essa criança produziu 19,697% mais tipos diferentes de palavras que antes (aos dois anos e um mês). Aos dois anos e três meses ela produziu 6,962% mais tipos de palavras que a na fase anterior (dois anos e dois meses) e assim sucessivamente. No entanto, é errado achar que a média de crescimento é a média aritmética dos respectivos crescimentos entre as fases:

Código em R

```
> mean(increases)·#·ERRADO¶
[1] 1.128104
```

Você pode facilmente testar que este não é o resultado correto. Se esse número fosse a taxa média real de aumento, então o produto de 132 (o primeiro tamanho do léxico) e essa taxa de 1,128104 à potência de 5 (o número de vezes que a suposta “taxa média” se aplica) deve ser o valor final de 240. Este, porém, não é o caso:

Código em R

```
> 132*mean(increases)^5¶
[1] 241.1681
```

Ao invés disso, você deve calcular a média geométrica. A média geométrica de um vetor x com n elementos é calculada a partir da seguinte fórmula (3.1.3.4), e veremos que essa é a média correta:

$$(16) \quad \text{média}_{geom} = (x_1 \cdot x_2 \cdot \dots \cdot x_{n-1} \cdot x_n)^{\frac{1}{n}}$$

Código em R

```
> rate.increase=prod(increases)^(1/length(increases)); rate.increase¶
[1] 1.127009
> 132*rate.increase^5¶
[1] 240
```

É verdade que a diferença entre o valor correto – 240 – e o valor incorreto – 241.1681 – é pequena. Mas o valor 241.1681 é ainda incorreto e essa diferença nem sempre será pequena. O exemplo na página Wikipédia (procure Média Geométrica) ilustra essa discrepância: Se você faz um experimento e encontra um taxa de crescimento de 10,000 e em um segundo experimento você encontra uma taxa de crescimento de 0,0001, a média aritmética será aproximadamente 5,000, enquanto a média geométrica será 1.²

Finalmente, vamos ressaltar mais uma vez como pode ser mais útil exibir em gráficos as palavras ou números e ao invés de pontos, triângulos, ... Tente gerar a Figura 26, na qual a posição das palavras com relação ao eixo-y corresponde à duração média das disfluências (exemplo: 928,4 para mulheres, 901,6 para homens, etc.). (A linha horizontal representa a média geral - talvez você não saiba ainda fazer isso). Apenas uma inspeção rápida no gráfico revela várias tendências: os homens estão abaixo da média, enquanto as mulheres estão acima da média, as disfluências silenciosas têm aproximadamente a duração média, etc.

3.1.4 Medidas de dispersão

Muitas pessoas conhecem as medidas de tendência central. O que muitas pessoas não sabem no entanto é que você nunca – NUNCA – deve reportar uma medida de tendência central sem a sua medida de dispersão correspondente. Sem a medida de dispersão, você nunca vai saber se a sua medida de tendência central é uma boa representação da distribuição dos seus dados. Vamos ver um exemplo para deixar essa ideia mais clara. Suponhamos que você mediu, durante um ano, a média mensal da temperatura de duas cidades:

² Uma maneira alternativa de calcular a média geométrica: `exp(mean(log(crescimento)))¶`.

Figura 26 – Duração média das partículas

Código em R

```
> town1=c(-5, -12, 5, 12, 15, 18, 22, 23, 20, 16, 8, 1)
> town2=c(6, 7, 8, 9, 10, 12, 16, 15, 11, 9, 8, 7)
> mean(town1);mean(town2)
[1] 10.25
[1] 9.833333
```

Com base apenas nas médias, as duas cidades parecem ter uma situação climática muito semelhante, mas até mesmo uma observação rápida da Figura 27 mostra que isso não é verdade. Apesar das médias semelhantes, eu sei para onde eu gostaria de ir em fevereiro. Obviamente, a média da cidade 2 representa a tendência central da cidade muito melhor que a média da cidade 1: a variação das temperaturas da cidade 1 oscila muito mais em torno da média quando comparadas com as temperaturas da cidade 2. Assim, é sempre importante relatar a média de tendência geral juntamente com uma medida de dispersão: para a moda, a melhor medida de dispersão é a entropia relativa; para a mediana e dados intervalares que não apresentam uma distribuição normal ou que exibem *outliers*, e o desvio padrão ou a variância para o intervalo normal/razão dos dados.

Figura 27 – Curvas de temperatura para as duas cidades

3.1.4.1 Entropia Relativa

Uma medida de dispersão simples para dados categóricos é a entropia relativa H_{rel} . H_{rel} é igual a 1 quando os níveis da variável categórica em questão são igualmente frequentes, e é igual a 0 quando todos os dados pertencem ao mesmo nível. Para variáveis categóricas com n níveis, H_{rel} é calculada pela fórmula apresentada em (3.1.4.1), onde p_i corresponde à frequência (em percentual) de cada um dos níveis da categoria:

$$(17) \quad H_{rel} = - \frac{\sum_{i=1}^n (p_i \ln p_i)}{\ln n}$$

Suponhamos que você contabilize os artigos de 300 sintagmas nominais e encontre 164 casos que não possuem nenhum determinante, 33 casos com artigo indefinido e 103 casos com artigo definido. H_{rel} será computado da seguinte maneira:

Código em R

```
> article=c(164, 33, 103)
> perc=article/sum(article)
> hrel=-sum(perc*log(perc))/log(length(perc)); hrel
[1] 0.8556091
```

É importante notar que a fórmula acima não produz o resultado desejado de 0 quando apenas casos sem determinantes são observados porque $\log(0)$ não está definido:

Código em R

```
> article=c(300, 0, 0)
> perc=article/sum(article)
> hrel=sum(perc*log(perc))/log(length(perc)); hrel
[1] NaN
```

Geralmente, esse problema é resolvido com a definição do $\log(0)$ para zero (ou às vezes somando 1 em cada um dos dados antes de calcular o logaritmo). Esse é um caso onde escrever uma função para calcular logaritmos que podem lidar com 0s pode ser útil. Por exemplo, definimos uma função `logw0` que define o valor do logaritmo do valor 0 e então calculamos a entropia relativa normalmente:

Código em R

```
> logw0=function(x) {
+   ifelse(x>0, log(x), 0)
+ }
> hrel=sum(perc*logw0(perc))/logw0(length(perc)); hrel
[1] 0
```

Vamos falar com mais detalhes sobre distribuição de dados categóricos na Seção 4.1.1.2.

3.1.4.2 Amplitude

A maneira mais simples de mensurar dispersão de dados intervalares ou razões é através da amplitude, ou seja, da diferença entre o maior e o menor valor da distribuição. Você pode simplesmente utilizar a função `range`, a qual requer como argumento o vetor com os valores que deseja calcular a amplitude e então calcular a diferença entre os valores com o comando `diff`, ou calcular você mesmo a amplitude a partir dos valores mínimo e máximo:

Código em R

```
> range(LENGTH)¶
[1] 251 1600
> diff(range(LENGTH))¶
[1] 1349
> max(LENGTH)-min(LENGTH)¶
[1] 1349
```

Essa medida é muito simples de calcular, mas também muito sensível: um *outlier* é suficiente para criar resultados pouco significativos. Por esse motivo, a amplitude é pouco utilizada como medida de dispersão.

3.1.4.3 Quantis e Quartis

Uma outra maneira simples, porém útil e poderosa, de medir a dispersão envolve o cálculo dos quantis de uma distribuição. Nós já vimos quantis anteriormente no contexto de distribuições de probabilidades na Seção 1.3.4. Teoricamente, para calcular os quantis você precisa ordenar os valores em ordem crescente, e então encontrar os valores que dividem os $x\%$, $y\%$ inferiores dos dados; quando esses percentuais são 25%, 50%, e 75%, eles são chamados quartis. No R, você pode utilizar a função `quantile`, veja o exemplo a seguir (note a opção `type=1`):

Código em R

```
> a=1:100¶
> quantile(a, type=1)¶
0% 25% 50% 75% 100%
1 25 50 75 100
```

Se você escrever os números inteiros de 1 a 100, o número 25 irá dividir a distribuição de forma que os 25% inferiores estarão antes dele. O valor que divide a distribuição ao meio (50%) é a mediana e os valores 0% e 100% são o mínimo e o máximo, respectivamente. Vamos ver alguns outros argumentos dessa função. O primeiro, `probs`, permite que você defina outros percentuais: O segundo, `type=...`, permite que você escolha outras formas como os quantis devem ser computados. Para distribuições discretas, `type=1` é provavelmente a melhor maneira de calcular quantis. Para variáveis contínuas, a definição padrão `type=7` é a melhor.

Código em R

```
> quantile(a, probs=c(0.05, 0.1, 0.5, 0.9, 0.95), type=1)¶
5% 10% 50% 90% 95%
5 10 50 90 95
```

No final das contas, quanto mais o quartil 25% e o quartil 75% se diferirem um do outro, mais heterogênea é a distribuição dos seus dados. Podemos ver isso olhando os dados das temperaturas das cidades: vamos calcular o intervalo interquartil – a diferença entre o quartil 75% e o quartil 25%. O intervalo interquartil dos dados da cidade 1 é muito maior que o intervalo interquartil dos dados da cidade 2.

Código em R

```
> quantile(town1)
0% 25% 50% 75% 100%
-12.0 4.0 13.5 18.5 23.0
> IQR(town1)
[1] 14.5
> quantile(town2)
0% 25% 50% 75% 100%
6.00 7.75 9.00 11.25 16.00
> IQR(town2)
[1] 3.5
```

Podemos também utilizar as funções que aprendemos até agora às durações das disfluências:

Código em R

```
> quantile(LENGTH, probs=c(0.2, 0.4, 0.5, 0.6, 0.8, 1), type=1)
20% 40% 50% 60% 80% 100%
519 788 897 1039 1307 1600
```

Isto é, os 20% centrais de todas as durações de disfluências estão entre 788 e 1.039 (como você pode verificar com `sort(LENGTH)[401:600]`, 20% dos comprimentos são menores ou iguais a 519, 20% dos valores são 1.307 ou maiores, etc.

Uma aplicação interessante do `quantile` é usá-lo para dividir vetores de variáveis contínuas em grupos. Por exemplo, se você quiser dividir o vetor `LENGTH` em 5 grupos de faixas de valores praticamente iguais, você pode usar a função `cut`, a qual divide vetores em grupos, e a função `quantile`, a qual instrui `cut` sobre como o grupo deve ser. Ou seja, existem 200 valores de `LENGTH` entre e incluindo 251 e 521 etc.

Código em R

```
> LENGTH.GRP<-cut(LENGTH, breaks=quantile(LENGTH, probs=
c(0, 0.2, 0.4, 0.6, 0.8, 1)), include.lowest=TRUE)
> table(LENGTH.GRP)
LENGTH.GRP
[251,521] (521,789] (789,1.04e+03]
200 200 200
(1.04e+03,1.31e+03] (1.31e+03,1.6e+03]
203 197
```

3.1.4.4 Desvio médio

Uma outra maneira de caracterizar a dispersão de uma distribuição é através do cálculo do desvio médio. Você primeiro calcula a diferença absoluta entre cada um dos dados e a média da distribuição (veja a função `abs`), e então você calcula a média dessas diferenças. Para os dados da temperatura da cidade 1, o desvio médio é 9,04:

Código em R

```
> town1
[1] -5 -12 5 12 15 18 22 23 20 16 8 1
> town1-mean(town1)
[1] -15.25 -22.25 -5.25 1.75 4.75 7.75 11.75 12.75 9.75 5.75
-2.25 -9.25
> abs(town1-mean(town1))
[1] 15.25 22.25 5.25 1.75 4.75 7.75 11.75 12.75 9.75 5.75 2.25 9.25
> mean(abs(town1-mean(town1)))
[1] 9.041667
> mean(abs(town2-mean(town2)))
[1] 2.472222
```

Para as durações das disfluências, temos:

Código em R

```
> mean(abs(LENGTH-mean(LENGTH)))
[1] 329.2946
```

Apesar de ser uma medida aparentemente intuitiva e fácil de calcular, poucas pessoas ainda a utilizam. Feliz ou infelizmente (ver [Gorard \(2004\)](#)), você encontrará mais frequentemente a medida de dispersão discutida a seguir, o desvio padrão.

3.1.4.5 O desvio padrão/variância

O desvio padrão sd de uma distribuição x com n elementos é definida pela fórmula em (3.1.4.5). A princípio a fórmula pode parecer difícil, mas o desvio padrão é conceptualmente semelhante ao desvio médio. Para calcular o desvio médio, você calcula a diferença entre cada um dos dados e a média, e calcula o valor absoluto dessas diferenças. Para o desvio padrão, você calcula a diferença entre cada dado e a média, e então eleva ao quadrado todas essas diferenças (e depois de dividir a soma desses quadrados por $n - 1$, você calcula a raiz quadrada).

$$(18) \quad sd = \left(\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \right)^{\frac{1}{2}}$$

Vamos traduzir a fórmula para o R para deixá-la mais clara:

Código em R

```
> town1
[1] -5 -12 5 12 15 18 22 23 20 16 8 1
> town1-mean(town1)
[1] -15.25 -22.25 -5.25 1.75 4.75 7.75 11.75 12.75 9.75 5.75 -2.25 -9.25
> (town1-mean(town1))^2
[1] 232.5625 495.0625 27.5625 3.0625 22.5625 60.0625 138.0625 162.5625 95.0625
33.0625 5.0625 85.5625
> sum((town1-mean(town1))^2)
[1] 1360.25
> sum((town1-mean(town1))^2)/(length(town1)-1)
[1] 123.6591
> sqrt(sum((town1-mean(town1))^2)/(length(town1)-1))
[1] 11.12021
```

Há, claro, uma maneira mais fácil...

Código em R

```
> sd(town1)
[1] 11.12021
> sd(town2)
[1] 3.157483
```

Nota interessante: o desvio padrão é a raiz quadrada de uma outra medida de dispersão, a variância, a qual pode ser calculada a partir da função `var`.

Recomendação para estudo/leitura

a função `mad` que calcula uma outra medida de dispersão importante: a mediana do desvio médio

3.1.4.6 O coeficiente de variação

Apesar de o desvio padrão ser provavelmente uma medida de dispersão mais difundida, ele tem um ponto fraco: o seu valor é altamente dependente da média da distribuição, como você pode ver no exemplo abaixo:

Código em R

```
> sd(town1)
[1] 11.12021
> sd(town1*10)
[1] 111.2021
```

Quando os valores, e consequentemente a média, são aumentados por uma ordem de magnitudes, o desvio padrão também é alterado. Por isso, você nunca deve comparar os desvios padrões de distribuições que contêm médias diferentes (a não ser que você normalize a distribuição). Quando você divide o desvio padrão pela média de uma distribuição, você encontra o coeficiente de variação. Note como o coeficiente de variação não é afetado pela multiplicação por 10. Note também que a cidade 1 tem um grau de dispersão maior quando comparada com a cidade 2.

Código em R

```
> sd(town1)/mean(town1)¶
[1] 1.084899
> sd(town1*10)/mean(town1*10)¶
[1] 1.084899
> sd(town2)/mean(town2)¶
[1] 0.3210999
```

3.1.4.7 Funções de *Summary*

Para obter várias estatísticas que resumem os valores de um vetor (ou fator), você pode utilizar a função `summary`, cujo resultado retornado é auto-explicativo.

Código em R

```
> summary(town1)¶
Min. 1st Qu. Median Mean 3rd Qu. Max.
-12.00 4.00 13.50 10.25 18.50 23.00
```

Um tipo de gráfico muito útil na descrição dos seus dados é o que chamamos de `boxplot`. Na sua forma mais simples, a função `boxplot` requer como argumento apenas o vetor a partir do qual se deseja fazer o gráfico. No entanto, adicionamos também `notch=TRUE`, o qual explicarei mais adiante, bem como linhas com pequenos sinais de “+” para indicar o local das médias. Observe que estou atribuindo a saída de `boxplot` a uma estrutura de dados chamada `boxsum` para inspeção posterior.

Código em R

```
> boxplot(town1, town2, notch=T, names=c("Town 1", "Town 2"))
> text(1:2, c(mean(town1), mean(town2)), c("+", "+"))
```

Esse tipo de gráfico, veja a Figura 28, tem muitas informações úteis:

- as linhas em negrito representam as medianas dos dois vetores;
- as linhas horizontais que marcam os limites mínimo e máximo representam as franjas (aproximadamente os quartis 75% e 25%);
- as linhas pontilhadas verticais representam os valores os valores mínimo e máximo que não ultrapassam o intervalo de 1,5 interquartil;
- cada *outlier* que ocorre fora desse intervalo é representado por um ponto;
- Os degraus (notches) à direita e à esquerda das caixas estendem-se no intervalos de $\pm 1.58 * IQR / \sqrt{n}$: se os degraus dos dois boxplots se sobrepõem, então eles provavelmente não serão significativamente diferentes.

A Figura 28 mostra que as temperaturas médias das duas cidades são bem semelhantes e provavelmente não são estatisticamente diferentes uma da outra. Você pode também notar que a dispersão das temperaturas da cidade 1 é maior que a dispersão das temperaturas da cidade 2. Muitas vezes, um `boxplot` é suficiente para mostrar que qualquer análise subsequente é desnecessária. *Boxplots* são extremamente úteis e serão frequentemente usados nos próprios capítulos. No entanto, existem situações nas quais o gráfico `ecdf` (gráfico de probabilidade acumulada) introduzido acima é melhor, e o exemplo

Figura 28 – *Boxplot* das temperaturas das duas cidades

a seguir é modelado com base no que aconteceu em um conjunto de dados real de um aluno que eu supervisionei. Execute o código no arquivo de código e considere a figura 29.

Como você pode ver no arquivo de código, eu criei um vetor `x1` que realmente contém dados de duas distribuições muito diferentes, enquanto que o vetor `x2` contém dados de apenas uma distribuição mais ampla.

Figura 29 – *Boxplot* (painel da esquerda) e o gráfico de probabilidade acumulada (painel da direita) de dois vetores

Crucialmente, os *boxplots* não revelam isso de forma alguma. Sim, o segundo *boxplot* mais escuro é mais amplo e tem alguns outliers, mas o fato de que o primeiro *boxplot* mais leve representa um vetor contendo dados de duas distribuições diferentes é completamente ausente no gráfico. Por outro lado, os

gráficos ecdf à direita mostram isso muito claramente: a linha mais escura para o segundo vetor aumenta gradativamente de uma maneira que sugere uma distribuição normal, enquanto que a linha mais clara para o primeiro vetor mostra que ele contém duas distribuições normais, dados os dois segmentos de curva em forma de s. Assim, enquanto o gráfico ecdf não é tão intuitivamente compreensível como um *boxplot*, ele pode ser muito mais informativo.

Recomendação para estudo/leitura

as funções `hdr.boxplot` (`library(hdrcde)`), `vioplot` (`library(vioplot)`), e `bpplot` (`library(Hmisc)`) para alguns *boxplots* alternativos.

3.1.4.8 Erro padrão

O erro padrão de uma média aritmética é definido como o desvio padrão de várias médias derivadas de amostras do mesmo tamanho tiradas aleatoriamente de uma população com reposição. Vamos clarear essa definição: suponhamos que você colete uma amostra aleatória de uma população (uma variável qualquer) e calcule a média dessa amostra. Essa média provavelmente não será igual à média da população. Caso você colete uma outra amostra aleatória de mesmo tamanho da mesma população e também calcule a média, é muito provável que as médias não serão iguais. Se você tirar várias (10 mil, por exemplo) amostras aleatórias do mesmo tamanho de uma população com reposição, e calcular a média de cada uma delas, o desvio padrão de todas essas médias será o erro padrão.

Código em R

```
> means<-vector(length=10000)
> for (i in 1:10000)
+ means[i]<-mean(sample(LENGTH, size=1000, replace=TRUE))
+
> sd(means)
[1] 12.10577
```

O erro padrão de uma média é calculado a partir da fórmula em (3.1.4.8), e com base em (3.1.4.8) você já pode notar que quanto maior for o erro padrão de uma média, menor é a probabilidade de que essa média seja representativa da média da população, e que quanto maior for o tamanho da amostra, menor será o erro padrão:

$$(19) \quad se_{média} = \sqrt{\frac{var}{n}} = \frac{sd}{\sqrt{n}}$$

Assim, o erro padrão da duração média das disfluências do nosso exemplo é mostrado abaixo, e é muito semelhante ao nosso resultado de reamostragem acima:

Código em R

```
> mean(LENGTH)
[1] 915.043
> sqrt(var(LENGTH)/length(LENGTH))
[1] 12.08127
```

Você pode também calcular o erro padrão para estatísticas além da média aritmética, mas o único exemplo adicional que analisaremos aqui é o erro padrão da frequência relativa p , o qual pode ser calculado a partir da fórmula (19):

$$(20) \quad se_{percentual} = \sqrt{\frac{p(1-p)}{n}}$$

Desta forma, o erro padrão do percentual de todos os “silêncios” entre as disfluências coletadas (33,2% de 1.000 disfluências) é:

Código em R

```
> prop.table(table(FILLER))  
FILLER  
silence uh uhm  
0.332 0.394 0.274  
> sqrt(0.332*(1-0.332)/1000)  
[1] 0.01489215
```

Vamos entender melhor a importância do erro padrão na Seção 3.1.6, quando o utilizaremos para calcular intervalos de confiança. Note que quando você compara as médias de duas amostras aproximadamente do mesmo tamanho e seus intervalos médias $\pm$ erros padrão se sobrepõem, então você sabe que as médias da amostra não são significativamente diferentes. No entanto, se esses intervalos não se sobrepõem, isso não mostra que as médias são significativamente diferentes (ver [Crawley \(2005, p. 169\)](#)). No Capítulo 5, veremos como calcular o erro padrão das diferenças entre médias, as quais podem ser calculados pela fórmula (20).

$$(21) \quad \text{se}_{\text{diferença entre duas médias}} = \sqrt{SE_{\text{média_grupo1}}^2 + SE_{\text{média_grupo2}}^2}$$

Conselho

Erros padrão só são úteis se os dados para os quais ele se aplica são distribuídos de maneira normal ou com o tamanho da amostra $n \geq 30$.

3.1.5 Centralização e padronização (escores-z)

Muitas vezes precisamos comparar valores oriundos de diferentes escalas de medida. Um exemplo ([BORTZ, 2005](#)): se um aluno X teve um aproveitamento de 80% em um curso e um outro aluno Y teve um aproveitamento de 60% em um outro curso, você poderia dizer que o aluno X teve um aproveitamento melhor do que o aluno Y? Por um lado, sim: 80% é melhor que 60%. Por outro lado, as avaliações que o aluno X fez podem ter sido mais fáceis do que as avaliações feitas pelo aluno Y. Logo, é importante e necessário relativizar/padronizar as notas dos dois alunos com base na performance deles nos respectivos cursos. (Você encontrou uma situação semelhante acima na Seção 3.1.3.6 quando você aprendeu que nem sempre é apropriado comparar diretamente diferentes desvios padrão.) Suponhamos que as notas obtidas nos dois cursos pelos dois alunos sejam:

Código em R

```
> grades.course<-rep((seq(0, 100, 20)), 1:6); grades.course.X  
[1] 0 20 20 40 40 40 60 60 60 60 80 80 80 80 80 100 100 100 100 100 100  
> grades.course.Y<-rep((seq(0, 100, 20)), 6:1); grades.course.Y  
[1] 0 0 0 0 0 0 20 20 20 20 20 40 40 40 40 60 60 60 80 80 100
```

Uma forma de padronizar as notas é a partir da centralização delas, o que envolve subtrair a média do curso de cada valor individual dentro desse mesmo curso.

Código em R

```
> a=1:5
> centered.scores = a-mean(a); centered.scores
[1] -2 -1 0 1 2
```

Podemos ver como esses escores se relacionam com os valores originais em “a”: uma vez que a média de “a” é 3, os primeiros dois valores centralizados são negativos (são menores que a média de a), o terceiro valor é 0 (não se desvia da média de a), e os dois últimos são positivos (são maiores que a média de a).

Uma outra maneira mais sofisticada de padronizar escores é através de padronização, ou seja, através da transformação dos valores a serem comparados nos chamados escores-z, os quais indicam quantos desvios padrão cada valor do vetor apresenta em relação à média do vetor. O escore-z de um valor de um vetor é a diferença entre esse valor e a média do vetor, dividido pelo desvio padrão do vetor. Você pode calculá-lo manualmente da seguinte maneira:

Código em R

```
> z.escores=(a-mean(a))/sd(a); z.escores
[1] -1.2649111 -0.6324555 0.0000000 0.6324555 1.2649111
```

A relação entre os escores-z e os valores originais de a é muito semelhante à relação existente entre os escores centralizados e os valores originais de a: uma vez que a média de “a” é 3, os primeiros dois escores-z são negativos, pois os valores originais são menores que a média, o terceiro valor não se desvia da média (por isso o escore-z é 0), e os dois últimos são positivos, pois os valores originais são maiores do que a média de “a”. Note que a média dos escores-z é 0 e o desvio padrão é igual a 1:

Código em R

```
> mean(z.escores)
[1] 0
> sd(z.escores)
[1] 1
```

Todas as duas maneiras de padronização podem ser feitas automaticamente no R através da função `scale`, a qual requer três argumentos: o vetor que se deseja padronizar, `center=...` (o padrão é TRUE) e `scale=...` (o padrão é TRUE). Se você não definir nenhum dos argumentos – apenas o vetor – o resultado da função será uma matriz com os escores-z e com a média e desvio padrão dos escores originais:

Código em R

```
> scale(a)
[,1]
[1,] -1.2649111
[2,] -0.6324555
[3,] 0.0000000
[4,] 0.6324555
[5,] 1.2649111
attr(,"scaled:center")
[1] 3
attr(,"scaled:scale")
[1] 1.581139
```

Se você definir o argumento `scale` para `FALSE`, então você obterá uma matriz com os escores centralizados:

Código em R

```
> scale(a, scale=F)
[,1]
[1,] -2
[2,] -1
[3,] 0
[4,] 1
[5,] 2
attr(,"scaled:center")
[1] 3
```

Se utilizarmos a função acima para o nosso exemplo dos dois cursos, você irá notar que os 80% do aluno X está a apenas 0,436 desvios padrões da média e 13,33 pontos percentuais acima da média do seu curso, ao passo que os 60% do aluno Y está a 0,873 desvios padrão da média e 26,67 pontos percentuais acima da média do seu curso. Assim, apesar da nota do aluno X ser maior que a nota do aluno Y, a performance do aluno Y foi melhor que a performance do aluno X. A padronização de dados é, de fato, frequentemente útil.

3.1.6 Intervalo de Confiança

Na maioria das vezes, não é possível investigar toda a população de interesse, seja porque a população é muito grande, ou porque é muito caro investigar a população toda, ou porque isso tomaria muito tempo. Porém, apesar de você saber que amostras distintas de uma mesma população fornecem estimativas distintas da mesma população, você sempre espera que a sua amostra seja suficientemente confiável, ou seja, uma boa estimativa dos parâmetros reais da população:

- em uma amostra de 1.000 partículas de disfluência, em que a duração média delas seja 915 ms, você espera que essa média possa ser generalizada para a população de partículas de disfluência;
- em uma amostra de 1.000 partículas de disfluência em que 33,2% são “silêncios”, você espera poder generalizar esse percentual para a população e investigações futuras.

Até agora aprendemos a calcular percentuais e médias para as nossas amostras - a questão sobre como saber se essas estimativas são válidas para serem generalizadas para a população é o tópico desta seção. Na Seção 3.1.6.1, vamos aprender como calcular intervalo de confiança para médias aritméticas, e na Seção 3.1.6.2, vamos ver como calcular intervalos de confiança para percentuais. A relevância de intervalos de confiança é grande: sem eles não sabemos se podemos ou não generalizar nossas estimativas da amostra para a população; intervalos de confiança podem ser calculados para uma variedade de medidas e estimativas além daquelas discutidas aqui, não somente médias e percentuais.

3.1.6.1 Intervalos de confiança para médias aritméticas

Se você calcula a média de uma variável com base em uma amostra, você certamente espera que essa média seja representativa da média da população. No nosso exemplo, a duração média das partículas de disfluência é 915,043 ms (desvio padrão: 382,04). Sabemos, no entanto, que uma outra amostra da mesma população irá ter uma média distinta. Assim, precisamos estabelecer o nível de confiança para a média que encontramos. O chamado intervalo de confiança, que você deve sempre fornecer juntamente com a média, é o intervalo de valores em torno da média que corresponde aos valores que não diferem significativamente da média. A expressão “difere significativamente” já sugere que o intervalo de confiança é definido como sendo 1-nível de significância, ou seja, $1-0,05 = 0,95$.

Como primeiro passo, calculamos primeiro o erro padrão da média aritmética, de acordo com a fórmula em 3.1.4.8.

Código em R

```
> se=sqrt(var(LENGTH)/length(LENGTH)); se
[1] 12.08127
```

Esse valor (o erro padrão) é utilizado na fórmula (3.1.6.1) para calcular o intervalo de confiança. O parâmetro t da fórmula (3.1.6.1) refere-se à distribuição que mencionamos na Seção 1.3.4.3, e o seu cálculo requer o número de graus de liberdade. No nosso exemplo, o número de graus de liberdade df é o número de elementos do nosso vetor-1, ou seja, 999. Uma vez que você quer calcular o valor- t com base em um valor- p , você precisa utilizar a função `qt`, e uma vez que você quer um intervalo de confiança bilateral – 95% dos valores em torno da média, ou seja, valores maiores e menores que a média – você deve calcular o valor- t para 2,5% (pois 2,5% em cada lado resulta no total desejado, os 5%):

$$(22) \quad CI = \bar{x} \pm t SE$$

Código em R

```
> t.value =qt(0.025, df=999, lower.tail=F); t.value
[1] 1.962341
```

Agora sim podemos calcular o intervalo de confiança:

Código em R

```
> mean(LENGTH)-(se*t.value); mean(LENGTH)+(se*t.value)
[1] 891.3354
[1] 938.7506
```

Para fazer o mesmo cálculo de maneira mais simples, você pode utilizar a função `t.test` como o vetor de interesse e `conf.level=...` para definir o percentual desejado. O programa R irá calcular o teste de significância – vamos ver detalhes desse teste mais adiante. Por agora, nos interessa apenas o intervalo de confiança. Para isso, utilizaremos `$conf.int`:

Código em R

```
> t.test(TAMANHO, conf.level=0.95)$conf.int
[1] 891.3354 938.7506
attr(,"conf.level")
[1] 0.95
```

Este intervalo de confiança

identifica uma gama de valores dentro os quais um pesquisador pode ter 95% de certeza que irá encontrar o valor verdadeiro de um parâmetro de população (por exemplo, uma média populacional). Dito em termos probabilísticos, o pesquisador pode afirmar que existe uma probabilidade de 0.95 que o intervalo de confiança contém o valor verdadeiro do parâmetro da população (Sheskin (2011, p. 75); veja também Field, Miles e Field (2012, p. 45))³

³ Uma maneira diferente de explicar os intervalos de confiança é a seguinte: "Um erro comum é interpretar erroneamente o intervalo de confiança como uma afirmação sobre o parâmetro desconhecido [aqui, a porcentagem na população, STG]. Não é

Quando comparamos duas médias de duas amostras com tamanhos aproximadamente iguais e seus intervalos de confiança de 95% não se sobrepõem, então a diferença entre essas médias é estatisticamente significativa, o que nos leva a concluir que existe uma diferença real nas médias das duas populações. É importante ressaltar, no entanto, que duas médias podem ser estatisticamente significativas mesmo quando seus intervalos de confiança se sobrepõem (veja [Crawley \(2005, p. 169\)](#)).

3.1.6.2 Intervalo de confiança para porcentagens

A mesma lógica apresentada para média se aplica para o cálculo de intervalo de confiança para porcentagens. Dado um percentual de uma amostra, você quer saber se esse percentual é mesmo confiável para estimar o percentual real da população. No nosso exemplo, o percentual de “silêncios” na amostra foi de 33,2%. O procedimento para o cálculo do intervalo de confiança é bem semelhante. Primeiramente, calculamos o erro padrão dessa estimativa de acordo com a fórmula em (3.1.4.8), e então esse erro padrão é utilizado na fórmula em (3.1.6.2).

Código em R

```
> se=sqrt(0.332*(1-0.332)/1000); se
[1] 0.01489215
```

$$(23) \quad CI = a \pm z SE$$

O parâmetro z na fórmula acima corresponde ao escore- z mencionado na Seção 1.3.4.3 acima, o qual define 5% da área abaixo de uma distribuição normal padrão – 2,5% a partir da parte de cima e 2,5% a partir da parte de baixo.

Código em R

```
> z.score=qnorm(0.025, lower.tail=FALSE); z.score
[1] 1.959964
```

Para calcular o intervalo de confiança de 95% para o percentual de “silêncios” do nosso exemplo, você deve digitar:

Código em R

```
> z.scores=qnorm(0.025, lower.tail=FALSE)
> 0.332-z.score*se; 0.332+z.score*se
[1] 0.3611881
[1] 0.3028119
```

Uma maneira mais simples para fazer esse cálculo requer o uso da função `prop.test`, que testa se a porcentagem encontrada na amostra se difere significativamente do que era esperado. Novamente, os detalhes do teste não são relevantes ainda, mas vamos ver o intervalo de confiança que essa função retorna. O programa R precisa da frequência observada (332), do tamanho da amostra (1000) e da probabilidade desejada para o intervalo de confiança. O programa utiliza uma fórmula um pouco diferente da nossa, mas mostra praticamente o mesmo resultado.

verdade que a probabilidade de que um parâmetro seja incluído num intervalo de confiança de 95% é 95%. O que é verdadeiro é que se tirarmos um número grande de intervalos de confiança de 95 %, podemos esperar que o valor verdadeiro do parâmetro seja incluído nos intervalos calculados 95% das vezes (Good and Hardin 2012:156)

Código em R

```
> prop.test(332, 1000, conf.level=0.95)$conf.int
[1] 0.3030166 0.3622912
attr(,"conf.level")
[1] 0.95
```

Recomendação para estudo/leitura

Dalgaard (2002, Cap. 7.1 e Cap. 4.1), Crawley (2005, p. 167)

Conselho

Uma vez que o procedimento para o cálculo do intervalo de confiança se baseia no erro padrão, se os seus dados não estão distribuídos de maneira normal ou o tamanho da amostra é muito pequeno, você deve considerar a possibilidade de utilizar outros métodos para estimar intervalos de confiança (*bootstrapping*, por exemplo).

3.2 Estatística bivariada

Até agora, somente lidamos com gráficos e estatísticas que descrevem apenas uma variável, ou vetor/fator. Nesta seção, vamos ver os métodos utilizados para descrever duas variáveis e a relação entre elas. Vamos, novamente, começar com frequências, depois falaremos de médias e por fim, discutiremos correlação. Você irá notar que muitas funções das seções anteriores serão utilizadas na descrição e análise de duas variáveis.

3.2.1 Frequências e tabulação de dados

Vamos começar com duas variáveis categóricas/nominais. Geralmente, estamos interessados em saber com qual frequência de ocorrência da combinação de variáveis. A maneira mais simples de fazer isso é através da tabulação dos dados. Vamos retornar aos nossos dados iniciais sobre partículas de disfluência:

Código em R

```
> UHM=read.table(choose.files())
> attach(UHM)
```

Suponhamos que estamos interessados em saber se homens e mulheres se diferem com relação ao tipo de partículas de disfluências que eles produzem. Perguntas: Existem variáveis dependentes e independentes nesse modelo? Se sim, quais são?

Pausa para pensar

Neste caso, SEX é a variável independente e Partícula (FILLER) é a variável dependente. Calcular as frequências dessa variável em cada um dos níveis da variável independente (e suas combinações) é fácil, pois podemos utilizar a mesma função que utilizamos anteriormente: `table`. Basta que você forneça como argumento o segundo vetor (ou fator) e o programa lista os níveis do primeiro vetor nas linhas e o nível do segundo vetor nas colunas:

Código em R

```
> freqs=table(FILLER, SEX); freqs
SEX
FILLER female male
silence 171 161
uh 161 233
uhm 170 104
```

Na verdade, você pode fornecer quantos vetores/fatores quiser como argumento da função `table`, experimente fazer isso, e retomaremos essa questão mais abaixo. Da mesma forma que fizemos na Seção anterior, aqui também podemos criar tabelas de percentuais utilizando a função `prop.table`, mas para tabelas bi-dimensionais existem outras formas de calcular percentuais. Você deve especificar essas maneiras com o argumento `margin=...`. A definição padrão é `margin=NULL`, que calcula os percentuais com base no número total de elementos na tabela. Em outras palavras, a soma de todos os percentuais da tabela é igual a 1. Uma outra maneira é calcular os percentuais das linhas: defina `margin=1` e todos os percentuais das linhas somam 1. Da mesma maneira, você pode calcular percentuais para as colunas, de forma que os valores em cada coluna somem 1. Para isso, você deve definir `margin=2`. Para o nosso exemplo, essa é a melhor maneira, já que os percentuais que somam 1 são os da variável dependente.

Código em R

```
> porcents=prop.table(table(FILLER, SEX), margin=2)
> porcents
SEX
FILLER female male
silence 0.3406375 0.3232932
uh 0.3207171 0.4678715
uhm 0.3386454 0.2088353
```

Você já pode ver que os homens parecem preferir *uh* e não gostar muito de *uhm*, ao passo que as mulheres não apresentam uma preferência bem definida. No entanto, isso ainda não é um teste de significância.

A função `addmargins` mostra a frequência total para as colunas e linhas (ou outros limites definidos pelo usuário, como médias, por exemplo):

Código em R

```
> addmargins(freqs) # cf. also colSums and rowSums
SEX
FILLER female male Sum
silence 171 161 332
uh 161 233 394
uhm 170 104 274
Sum 502 498 1000
```

Recomendação para estudo/leitura

as funções `xtabs` e `ftable` para gerar outros tipos mais complexos de tabelas

3.2.1.1 Gráficos de barra e mosaicos

É claro que as tabelas que produzimos até agora podem ser representadas graficamente. A maneira mais simples de fazer isso é fornecer uma fórmula como o principal argumento da função `plot`. Tais fórmulas consistem em uma variável dependente (no nosso exemplo: `Filler: Filler`), o sinal til (“~” significa “em função de”), e uma variável independente (no nosso caso: `GENRE:GENRE`).

Código em R

```
> plot(FILLER~GENRE)¶
```

A largura e a altura das linhas, das barras e das seis caixinhas representam as frequências observadas. Por exemplo, a barra para diálogos é um pouco mais larga que a barra para monólogos porque há mais diálogos nos dados; a linha que representa *uh* é a mais larga pois essa foi a partícula mais frequente, etc.

Outras gráficos parecidos podem ser produzidos com o comando abaixo:

Código em R

```
> plot(GENRE,FILLER)¶
> plot(table(GENRE, FILLER))¶
> mosaicplot(table(GENRE, FILLER))¶
```

Esses gráficos são chamados de gráficos de barras empilhadas ou mosaicos e são - juntamente com os gráficos a serem apresentados abaixo - maneiras geralmente eficientes de representar dados tabulados graficamente. No arquivo que contém o *script* com as funções e comandos deste capítulo, você irá encontrar os códigos para fazer um outro tipo de gráfico útil.

Figura 30 – Gráfico de barra / mosaico para `FILLER~GENRE`

3.2.1.2 Spineplots

Muitas vezes, a sua variável dependente é categórica/nominal e sua variável independente é intervalar. Suponhamos que “FILLER” seja a variável dependente de interesse e que ela é influenciada pela variável independente `LENGTH`. (Esse tipo de raciocínio não faz sentido aqui. Vamos fazê-lo apenas para fins de exemplificação). Você pode, então, utilizar a função `spineplot` com a seguinte fórmula:

Código em R

```
>spineplot(FILLER~LENGTH)¶
```

Figura 31 – *Spineplot* para FILLER~LENGTH

O eixo-y representa a variável dependente e seus três níveis. O eixo-x representa a variável intervalar independente, que é separada em intervalos de valores, assim como seria com a função `hist` (o que também quer dizer que você pode modificar os intervalos utilizando `breaks=...`; veja a Seção 3.1.1.5 acima).

3.2.1.3 Gráficos de linha

Além desses gráficos, você também pode fazer gráficos de linha que sumarizem as frequências de seus dados. Se você criar uma tabela com as frequências relativas, então a criação de um gráfico de linha pode ser feita da seguinte maneira:

Código em R

```
> fill.table<-prop.table(table(FILLER, SEX), 2); fill.table¶
SEX
FILLER    female      male
silence 0.3406375 0.3232932
uh       0.3207171 0.4678715
uhm      0.3386454 0.2088353
> plot(fil.table[,1], ylim=c(0, 0.5), xlab="Disfluency",
       ylab="Relative frequency", type="b")¶
> points(fil.table[,2], type="b")¶
```

Um código mais avançado para gerar o gráfico da Figura 33 pode ser encontrada no arquivo com os códigos e fórmulas deste capítulo. (Outra vez, você não pode compreender o código imediatamente, mas não levará por muito tempo para que você consiga fazer isso.)

Aviso Importante

Algumas vezes, recomendamos que você não represente frequências com gráficos de linha como o apresentado acima, pois as linhas ‘sugerem’ que existem frequências entre os valores da variável nominal, o que, obviamente, não é o caso. Novamente, você deve explorar a função `dotchart` para fazer isso.

Figura 32 – Gráfico de linha com os percentuais e interação entre Sexo e Partícula

Conselho

a função `plotmeans` (da biblioteca `gplots`) para gerar gráficos de linhas com médias e intervalos de confiança

3.2.2 Médias

Quando a sua variável dependente é intervalar, de razão ou ordinal e sua variável independente é nominal/categórica, o interesse geralmente não é na identificação das frequências de valores específicos da variável dependente, mas sim na exploração das medidas de tendência central para cada nível da variável independente. Um exemplo disso seria o seu interesse em saber se a duração das partículas de disfluência para homens e mulheres é diferente. Para isso, calculamos a duração média das partículas para homens e mulheres:

Código em R

```
> mean(LENGTH[SEX=="female"])
[1] 928.3984
> mean(LENGTH[SEX=="male"])
[1] 901.5803
```

Esse procedimento pode ser muito primitivo por três motivos:

- você deve definir os valores de `LENGTH` que você deseja incluir manualmente, o que requer muita digitação (especialmente quando a variável independente tem mais de dois níveis, ou, pior, quando você tem mais de uma variável independente);

- você deve saber todos os níveis relevantes das variáveis independentes - caso contrário, você não poderia usá-los para a seleção de subconjuntos;
- você somente obtém as médias para os níveis que você explicitamente definiu. Caso você tenha cometido um erro no processo de codificação (exemplo: digitar “hommen” ao invés de “homem”), os erros obviamente não serão incluídos. E essa maneira de encontrar as médias não irá apontar esses erros para você.

Portanto, usamos uma maneira mais sofisticada de encontrar as médias para os vários níveis da sua variável independente por meio da função `tapply`, a qual pede na maioria das vezes três argumentos. O primeiro argumento é o vetor em que você deseja aplicar a função `mean`. O segundo argumento é o vetor ou fator que contém o mesmo número de elementos que o primeiro e que especifica os grupos de valores do primeiro vetor/fator aos quais a função será aplicada. O último argumento é a função, no caso `mean`. Então temos:

Código em R

```
> tapply(LENGTH, SEX, mean)¶
female      male
928.3984 901.5803
```

É óbvio que o resultado é o mesmo que encontramos na função manual acima, mas você o obteve de uma forma melhor. Você pode utilizar qualquer outra função diferente da `mean`: `median`, `IQR`, `sd`, `var`, ..., até mesmo as funções que você mesmo fizer. Por exemplo, o que você obterá como resultado se utilizasse a função `length`? Os números de durações observadas para cada sexo.

3.2.2.1 *Boxplots*

Na Seção 3.1.4.7, vimos como fazer *boxplots*, para casos onde tínhamos uma ou mais variáveis dependentes (tais como cidade 1 e cidade 2). Vamos ver agora como fazer *boxplots* para os casos em que temos uma ou mais variáveis independentes e uma variável dependente. Da mesma forma como fizemos anteriormente, a maneira mais simples é utilizando uma fórmula com o sinal til significando “em função de”:

Código em R

```
> boxplot(LENGTH~GENRE, notch=TRUE, ylim=c(0, 1600))¶
```

(Para obter o mesmo gráfico sem ter que especificar argumento algum, basta que execute o comando `plot(LENGTH~GENRE)`¶: O programa R irá ‘inferir’ que você quer um *boxplot* uma vez que um vetor é numérico e o outro é nominal). Através de uma inspeção simples no gráfico, você nota que ambas medianas estão próximas a 900 ms e que provavelmente elas não se diferem significativamente uma da outra, uma vez que os *degraus* se sobrepõem. A dispersão (variabilidade) de ambos os gêneros parece ser bem parecida (as caixas e linhas são praticamente do mesmo tamanho) e não apresentam *outliers*.

Pergunta rápida: você pode inferir o que esta linha faz?

Código em R

```
> text(seq(levels(SEX)), tapply(LENGTH, SEX, mean), "+")¶
```

Ela adiciona sinais de adição no *boxplot* representando as médias de `LENGTH` para cada `GENRE`: `seq(levels(GENRE))` retorna 1:2, que é usado como as coordenadas x; o código `tapply` retorna as médias de `LENGTH` para cada `GENRE`, o o “+” é o que é gerado no gráfico.

Figura 33 – *Boxplot* para $LENGTH \sim GENRE$

Pausa para pensar

3.2.2.2 Gráficos de interação

Até agora, analisamos gráficos representando uma variável ou uma variável que depende de outra variável. Porém, há também casos em que você quer caracterizar a distribuição de uma variável de escala de razão/intervalar que depende de duas, digamos, variáveis nominais/categóricas. Para obter as médias das combinações dos níveis das variáveis independentes, você pode utilizar a função `tapply`. Você deve especificar ambas variáveis independentes na forma de uma lista. Os seguintes comandos mostram como obter as mesmas médias de duas maneiras distintas (de forma que você possa ver quais variáveis vão nas linhas e quais vão nas colunas):

Código em R

```
> tapply(LENGTH, list(SEX, FILLER), mean)¶
silence uh uhm
female 942.3333 940.5652 902.8588
male 891.6894 904.9785 909.2788
> tapply(LENGTH, list(FILLER, SEX), mean)¶
female male
silence 942.3333 891.6894
uh 940.5652 904.9785
uhm 902.8588 909.2788
```

Tais resultados são visualizados de maneira mais clara em tabelas, pois assim vemos as médias das interações bem como as médias para cada nível individualmente (assim como representado na Figura 34. Considere a Tabela 17 a fórmula em sua legenda exemplificando a sintaxe do R em questão.

Um sinal de mais entre as variáveis refere-se apenas à adição de efeitos principais de variáveis (ou seja, efeitos de variáveis isoladamente, por exemplo, quando você apenas verifica as duas médias para `SEX` na linha inferior de totais ou as três médias para `FILLER` na coluna mais à direita de totais). Um sinal de dois-pontos entre variáveis refere-se apenas à interação das variáveis (ou seja, aos efeitos de combinações de variáveis como quando você verifica as seis médias no corpo principal da tabela

Tabela 17 – Médias para $LENGTH \sim FILLER * SEX$

	SEX: FEMALE	SEX: MALE	Total
FILLER: uh	940,57	904,98	919,52
FILEER: uhm	902,86	909,28	905,30
FILLER: silencio	942,33	891,69	917,77
Total	928,40	901,58	915,04

onde SEX e FILLER são combinados). Finalmente, um asterisco entre variáveis denota tanto os efeitos principais como a interação (aqui, todas as 12 médias). Com duas variáveis A e B, $A*B$ é o mesmo que $A + B + A:B$.

Agora vamos para os resultados, os quais são compreendidos mais facilmente quando são representados graficamente. Você pode criar e modificar um gráfico de interação manualmente, mas para uma inspeção rápida nos dados, você pode utilizar a função `interaction.plot`. A função requer pelo menos três argumentos:

- `x.factor`: um vetor/fator que terão os níveis representados no eixo-x;
- `trace.factor`: um vetor/fator que será representado por linhas. Cada nível será uma linha distinta;
- `response`: um vetor com os dados numéricos que se deseja calcular a média. Esse vetor será representado no eixo-y.

Isso significa que, a depender de qual variável você deseja ver representadas pelas linhas do gráfico de interação, você tem duas opções de gráficos. Apesar de as médias serem obviamente idênticas, é sempre interessante analisar os dois gráficos e selecionar aquele que é mais fácil de ser interpretado. Na Figura 34, você encontra os dois gráficos para os valores acima e eu prefiro o segundo.

Código em R

```
> interaction.plot(SEX, FILLER, LENGTH); grid()¶
> interaction.plot(FILLER, SEX, LENGTH); grid()¶
```

Pela inspeção dos gráficos, vemos que o comportamento (a duração) da partícula *uhm* é diferente do comportamento das partículas *uh* e “silêncios”: a duração média de *uh* e “silêncio” é maior para as mulheres do que para os homens, mas a duração média de *uhm* para mulheres é menor do que para homens. Uma pergunta importante: porque não reportar apenas as médias calculadas com `tap11y` e os gráficos da Figura 34 no seu estudo?

Pausa para pensar

Primeiramente, como já dito anteriormente, você nunca deve reportar uma média sem a sua medida de dispersão correspondente. Assim, quando você deseja fornecer as médias, você deve incluir também, por exemplo, os respectivos desvios padrão, erros padrão e intervalos de confiança:

Código em R

```
> tapply(LENGTH, list(SEX, FILLER), sd)¶
silence uh uhm
female 361.9081 397.4948 378.8790
male 370.6995 397.1380 382.3137
```

Figura 34 – Gráfico de Interação para $LENGTH \sim FILLER:SEX$

Pausa para pensar

Como você obtém os erros padrão e os intervalos de confiança?

Código em R

```
> se<-tapply(LENGTH, list(SEX, FILLER), sd)/
sqrt(tapply(LENGTH, list(SEX, FILLER), length)); se
silence uh uhm
female 27.67581 31.32698 29.05869
male 29.21522 26.01738 37.48895
> t.value<-qt(0.025, df=999, lower.tail=FALSE); t.value
[1] 1.962341
> tapply(LENGTH, list(SEX, FILLER), mean)-(t.value*se)
silence uh uhm
female 888.0240 879.0910 845.8357
male 834.3592 853.9236 835.7127
> tapply(LENGTH, list(SEX, FILLER), mean)+(t.value*se)
silence uh uhm
female 996.6427 1002.0394 959.882
male 949.0197 956.0335 982.845
```

Uma inspeção rápida nos resultados acima mostra que os desvios padrão são grandes e que as médias mais/menos um erro padrão se sobrepõem (assim como ocorre para os intervalos de confiança), o que mostra que as diferenças não são significativas. Você pode visualizar isso através da função `boxplot`, que também aceita fórmulas com mais de uma variável independente (`boxplot(LENGTH~SEX*FILLER, notch=TRUE)`), com um asterisco para representar as interações).

Segundo: os gráficos da Figura 35 não podem ser utilizados da maneira que estão. O programa R escolheu o intervalo para o eixo-y de forma que ele seja o menor possível e incorpore, ao mesmo tempo, todos os dados necessários. No entanto, esse pequeno intervalo aumentou as diferenças apresentadas. Um gráfico mais adequado teria que incluir o valor $y = 0$ (como você pode ver nos comandos abaixo) ou ter um intervalo em que a amplitude da variável `LENGTH` seja toda representada (veja no segundo par das linhas abaixo):

Código em R

```
> interaction.plot(SEX, FILLER, LENGTH, ylim=c(0, 1000))
> interaction.plot(FILLER, SEX, LENGTH, ylim=c(0, 1000))
> interaction.plot(SEX, FILLER, LENGTH, ylim=range(LENGTH))
> interaction.plot(FILLER, SEX, LENGTH, ylim=range(LENGTH))
```

3.2.3 Coeficientes de correlação e regressão linear

Na última seção deste capítulo, vamos falar de casos em que as variáveis dependentes e independentes são numéricas. Para isso, vamos trabalhar com um outro conjunto de dados. Vamos primeiramente limpar toda a memória da nossa área de trabalho:

Código em R

```
> rm(list=ls(all=TRUE))
```

Vamos examinar se existe alguma relação entre o tempo de reação (medido em milésimos de segundo) de falantes de uma segunda língua em experimentos de decisão lexical e o tamanho das palavras utilizados como estímulo. Assim temos:

- uma variável dependente numérica e contínua: o tempo de reação em milésimos de segundo do participante MS_LEARNER;
- uma variável independente também numérica: o tamanho das palavras utilizadas como estímulo (medidas em letras) LEARNER.

A correlação existente entre duas variáveis é geralmente medida pelo que chamamos de coeficiente de correlação r . O valor do coeficiente de correlação r varia entre -1 e +1. A Tabela 18 explica o que esses valores significam: o sinal indica a direção da correlação e o valor absoluto indica o tamanho da correlação. Quando a correlação é igual a 0, não existe qualquer tipo de relação entre as duas variáveis em questão. É por isso que a hipótese nula em um teste de correlação assume que $r = 0$ – o teste bilateral, com hipótese alternativa $r \neq 0$.

Tabela 18 – Coeficientes de correlação e como interpretá-los

Coeficiente de correlação	Tamanho da correlação	Tipo de correlação
$0,7 < r \leq 1$	muito alta	correlação positiva: quanto mais uma variável aumenta , mais a outra aumenta ... quanto mais uma variável diminui , mais a outra diminui ...
$0,5 < r \leq 0,7$	alta	
$0,2 < r \leq 0,5$	intermediária	
$0 < r \leq 0,2$	baixa	
$r \approx 0$	não há correlação (H_0)	
$0r > \geq -0,2$	baixa	correlação negativa: quanto mais uma variável aumenta , mais a outra diminui ... quanto mais uma variável diminui , mais a outra aumenta ...
$-0,2 > r \geq -0,5$	intermediária	
$-0,5 > r \geq -0,7$	alta	
$-0,7 > r \geq -1$	muito alta	

Vamos importar os dados para o R e gerar os dados, usando linhas de códigos que já devem ser familiares para você agora:

Código em R

```
> ReactTime=read.delim(file.choose())
> str(ReactTime); attach(ReactTime)
'data.frame': 20 obs. of 3 variables:
 $ CASE : int 1 2 3 4 5 6 7 8 9 10 ...
 $ LENGTH : int 14 12 11 12 5 9 8 11 9 11 ...
 $ MS_LEARNER: int 233 213 221 206 123 176 195 207 172 ...
> plot(MS_LEARNER~LENGTH, xlim=c(0, 15), ylim=c(0, 300),
xlab="Word length in letters", ylab="Reaction time of
learners in ms"); grid()
```

Veja a Figura 35 antes de continuar a leitura. Que tipo de correlação – positiva ou negativa – existe entre as variáveis?

Pausa para pensar

Existe uma correlação positiva, pois quanto mais o tamanho da palavra aumenta, maior é o tempo de reação. Mas precisamos quantificar essa correlação (encontrar o tamanho dela). Para isso, calculamos o chamado coeficiente de correlação de Pearson r (também conhecido como coeficiente de correlação produto-momento).

Primeiramente, vamos fazer esse cálculo manualmente: para começar, devemos calcular a covariância das duas variáveis através da seguinte fórmula (3.2.3).

Figura 35 – Gráfico de dispersão para MS_LEARNER~LENGTH

$$(24) \quad \text{Covariância}_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

Como você pode ver, a covariância envolve o cálculo da diferença entre o valor e a média de cada variável. Por exemplo, quando o i -ésimo valor dos dois vetores x e y estão acima das respectivas médias, esse par contribui um valor alto e positivo para o cálculo da covariância. No R, você consegue calcular a covariância manualmente ou através da função `cov`, que exige como argumento os dois vetores de interesse:

Código em R

```
> covariance=sum((LENGTH-mean(LENGTH))*(MS_LEARNER-
mean(MS_LEARNER)))/(length(MS_LEARNER)-1)¶
> covariance=cov(LENGTH, MS_LEARNER); covariance¶
[1] 79.28947
```

O sinal da covariância já indica que as variáveis são correlacionadas positivamente ou negativamente; nesse caso, a correlação é positiva. No entanto, não podemos utilizar o valor da covariância para quantificar o tamanho da correlação, pois o valor da covariância depende da escala de medida das variáveis. Por exemplo, se você multiplicar os vetores por 10, a covariância irá ficar 100 vezes maior. Obviamente a correlação deve continuar a mesma:

Código em R

```
> cov(MS_LEARNER*10, LENGTH*10)¶
[1] 7928.947
```

Logo, para normalizar a covariância, devemos dividi-la pelo produto dos desvios padrão dos dois vetores. Com isso, obtemos o valor do coeficiente de correlação r . Esta é uma correlação positiva muito alta, pois r está próximo do máximo teórico de 1. No R, podemos fazer tudo isso com mais eficiência

com a função *cor*. Seus dois primeiros argumentos são os dois vetores em questão, e o terceiro especifica o tipo de correlação desejado:

Código em R

```
> covariance/(sd(LENGTH)*sd(MS_LEARNER))  
[1] 0.9337171  
> cor(MS_LEARNER, LENGTH, method="pearson")  
[1] 0.9337171
```

No entanto, a correlação pode ser investigada mais a fundo. Podemos utilizá-la para fazer previsões sobre a variável dependente com base na variável independente. Esse método é conhecido como regressão linear. De uma maneira bem geral, podemos entender a regressão linear como um método que tenta traçar uma linha reta que represente de maneira acurada a dispersão dos dados. O termo “acurado” aqui é definido como “a linha que minimiza a soma dos quadrados das distâncias entre cada escore (das duas variáveis) e a linha que descreve a distribuição desses dados”. Em outras palavras, a linha de regressão perpassa os dados de forma que os desvios seja minimizados o máximo possível. Essa linha é definida por uma equação de regressão com dois parâmetros: a que representa a interceptação da reta no eixo-y e b que representa o declive da reta. Sem discutir a fórmula propriamente dita, vamos ver como estimar os valores dos parâmetros a e b no programa R. Utilizando notação que você já conhece, você define um modelo linear usando a função *lm* e a partir dele você consegue encontrar os valores de a e b.

Código em R

```
> model=lm(MS_LEARNER ~ LENGTH); model  
Call:  
lm(formula = MS_LEARNER ~ LENGTH)  
Coefficients:  
(Intercept) LENGTH  
93.61 10.30
```

A interceptação da reta no eixo-y – em outras palavras, o valor de y quando x = 0 – é 93,61, e o declive da linha de regressão é 10,3, que significa dizer que para cada aumento de uma letra, estima-se que o tempo de reação irá aumentar 10,3 milésimos de segundos. Como utilizar a linha de regressão para fazer estimativas? No nosso exemplo, não temos uma palavra com 16 letras, mas a partir da linha de regressão, podemos estimar qual seria o tempo de reação se houvesse tal palavra:

$$\begin{array}{rclclcl} \text{predicted reaction time} & = & \text{intercept} & + & b \times \text{LENGTH} \\ 258,41 & \approx & 93,61 & + & 10,3 \times 16 \end{array}$$

Código em R

```
> 93.61+10.3*16  
[1] 258.41
```

(É óbvio que essa é uma previsão simplista, uma vez que não leva em consideração outros fatores que certamente influenciam o tempo de reação em uma tarefa de decisão lexical, mas no modelo de regressão atual é assim que ela seria feita). Uma maneira alternativa de fazer o mesmo cálculo é através da função *predict*, a qual exige como primeiro argumento o modelo linear e como segundo argumento uma estrutura de dados, por exemplo, uma lista chamada *newdata*, que contém uma coluna com os

valores da variável independente para a qual se deseja fazer previsões. Com a exceção de diferenças resultantes do fato de eu estar usando apenas dois decimais, você obterá o mesmo resultado:

Código em R

```
> predict(modelo, newdata=list(LENGTH=16))
258.4850
```

O uso de `expand.grid` é exagerado aqui para uma estrutura de dados com um único comprimento, mas eu estou o usando aqui porque ele antecipa nossos usos de `prever` e `expand.grid` abaixo, onde podemos realmente obter previsões para um grande número de valores de uma só vez (como a seguir, o output não é mostrado aqui):

Código em R

```
predict(model, newdata=expand.grid(LENGTH=1:16))
```

Se você apenas fornecer como argumento da função (`predict`) o modelo linear, você obtém como resultado os valores que o modelo prediz para cada comprimento de palavra nos seus dados (você obtém o mesmo resultado com a função `fitted`).

Código em R

```
> round(predict(model), 2)
1 2 3 4 5 6 7 8
237.88 217.27 206.96 217.27 145.14 186.35 176.05 206.96
9 10 11 12 13 14 15 16
186.35 206.96 196.66 165.75 248.18 227.57 248.18 186.35
17 18 19 20
196.66 155.44 176.05 206.96
```

O primeiro valor do vetor `LENGTH` é 14, então o primeiro valor do resultado acima é o tempo de reação que a linha de regressão prediz para palavras de 14 letras, e assim por diante. Já que agora temos os parâmetros necessários, podemos representar a linha de regressão graficamente. Para isso, utilizaremos a função `abline`, a qual exige como argumento o modelo linear criado, ou os valores do intercepto e da inclinação; cf. Figura 36:

Código em R

```
> plot(MS_LEARNER LENGTH, xlim=c(0, 15), ylim=c(0, 300),
xlab="Word length in letters", ylab="Reaction time of
learners in ms"); grid()
> abline(model) # abline(93.61, 10.3)
```

Está claro o porquê do coeficiente de correlação ser tão alto: a linha de regressão é uma síntese excelente dos dados, uma vez que cada um deles está bem próximo da linha. (No arquivo com as funções do capítulo, você irá ver uma maneira de fazer esse gráfico um pouco mais informativo). Podemos até medir a distância entre cada um dos dados e a linha de regressão.

Essa diferença - a distância vertical entre um valor observado no eixo y/reação de tempo e o valor de y na linha de regressão para o valor x correspondente - é conhecida como resíduo, e a função `residuals` requer como argumento apenas o modelo linear criado anteriormente:

Figura 36 – Gráfico de dispersão e linha de regressão para MS_LEARNER~LENGTH

Código em R

```
> round(residuals(model), 2)¶
1 2 3 4 5 6 7 8
-4.88 -4.27 14.04 -11.27 -22.14 -10.35 18.95 0.04
9 10 11 12 13 14 15 16
-14.35 -6.96 8.34 11.25 7.82 -14.57 7.82 1.65
17 18 19 20
-1.66 10.56 6.95 3.04
```

Você pode facilmente se certificar manualmente de que esses valores são de fato resíduos:

Código em R

```
> round(MS_LEARNER-(predict(model)+residuals(model)), 2)¶
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Dois pontos importantes merecem destaque. Primeiramente, equações e linhas de regressão e equação são mais adequadas ao intervalo dos valores observados. No nosso exemplo, a equação da regressão foi calculada com base nos tamanhos das palavras entre 5 e 15 letras. Isso quer dizer que essa equação seria menos confiável para palavras com 50 letras ou mais. Em segundo lugar, a interpretação das previsões permitidas pela linha de regressão deve ser feita com cautela, pois previsões teoricamente sem sentido são possíveis: no nosso exemplo, a linha de regressão prediz um tempo de reação de 0 milésimos de segundos para palavras com tamanho -9 letras. Tais considerações serão importante mais adiante.

O coeficiente de correlação r nos permite especificar o quanto da variação de uma variável pode ser explicada pela outra variável. O que isso significa? Na nossa planilha de dados, os escores das duas variáveis – MS_LEARNER e LEARNER – não são todos idênticos, ou seja, eles variam em torno das respectivas médias. Essa variação, como vimos, é chamada de dispersão e é medida pelo desvio padrão ou pela variância. Se você elevar o coeficiente de correlação r ao quadrado e multiplicar o resultado por 100,

você irá obter a quantidade de variação em uma variável que é explicada pela outra variável. No nosso exemplo, como $r = 0.933$, podemos dizer que 87,18% da variação dos tempos de reação podem ser explicadas pelo tamanho das palavras. Esse valor é conhecido como coeficiente de determinação r^2 .

Aliás, às vezes eu ouvi alunos ou colegas compararem dois valores- r de tal forma que eles dizem algo como: “Oh, aqui $r = 0,6$, bom, isso é o dobro desse conjunto de dados, onde $r = 0,3$ ”. Até mesmo numericamente falando, isso é pelo menos enganoso, se nada pior. Sim, 0,6 é duas vezes mais que 0,3, mas não se deve comparar os valores- r diretamente dessa forma - é preciso aplicar primeiramente a chamada transformação- Z de Fisher, a qual é exemplificada nas duas linhas a seguir:

Código em R

```
> r<-0.3; 0.5*log((1+r)/(1-r))
[1] 0.3095196
> r<-0.6; 0.5*log((1+r)/(1-r))
[1] 0.6931472
> 0.6931472/0.3095196
[1] 2.239429
```

Portanto, um valor- r de 0,6 é duas vezes mais alto do que um de 0,3, mas isso reflete uma correlação que é na verdade cerca de $2^{1/4}$ vezes mais forte. Que tal escrever uma função `fisher.z` que possa calcular o Z a partir de r para você ...

O coeficiente produto-momento r é provavelmente o coeficiente de correlação mais utilizado. No entanto, existem situações nas quais a sua utilização não é adequada. Primeiro, quando as variáveis de interesse são ordinais ou quando elas não estão distribuídas de maneira normal (veja Seção 4.4 a seguir), então é interessante utilizar um outro coeficiente de correlação (Kendall's tau τ , por exemplo). Esse coeficiente de correlação é baseado apenas na ordem dos valores das variáveis, de forma que é mais apropriado para dados ordinais. Segundo, quando há *outliers* na distribuição das variáveis, é também mais apropriado a utilização do Kendall's τ , uma vez que a correlação baseada na ordem dos dados é menos sensível a valores extremos. Veja, por exemplo, que na Figura 37) temos um gráfico de dispersão que contém um *outlier* bem distinto no canto direito superior do gráfico. Se esse *outlier* não puder ser excluído, então o coeficiente de correlação r será fortemente influenciado por ele, mas não o coeficiente de correlação τ . O coeficiente de Pearson r e o de Kendall τ para todos os dados com exceção do *outlier* são 0,11 and 0,1, respectivamente, e a linha de regressão com o declive menor mostra que claramente não há correlação entre as variáveis. No entanto, se incluirmos o *outlier*, o coeficiente de Pearson r de repente se torna 0,75 (e a inclinação da reta muda consideravelmente) ao passo que o coeficiente Kendall τ continua consideravelmente pequeno: 0,14.

Mas como calcular o coeficiente de Kendall τ ? O cálculo desse coeficiente é bastante complexo (especialmente com exemplos e laços maiores), e por essa razão vamos ver como fazê-lo apenas automaticamente no R. A função é a mesma que utilizamos para o cálculo do coeficiente de Pearson r – `cor` – mas o argumento `method=...` deve ser alterado. Para os dados do nosso exemplo, o coeficiente de Kendall τ é alto também, pouca coisa menor que o coeficiente r . (Note que as correlações são bidirecionais - a ordem com a qual digitamos os vetores não importa - mas a regressão linear não funciona assim porque você tem uma variável dependente e uma variável independente, e também porque o que é colocado antes e depois do sinal til “~” faz diferença.)

Código em R

```
> cor(LENGTH, MS_LEARNER, method="kendall")
[1] 0.8189904
```

Toda a discussão anterior assumiu que existe uma relação linear entre as duas variáveis ou uma relação que é melhor caracterizada por uma linha reta. Nem sempre esse é o caso. Uma situação em que nem o coeficiente r nem o coeficiente τ são apropriados é quando a relação entre duas variáveis

Figura 37 – O efeito de um *outlier* no coeficiente de correlação r

não é linear. Uma inspeção rápida nos dados pode revelar a não linearidade da correlação entre duas variáveis. A Figura 38 representa um exemplo bem conhecido de Anscombe (1973) (os dados podem ser encontrados no arquivo `<_inputfiles/03-2-3_anscombe.csv>`). Esses dados apresentam a peculiaridade de que

- as médias e as variâncias da variável x ;
- as médias e as variâncias da variável y ;
- as correlações e a linha de regressão de x e y ;

são todas idênticas, apesar de as distribuições dos dados serem bem diferentes.

No quadrante superior esquerdo da Figure 38, vemos um caso onde os coeficientes r e τ não são problemáticos. No quadrante superior direito, temos uma situação em que existe uma correlação curvilínea entre x e y – a utilização de uma regressão linear nesse caso não faria sentido.⁴ Nos dois quadrantes inferiores, vemos duas distribuições em que é notável o efeito de um *outlier* no coeficiente r e na linha de regressão. Como todas as estimativas desses dados são iguais (médias, variâncias e correlações), esse exemplo mostra perfeitamente a importância de uma inspeção visual dos dados, e é por isso que nos próximos capítulos essa exploração visual quase sempre ocorre antes da aplicação dos procedimentos estatísticos.

Agora você deve fazer os exercícios do Capítulo 3...

Conselho

Não deixe que a multiplicidade de funções gráficas e configurações de R e/ou do seu software de planilha levem você a produzir exageros visuais. O fato de você poder usar 6 fontes diferentes, 10 cores, e diversos símbolos bonitos não significa que você deve usá-los todos: a visualização deve ajudar você e/ou o leitor a compreender algo que de outra forma seria difícil de entender, o que também significa que você deve se certificar de que seus gráficos são bastante auto-suficientes, ou seja, que eles contêm todas as informações necessárias para que eles sejam compreendidos (por

⁴ Não iremos discutir regressão não-linear neste livro. Para saber mais sobre regressão não-linear veja Crawley (2007, Cap. 18 e 20).

Figura 38 – A sensibilidade das correlações lineares: os dados Anscombe

exemplo, nomes significativos para o gráfico e para os eixos, legendas etc.) - um gráfico pode precisar de explicação, mas se a explicação for três quartos de uma página, é provável que o seu gráfico não esteja ajudando (ver Keen (2010, Cap. 1)).

Recomendação para estudo/leitura

- a função `s.hist` (da biblioteca `ade4`) e `scatterplot` (da biblioteca `car`) para produzir gráficos mais refinados de dispersão com histogramas e *boxplots*.
- Good e Hardin (2012, Cap. 8), Crawley (2007, Cap 5, 27) e Braun e Murdoch (2008, Cap. 8: Seção 3.2) para exemplos de gráficos úteis. <<http://cran.r-project.org/src/contrib/Views/Graphics.html>>

Estatística Analítica

As perguntas mais importantes da vida são, na sua grande maioria, apenas questões de probabilidades.
Pierre-Simon Laplace (from <<http://www-rohan.sdsu.edu/%7Emalouf/>>)

Na minha descrição das fases de um estudo empírico no Capítulo 1, eu deixei de mencionar um passo essencial: como decidir qual teste de significância utilizar (Seção 1.3.4). No presente capítulo, vamos discutir esse passo com detalhes bem como discutir como realizar uma variedade de testes estatísticos que você pode querer utilizar com os seus dados. Mais especificamente, neste capítulo explicarei como as estatísticas descritivas do Capítulo 3 são usadas para o teste de hipóteses. Por exemplo, na Seção 3.1, nós aprendemos a calcular medidas de tendência central (por exemplo, a média) e medidas de dispersão (por exemplo, desvio padrão) para uma certa amostra de dados. No presente capítulo, vamos aprender a testar se essas médias e desvios padrão se diferem significativamente de médias e desvios padrão já conhecidos, ou de médias e desvios padrão de uma amostra diferente.

No entanto, antes de começarmos com testes reais: como você decide qual dos muitos testes disponíveis é o adequado para suas hipóteses e dados? Uma maneira de tentar diminuir a grande variedade de testes é se perguntar as seis perguntas que vou listar de (24) a (29), e as respostas a essas perguntas geralmente indicam apenas um ou dois testes que você pode aplicar aos seus dados. (Um pouco mais tarde, vou também fornecer uma ajuda visual para facilitar este processo.).

Ok, vamos começar. A primeira pergunta é mostrada em (24).

(24) Que tipo de estudo você está conduzindo?

Normalmente, há apenas duas respostas possíveis para essa pergunta: “geração de hipóteses” e “teste de hipóteses”. A primeira resposta significa que você está lidando com um conjunto de dados (tipicamente grande) com a intenção de detectar estrutura(s) e criar hipóteses para estudos futuros; sua abordagem de análise é, portanto, orientada pelos dados (do inglês data-driven), ou da base para o topo (bottom-up). Um exemplo disso será discutido na Seção 5.6. A outra resposta corresponde ao que veremos na maioria dos exemplos neste livro, e significa que a sua abordagem dos dados envolve hipóteses específicas que você quer testar e requer os tipos de testes apresentados neste capítulo e na maioria do próximo.

(25) Que tipos de variáveis estão envolvidas em suas hipóteses, e quantas são?

Existem essencialmente dois tipos de respostas. Uma se refere ao valor de informação das variáveis e discutimos isso em detalhes na Seção 1.3.2.2 acima. O outro permite quatro possibilidades de respostas diferentes. Primeiro, você pode ter apenas uma variável dependente, e, neste caso, você normalmente quer calcular um teste de aderência para testar se os resultados de seus dados correspondem a outros resultados (de um estudo anterior) ou correspondem a uma distribuição conhecida (tal como uma distribuição normal). Exemplos incluem:

- A proporção de negações com “no” (por exemplo, *He is no stranger*) é igual à proporção de negações com “not” (por exemplo, *He is not a stranger*) nos seus dados? (isto é, os dois tipos de negação são igualmente prováveis)?
- O julgamento de aceitabilidade média que você obtém para uma sentença corresponde ao de um estudo anterior?

Em segundo lugar, você pode ter uma variável dependente e uma variável independente ou você pode ter apenas dois conjuntos de medidas (ou seja, duas variáveis dependentes). Em ambos os casos, você normalmente quer calcular um teste monofatorial de independência para determinar se os valores de uma/da variável independente estão correlacionados com os da outra variável dependente. Por exemplo,

- A animacidade da referência do objeto direto (uma variável independente categórica) se correlaciona com a escolha de uma de duas ordens constituintes pós-verbais (uma variável categórica dependente)?
- O julgamento de aceitabilidade médio (uma média de uma variável dependente de proporção/intervalo) varia em função do fato de os sujeitos que fazem os julgamentos serem falantes nativos da língua em questão ou não (uma variável categórica independente)?

Em terceiro lugar, você pode ter uma variável dependente e duas ou mais variáveis independentes, e em tal caso você quer calcular uma análise multifatorial (como uma regressão múltipla) para determinar se as variáveis independentes e suas interações se correlacionam com, ou prevêm, a variável dependente. Por exemplo,

- A frequência de um tipo de negação (uma variável dependente categórica com os níveis “NO” versus “NOT”; ver acima) depende do modo de comunicação (uma variável binária independente com os níveis FALADO versus ESCRITO), do tipo de verbo que é negado (uma variável independente categórica com os níveis *COPULA*, *HAVE* ou *LEXICAL*), e/ou da interação entre essas variáveis independentes?
- O tempo de reação a uma palavra *w* em uma tarefa de decisão lexical (uma variável dependente de proporção/escala) depende da classe gramatical de *w* (uma variável independente categórica), da frequência de *w* em um corpus de referência (uma variável independente de proporção/intervalo), do fato de o participante da pesquisa ter visto uma palavra semanticamente relacionada a *w* no teste anterior ou não (uma variável independente binária), do fato de o sujeito ter visto uma palavra fonologicamente semelhante a *w* no teste anterior ou não (uma variável binária independente) e/ou das interações entre essas variáveis independentes?

Em quarto lugar, você tem duas ou mais variáveis dependentes, e em tal caso você pode querer realizar uma análise multivariada, a qual pode ser exploratória (como a análise hierárquica de *clusters*, a análise de componentes principais, a análise fatorial, a escala multidimensional, etc.) ou de teste de hipóteses por natureza (MANOVA). Por exemplo, se você extraiu de um conjunto de dados de corpus 10 palavras e as frequências de todas as palavras de conteúdo que ocorrem perto delas, você pode executar uma análise de *cluster* para verificar quais palavras se comportam mais (ou menos) de forma semelhante entre si, o que frequentemente está correlacionado com a similaridade semântica.

(26) Os dados em seu conjunto de dados estão relacionados de tal forma que você pode associá-los uns aos outros de forma significativa e sistemática?

Esta questão diz respeito ao fato de você ter ou não o que chamamos de amostras independentes ou dependentes (e nos remete à noção de independência discutida na Seção 1.3.4.1). Por exemplo, suas duas amostras - o número de erros cometidos por dez falantes não-nativos do sexo masculino e por dez falantes não-nativos do sexo femininos em um teste de gramática, por exemplo - são independentes uma da outra se você não for capaz de mostrar uma relação significativa e sistemática entre o valor de

cada sujeito masculino a o valor de um sujeito feminino. Você não seria capaz de fazer tal coisa se você selecionasse aleatoriamente dez homens e dez mulheres para fazer o mesmo teste.

Existem duas maneiras de as amostras serem dependentes. Uma delas é se você testar os participantes da pesquisa mais de uma vez, por exemplo, antes e depois de um tratamento experimental. Neste caso, você poderia ligar significativamente cada valor na amostra pré-tratamento a um valor na amostra pós-tratamento, ou seja, você poderia conectar os dois valores de cada participante da pesquisa. As amostras são dependentes porque, por exemplo, se o participante #1 for muito inteligente e bom na linguagem testada, então essas características tornarão seus resultados melhores do que a média em ambos os testes, especialmente em comparação a um participante que é menos inteligente e menos proficiente na língua em questão e que terá um desempenho pior em ambos os testes. Reconhecer que as amostras são dependentes desta forma tornará o teste pré e após tratamentos mais precisos.

A segunda maneira de amostras serem dependentes pode ser explicada usando o exemplo acima de dez homens e dez mulheres. Se os dez homens fossem os maridos das dez mulheres, então pode-se considerar as amostras como dependentes. Por quê? Porque os cônjuges são em média mais parecidos entre si do que pessoas selecionadas aleatoriamente: muitas vezes eles têm QIs similares, profissões semelhantes, passam mais tempo uns com os outros do que com pessoas selecionadas aleatoriamente, etc. Assim, deve-se associar cada marido com sua esposa, tornando essas duas amostras dependentes.

A independência dos dados é muitas vezes um critério muito importante: muitos testes pressupõem que os dados sejam independentes, e para muitos testes você deve escolher seu teste dependendo do tipo de amostras que você tem.

(27) Qual é a estatística da variável dependente nas hipóteses estatísticas?

Existem essencialmente cinco respostas diferentes para essa pergunta, as quais também já foram mencionadas na Seção 1.3.2.3 acima. Sua variável dependente pode envolver frequências/contagens, tendências centrais, dispersões, correlações ou distribuições.

(28) Qual é a aparência da distribuição dos dados ou da estatística de teste? Normal, alguma outra que pode, em última instância, ser descrita por uma função de probabilidade (ou algo que pode ser transformado para parecer uma função de probabilidade), ou alguma outra?

(29) Quais são os tamanhos das amostras coletadas? $n < 30$ or $n \geq 30$?

Essas questões referem-se à Seção 1.3.4, onde expliquei duas coisas: primeiro, se os seus dados/estatísticas do teste seguirem uma distribuição de probabilidade específica, você pode usar um teste paramétrico mais simples do ponto de vista computacional. Caso contrário, você deve usar um teste não-paramétrico. Em segundo lugar, se os tamanhos das amostras forem suficientemente grandes, mesmo os dados de uma distribuição decididamente não normal podem começar a parecer de uma distribuição normal e, portanto, permitirem que você aplique testes paramétricos. É mais seguro, no entanto, ser muito cauteloso e talvez ser conservador e executar ambos os tipos de testes.

Vamos agora usar um gráfico (<sflwr_navigator.png>) que visualiza esse processo, o qual você deve ter baixado como parte de todos os arquivos do site parceiro. Vamos exemplificar o uso desse gráfico usando o cenário de exemplo acima: você levanta a hipótese de que o julgamento de aceitabilidade médio (uma média de uma variável dependente ordinal) varia em função do fato de os participantes da pesquisa serem falantes nativos da língua em questão ou não (uma variável independente binária/categórica).

Você começa na caixa redonda vermelha com *abordagem*. Depois, como o cenário acima é de teste de hipóteses, você desce para *estatística*. Em seguida, como o cenário acima envolve médias, você desce para a caixa azul arredondada com *média*. Continuando, uma vez que a hipótese envolve tanto uma variável dependente como uma independente, você desce à direita, via *1 DV 1 IV* para a caixa transparente com (testes para) *independência/diferença*. Você chegou a essa caixa através da caixa azul com *média*, então prossiga para a próxima caixa azul que contém o *valor da informação*. Agora você toma duas decisões: primeiro, a variável dependente é de natureza ordinal. Em segundo lugar, as amostras são independentes. Assim, você leva a seta para baixo até o canto inferior esquerdo, o que leva a uma caixa azul com Teste-U. Assim, o teste típico para a pergunta acima seria o Teste-U (a ser discutido abaixo), e a função do R para esse teste já é fornecida lá também: `wilcox.test`.

Agora, o que a seta tracejada significa? Ela significa que você também faria um Teste-U se sua variável dependente fosse intervalar/de escala de proporção mas violasse outras suposições do teste-t. Ou seja,

setas tracejadas fornecem testes alternativos para o primeiro teste escolhido de onde se originaram.

Obviamente, esse gráfico é uma simplificação e não contém tudo o que se quer saber, mas eu acho que ele pode ajudar os iniciantes a fazer as primeiras escolhas para os testes, então eu recomendo que, à medida que você continua com a leitura deste livro, você sempre determine, para cada seção, qual teste deve usar e como identificá-lo com base no gráfico.

Antes de começarmos, deixe-me lembrá-lo mais uma vez que, ao lidar com os seus próprios dados, você deve codificar suas variáveis nominais/categóricas com sequências de caracteres significativas para que o R as reconheça como fatores ao ler os dados de um arquivo. Além disso, assumirei que você baixou os arquivos de dados do site parceiro.

Recomendação para estudo/leitura

- Recomendações de estudo [Good e Hardin \(2012, Cap. 6\)](#) sobre a escolha de testes estatísticos

4.1 Distribuições e frequências

Nesta seção, vamos aprender a verificar se as distribuições e frequências de uma amostra se diferem significativamente de uma distribuição já conhecida (Seção 4.1.1) ou de uma outra amostra (Seção 4.1.2). Em ambos os casos, iremos começar com variáveis numéricas (intervalares/de razão) e depois falaremos de outros tipos de variáveis.

4.1.1 Ajuste de distribuição

4.1.1.1 Uma variável dependente (de escala de proporção)

Nesta seção, vamos aprender a verificar se a distribuição de uma variável dependente intervalar ou de razão é significativamente diferente de uma distribuição já conhecida. Vamos nos restringir ao caso mais comum – verificar se a distribuição de uma variável é normal, porque, como vimos na Seção 1.3.4, muitos testes estatísticos requerem que a variável esteja distribuída de maneira normal, logo precisamos saber como fazer esse tipo de teste.

Para aprender como fazer isso, vamos utilizar um exemplo do campo de aquisição de língua materna. Mais especificamente, vamos analisar a aquisição de tempo e aspecto da língua russa. Simplificando bastante, existe uma tendência de que aspectos perfectivos sejam altamente correlacionados com o tempo passado, ao passo que o aspecto imperfectivo é fortemente correlacionado com tempos que não sejam o passado. Essas correlações podem ser quantificadas a partir dos valores V de Cramer (veja [Stoll e Gries \(2009\)](#), e a Seção 4.2.1 abaixo). Suponhamos que você esteja interessado em investigar como esses valores – o V de Cramer – mudam com o tempo para uma criança falante de russo. Vamos assumir também que você tem 117 gravações da fala de uma criança e calculou o V de Cramer para cada uma dessas gravações. Agora você quer saber se esses valores estão distribuídos de maneira normal. Esse cenário envolve:

- uma variável dependente numérica chamada TENSEASPECT, que consiste dos valores V de Cramer;
- nenhuma variável independente, pois não estamos testando se a distribuição da variável TENSEASPECT é influenciada por, ou está relacionada à algo mais.

Existem várias maneiras pelas quais você pode testar a normalidade dos seus dados. O teste que vamos utilizar aqui é conhecido como Teste de Shapiro-Wilk (lembre-se: verifique <sflwr_navigator.png> para entender como chegamos a esse teste!). O único pré-requisito exigido pelo teste é que os dados sejam numéricos. O procedimento é o seguinte:

Procedimentos

- Formulação das hipóteses
- Análise dos gráficos
- Cálculo da estatística W e da probabilidade de erro p

Como sempre, vamos começar pela formulação das hipóteses:

H_0 : Os dados estão distribuídos de maneira normal; $W = 1$.

H_1 : Os dados não estão distribuídos de maneira normal; $W \neq 1$.

Primeiramente, você deve importar os dados que se encontram na pasta <_inputfiles/04-1-1-1_tense-aspect.csv> para o R. Em seguida, vamos criar alguns gráficos para uma primeira inspeção da distribuição dos dados. O código abaixo produz o gráfico da esquerda, mas você pode produzir o da direita também utilizando o código do arquivo de códigos.

Código em R

```
> RussianTenseAsp=read.delim(file.choose())
> attach(RussianTenseAsp)
> hist(TENSE_ASPECT, xlim=c(0, 1), main=, xlab="Tense-Aspect
correlation", ylab="Frequency") # left panel
```

Figura 39 – Histograma dos valores V de Cramer refletindo as forças das correlações entre aspecto e tempo

Uma olhada rápida nos gráficos mostra que os dados parecem estar distribuídos de maneira normal. No entanto, é necessário um teste de verdade para conferir se realmente estão distribuídos de maneira

normal. O cálculo do teste Shapiro-Wilk é bem complicado para ser feito semi-manualmente, por isso não iremos discutir a sua fórmula aqui. No R, porém, o cálculo do teste é extremamente simples. Basta utilizarmos a função `shapiro.test`, a qual requer como argumento o vetor de interesse:

Código em R

```
> shapiro.test(TENSE_ASPECT)¶
Shapiro-Wilk normality test
data: TENSE_ASPECT
W = 0.9942, p-value = 0.9132
```

Mas o que esse resultado significa? Esse resultado simples nos ensina uma importante lição: Normalmente, quando executamos um teste estatístico, esperamos encontrar um resultado significativo, ou seja, um valor- p menor que 0,05, pois esse resultado nos permite aceitar a hipótese alternativa. Nesse caso, no entanto, podemos aceitar um resultado não-significativo porque é frequentemente mais fácil lidar com variáveis distribuídas de forma normal. A razão para isso é novamente a lógica por trás do paradigma de falsificação. Quando $p < 0,05$, rejeitamos a hipótese nula H_0 e aceitamos a hipótese alternativa H_1 . Mas, no caso sendo discutido, queremos que a hipótese nula H_0 seja verdadeira, pois é ela que sugere que a distribuição seja distribuída de maneira normal. O valor- p do nosso teste foi 0,9132, o que significa que não podemos rejeitar a hipótese nula H_0 , ou seja, devemos assumir que os dados estão distribuídos de maneira normal. A maneira mais adequada de reportar esse resultado em um artigo é: "De acordo com o teste Shapiro-Wilk, a distribuição dos valores V de Cramer medindo a correlação entre tense-aspect não é estatisticamente diferente de uma distribuição normal: $W = 0,9942$; $p = 0,9132$." (Em parênteses ou depois de um sinal de dois pontos você usualmente menciona todas as estatísticas que ajudaram você a decidir se aceitaria a hipótese alternativa ou não.)

Como uma alternativa ao teste de Shapiro-Wilk, você também pode usar um teste de Kolmogorov-Smirnov para teste de aderência. Esse teste requer a função `ks.test` e é mais flexível que o teste de Shapiro-Wilk, pois pode testar mais do que apenas a normalidade e também pode ser aplicado a vetores com mais de 5000 dados. Para testar o valor V de Cramer para normalidade, você deve fornecer os vetores como o primeiro argumento. Depois, você nomeia a distribuição com a qual você deseja compará-los (para normalidade, "pnorm"), e então, para definir os parâmetros da distribuição normal, você fornece a média e o desvio padrão dos valores de V de Cramer:

Código em R

```
> ks.test(TENSE_ASPECT, "pnorm", mean=mean(TENSE_ASPECT), sd=sd(TENSE_ASPECT))¶
One-sample Kolmogorov-Smirnov test data: TENSE_ASPECT D = 0.078, p-value = 0.4752
alternative hypothesis: two-sided
```

O resultado é o mesmo que acima: os dados não se diferem significativamente da normalidade. Você também recebe um aviso, pois `ks.test` pressupõe que não há dois valores iguais no *input*, mas aqui alguns valores (por exemplo, 0,27, 0,41 e outros) são atestados mais de uma vez. Abaixo você verá uma correção rápida para esse problema.

Recomendação para estudo/leitura

- também como alternativas para os testes acima, ver as funções `jarqueberaTest` e `dagoTest` ambos da `(library(fBasics))`
- a função `mshapiro.test` (`library(mvnormtest)`) para testar normalidade multivariada

- a função `qqnorm` e sua documentação (para o gráfico quantil-quantil)
- [Crawley \(2005, p. 100\)](#), [Crawley \(2007, p. 216\)](#), [Sheskin \(2011, Teste 7\)](#)

4.1.1.2 Uma variável dependente (categórica/nominal)

Nesta seção, vamos voltar ao exemplo apresentado no Capítulo 1 (Seção 1.3) sobre a alternância das construções da língua inglesa sobre colocação de partículas. Essa alternância está representada no exemplo (30).

Exemplo(31)

- a He picked up the book. (verbo – partícula – objeto direto)
- b He picked the book up. (verbo – objeto direto – partícula)

Como você já sabe, ambas as construções são perfeitamente aceitas por falantes nativos como sendo corretas. Além disso, raramente um falante nativo consegue explicar a preferência por uma ou pela outra. Assim, é plausível pensar que ambas construções são igualmente frequentes na língua. Vamos testar se isso é verdade. Essa situação envolve:

- uma variável dependente categórica/nominal CONSTRUCTION: verbo-partícula-objeto vs. CONSTRUCTION: verbo-objeto-partícula;
- nenhuma variável independente, uma vez que não estamos interessados em saber se alguma outra variável influencia o formato da distribuição da variável CONSTRUCTION.

Tais perguntas são geralmente investigadas com testes da família qui-quadrado, que é um dos testes mais importantes e mais utilizados em estatística. Uma vez que não existe nenhuma variável independente, esse teste verifica se a distribuição observada é parecida com uma distribuição conhecida/esperada (veja Seção 3.1.6.2). Esse tipo de teste é conhecido como qui-quadrado de aderência (*goodness-of-fit*) e envolve os seguintes passos:

Procedimentos

- Formulação da hipótese
- Cálculo da estatística descritiva e visualização dos dados
- Cálculo das frequências esperadas em H_0
- Verificação das pressuposições do teste:
 - todas as observações são independentes uma das outras
 - 80% das frequências esperadas são maiores ou iguais a 5¹
 - todas as frequências observadas são maiores que 1
- Cálculo da contribuição para o qui-quadrado para todas as frequências observadas
- Soma dessas contribuições para obter a estatística χ^2 , os graus de liberdade gl e da probabilidade de erro p

O primeiro passo é muito fácil. Como você já deve saber, a hipótese nula geralmente postula que os dados são distribuídos de maneira aleatória, ou seja, ambas construções ocorrem com frequência semelhante: 50% cada (da mesma forma que vários lançamentos de uma moeda resulta em uma distribuição aproximadamente balanceada). Assim:

H_0 : As frequências dos dois níveis da variável CONSTRUCTION são as mesmas - caso você encontre alguma diferença entre as duas, ela é resultado de uma variação aleatória; $n_{V \text{ Part DO}} = n_{V \text{ DO Part}}$

H_1 : As frequências dos dois níveis da variável CONSTRUCTION não são as mesmas; $n_{V \text{ Part DO}} \neq n_{V \text{ DO Part}}$

Note que essa é uma hipótese alternativa bilateral H_1 , uma vez que não há especificação da direção da diferença. O próximo passo é coletar alguns dados e contar o número de vezes que cada construção ocorre nos dados coletados. Para facilitar esse processo, vamos utilizar as frequências publicadas em [Peters \(2001\)](#). Ela conduziu um experimento em que os participantes tinham que descrever algumas fotos e contabilizou as frequências das duas construções. Veja Tabela 19.

Tabela 19 – Frequências observadas em [Peters \(2001\)](#)

Verbo – Partícula – Objeto Direto	Verbo – Objeto Direto – Partícula
247	150

Podemos já notar que há uma preferência para a construção em que a partícula ocorre logo após o verbo. Uma olhada rápida na Tabela 19 sugere que a hipótese nula H_0 parece não estar correta – pelo menos para esse conjunto de dados.

Uma observação muito importante aqui: os principiantes muitas vezes olham para algo como a Tabela 19 e dizem, “oh, ok, temos dados de intervalo/proporção: 247 e 150. Por que isso está errado?”

Pausa para pensar

Está errado porque a Tabela 19 não mostra os dados brutos - o que ela mostra para você já é um resumo numérico. Você não tem dados de intervalo/proporção - você tem um resumo de intervalo/proporção de dados categóricos, porque os números 247 e 150 resumem as frequências dos dois níveis da variável categórica CONSTRUCTION (que você provavelmente obteve ao aplicar a função `table` a um vetor/fator). Uma estratégia para não confundir isso é sempre contemplar conceitualmente como seria a tabela de dados brutos no formato “caso por variável” discutido na Seção 1.3.3. Neste caso, ela ficaria assim:

A partir deste formato, é bastante óbvio que a variável CONSTRUCTION é categórica. Portanto, não misture resumos de intervalo/proporção de dados categóricos com dados de intervalo/proporção.

Como primeiro passo, vamos ver uma representação gráfica dos dados. Uma primeira opção seria fazer um gráfico de pontos. Assim, primeiro você digita as duas frequências - primeiro os dados de frequência, depois os nomes dos dados de frequência (para o gráfico) - e então você cria um gráfico de pontos ou um gráfico de barras da seguinte maneira:

Tabela 20 – A versão caso por variável dos dados da Tabela 19

Caso	CONSTRUCTION
1	VPO
2	VPO
247	VPO
248	VOP
	VOP
397	VOP

Código em R

```
> VPCs=c(247, 150) # VPCs="verb-particle constructions"¶
> names(VPCs)=c("V-Part-DO", "V-DO-Part")¶
> dotchart(VPCs, xlim=c(0, 250))¶
> barplot(VPCs)¶
```

A pergunta agora é se essa diferença é estatisticamente significativa, ou se ela pode ter surgido por acaso. De acordo com o procedimento acima, você deve calcular as frequências que seriam esperadas se H_0 estivesse correto. Esse cálculo é fácil: uma vez que existem, nesse conjunto de dados, um total de 397 construções (247+150), a frequência esperada por H_0 é 397 dividido por 2:

Código em R

```
> VPCs.exp=rep(sum(VPCs)/length(VPCs), length(VPCs))¶
> VPCs.exp¶
[1] 198.5 198.5
```

Vamos checar se podemos fazer um teste qui-quadrado nesse caso. A frequência esperada para cada caso é maior do que 5 e podemos assumir que os dados de Peter são independentes, uma vez que estamos assumindo que cada construção foi produzida por um falante diferente. Nesse caso, podemos executar o teste qui-quadrado. O cálculo do teste é bem direto. Veja a fórmula (31).

$$(32) \quad \text{Qui-quadrado de Pearson} = \chi^2 = \sum_{i=1}^n \frac{(\text{observado} - \text{esperado})^2}{\text{esperado}}$$

A fórmula diz o seguinte: para cada frequência da sua amostra, você calcula a chamada “contribuição” para o Qui-quadrado. Para esse cálculo, você deve (i) calcular a diferença entre o valor observado e o valor esperado, (ii) elevar essa diferença ao quadrado e (iii) dividir esse resultado pela frequência esperada. A soma de todas as contribuições é o qui-quadrado que se obtém para o teste. No nosso exemplo, o qui-quadrado é aproximadamente 23,7.

$$(33) \quad \text{Qui-quadrado de Pearson} = \chi^2 = \frac{(247-198,5)^2}{198,5} + \frac{(150-198,5)^2}{198,5} \approx 23,7$$

Código em R

```
> sum(((VPCs-VPCs.exp)^2)/VPCs.exp)¶
[1] 23.70025
```

Esse valor, obviamente, aumenta na medida em que as diferenças entre os valores observados e esperados aumentam (pois o numerador se torna maior). Além disso, o qui-quadrado se torna 0 caso os valores observados sejam iguais aos valores esperados. Assim, podemos simplificar a hipótese estatística da seguinte maneira:

$$H_0: \chi^2 = 0.$$

$$H_1: \chi^2 > 0.$$

No entanto, o valor do qui-quadrado em si não diz se as diferenças encontradas são grandes o suficiente para serem estatisticamente significativas. O que fazer então com esse valor? Antigamente, quando computadores não eram utilizados em análises estatísticas, esse valor era utilizado como referência em uma tabela, para saber se a diferença era significativa ou não. Essas tabelas geralmente apresentam os três níveis de significância padrão nas colunas e diferentes valores para os graus de liberdade (gl) nas linhas. O número de graus de liberdade corresponde ao número de categorias menos 1. No nosso caso,

Tabela 21 – Valores de χ^2 críticos para $p_{bilateral} = 0,05; 0,01$ e $0,001$ para $1 \leq gl \leq 3$

	p = 0,05	p = 0,01	p = 0,001
gl = 1	3,841	6,635	10,828
gl = 2	5,991	9,21	13,816
gl = 3	7,815	11,345	16,266

$gl = 2 - 1 = 1$. A Tabela 21 é um exemplo de uma dessas tabelas de qui-quadrado para os três níveis de significância e $gl=1$ a 3.

No programa R, você pode encontrar esses valores através da função `qchisq`. Essa função requer três argumentos:

- `p`: os valores-p para os quais você deseja calcular os valores qui-quadrado críticos (com seus respectivos graus de liberdade);
- `df`: os graus de liberdade para os respectivos valores-p para os valores críticos de qui-quadrado desejados;
- `lower.tail=FALSE`: esse argumento serve para dizer ao R que você deseja saber apenas a área sob a curva da distribuição qui-quadrado que está à direita do valor maior que o valor chi-quadrado observado.

Código em R

```
> qchisq(c(0.05, 0.01, 0.001), 1, lower.tail=FALSE)
[1] 3.841459 6.634897 10.827566
```

Usuários mais avançados podem encontrar o código para gerar toda a Tabela 21 no arquivo de códigos. Uma vez que você tem essa tabela e seus respectivos valores, você pode testar a significância estatística do valor qui-quadrado observado. Para isso, basta verificar se o valor qui-quadrado observado por você é maior que o valor qui-quadrado da tabela (com os graus de liberdade correspondentes). Basta começar pelo menor valor qui-quadrado tabulado e compará-lo ao valor observado. Continue esse procedimento até que o valor observado seja maior que o valor tabulado. Aqui, você checa se o qui-quadrado observado é significativo no nível de significância de 5% (que no caso do nosso exemplo é: $23,7 > 3,841$). Então, você verifica se esse valor é também significativo no nível de significância de 1% (que, novamente, no caso do nosso exemplo é verdadeiro: $23,7 > 6,635$). Finalmente, você pode também verificar se esse valor é significativo no nível de significância de 0,1% (novamente, sim: $23,7 > 10,827$). Nesse caso, você pode rejeitar a hipótese nula H_0 . Normalmente, esse tipo de resultado é apresentado da seguinte maneira: “De acordo com o teste Qui-quadrado, a distribuição dos dois tipos de construção verbo-partícula é significativamente diferente da distribuição esperada ($\chi^2 = 23,7$; $gl = 1$; $p_{bilateral} < 0,001$): a construção em que a partícula ocorre logo após o verbo foi observada 247 vezes, apesar da frequência esperada ser de 199 ocorrências. Já a construção onde a partícula ocorre logo após o objeto direto ocorreu 150 vezes, sendo que a frequência esperada era de 199 ocorrências”.

Com um conjunto de dados mais complexo e maior, esse tipo de cálculo semi-manual é difícil e suscetível a erros, e por essa razão iremos simplificá-lo significativamente. Primeiramente, você pode calcular o valor- p diretamente a partir do valor qui-quadrado, utilizando a função `pchisq` com os mesmos três argumentos acima:

Código em R

```
> pchisq(23.7, 1, lower.tail=F)
[1] 1.125825e-06
```

Como você pode observar, o nível de significância que encontramos a partir da Tabela 21 é confirmado pela função acima: o valor- p é menor que 0,001, mais especificamente, 0,00000125825. No entanto, existe uma maneira ainda mais simples de executar esse teste. Para isso, utilizamos a função `chisq.test`, que, para nosso propósito agora, requer três argumentos:

- x : um vetor com as frequências observadas;
- p : um vetor com os percentuais esperados (note que esse argumento NÃO requer as frequências esperadas e sim o percentual);
- `correct=TRUE` ou `correct=FALSE`: quando o tamanho da sua amostra é pequeno ($15 \leq n \leq 60$), recomenda-se que se utilize uma correção chamada correção de Yates; `correct=TRUE` é a configuração padrão.²

No nosso caso fica fácil: já temos um vetor com as frequências observadas; a amostra é muito maior que 60 e o percentual para as frequências observadas, de acordo com H_0 é 0,5. Uma vez que a H_0 diz que as construções são igualmente frequentes, e uma vez que existem apenas duas construções, o vetor das probabilidades esperadas contém duas vezes $1/2 = 0,5$. Portanto:

Código em R

```
> chisq.test(VPCs, p=c(0.5, 0.5))  
Chi-squared test for given probabilities  
data: VPCs  
X-squared = 23.7003, df = 1, p-value = 1.126e-06
```

Você obtém o mesmo qui-quadrado que obteria com o cálculo manual, mas nesse caso você também obtém o respectivo valor- p imediatamente. O que você não obtém nesse caso são as frequências observadas, mas elas são fáceis de ser obtidas. A função `chisq.test`, na verdade, calcula mais do que ela retorna como resultado. Ela retorna uma estrutura de dados (chamada lista, para a qual você pode dar um nome) cujo conteúdo pode ser facilmente inspecionado (*output* não mostrado):

Código em R

```
> test=chisq.test(VPCs, p=c(0.5, 0.5))  
> str(test)
```

Assim, se você deseja obter as frequências observadas, você deve digitar um `$` e o nome do componente da lista desejada, e é claro que dará o resultado que você já conhece.

Código em R

```
> test$expected  
[1] 198.5 198.5
```

É importante mencionar que o método acima calcula o valor- p para testes bilaterais. No programa R, existem várias maneiras pelas quais você pode definir se você deseja testes unilaterais ou bilaterais. No entanto, isso não é possível com o teste Qui-quadrado. Se você precisa do valor crítico do teste Qui-quadrado para $p_{unilateral} = 0,05$ com $gl = 1$, então você precisa calcular o valor crítico do teste de Qui-quadrado para $p_{bilateral} = 0,1$ com $gl = 1$ (para isso, utilize `qchisq(0.1, 1, lower.tail=FALSE)`),

² Para ver outras opções `?chisq.test`, `formals(chisq.test)` ou `args(chisq.test)`.

uma vez que seu conhecimento prévio é recompensado de tal forma que um resultado menos extremo na direção prevista será suficiente (cf. Seção 1.3.4). Além disso, isso significa que quando você precisa do valor unicaudal para um valor Qui-quadrado, apenas considere metade do valor bicaudal do mesmo valor do qui-quadrado. Neste caso, se a sua H_1 tivesse sido direcional, este teria sido o seu valor- p . Mas, novamente: isso só funciona com $df=1$.

Código em R

```
> pchisq(23.7, 1, lower.tail=FALSE)/2
```

Conselho

Apontamos anteriormente que você nunca deve alterar as suas hipóteses *após* obter os resultados apenas para mostrar que seu estudo deu certo e suporta a ‘nova’ H_1 . A mesma lógica deve ser seguida aqui: você não deve nunca alterar uma hipótese bilateral para uma hipótese unilateral após encontrar os resultados (por exemplo, se obtiver $p_{bilateral} = 0,08$ – não-significativo – você não deve alterar para $p_{unilateral} = 0,04$ – significativo). A sua escolha por um teste unilateral deve ser motivada conceitualmente.

Outra advertência extremamente importante: nunca calcule um teste qui-quadrado como o acima em percentagens - sempre use frequências observadas ‘reais’!

Recomendação para estudo/leitura

- as funções `binom.test` ou `dbinom` para testes binominais
- a função `prop.test` (Seção 3.1.5.2) para testar frequências relativas/procentagens com relação aos desvios das frequências esperadas/ porcentagens.
- a função `dmultinom` para auxiliar em testes multinominais
- [Baayen \(2008, Seção 4.1.1\)](#), [Sheskin \(2011, Teste 8 e 9\)](#)

4.1.2 Testes para diferenças/independência

Na Seção 4.1.1, vimos os testes para *goodness-of-fit* (Testes de Qualidade de Ajuste ou Aderência) para distribuições e frequências. Vamos agora ver testes para diferenças/independência.

4.1.2.1 Uma variável dependente (ordinal/intervalar/de escala de proporção) e uma variável independente (nominal). Amostras independentes

Vamos ver um exemplo no qual duas amostras independentes são comparadas em relação às suas distribuições. Vamos testar se homens e mulheres são diferentes no que diz respeito ao uso de *hedges* no seu discurso (expressões do tipo *kind of* ou *sort of*). Vale ressaltar que estamos interessados na distribuição como um todo – não apenas médias e variâncias. Obviamente, poderíamos fazer isso se quiséssemos. No entanto, é possível que, enquanto as médias são muito semelhantes, as variâncias das duas distribuições são, na verdade, bem diferentes. Assim, um teste de diferença entre as médias poderia não revelar alguma outra diferença entre as distribuições.

Para o nosso exemplo, vamos supor que você gravou 60 conversas de dois minutos entre o seu assistente de pesquisa e uma outra pessoa. Para 30 das conversas, essa outra pessoa era um homem e para as outras 30, essa outra pessoa era uma mulher. Então você contabilizou o número de *hedges* que os

homens e as mulheres produziram. Você agora deseja testar se a distribuição de *hedges* para homens e mulheres são diferentes uma da outra. Essa pergunta envolve:

- uma variável independente nominal, SEX: MALE e SEX: FEMALE;
- uma variável dependente intervalar/escala de proporção: a quantidade de *hedges* produzidas: HEDGES.

A pergunta sobre se há alguma diferença, em termos de gênero, entre as distribuições das frequências de *hedges* será investigada com o teste Kolmogorov-Smirnov para duas amostras (novamente, cheque <sflwr_navigator.png>):

Procedimentos

- Formulação das hipóteses
- Tabulação das frequências observadas; inspeção gráfica
- Teste de pressuposições do teste: dados devem ser contínuos
- Computar a distribuição das frequências acumuladas para ambas amostras. Calcular a diferença máxima absoluta D de ambas distribuições e determinar a probabilidade p de erro

Primeiramente as hipóteses: a hipótese em formato de texto é bem direta, e sua versão estatística é baseada na estatística conhecida como D , a qual será explicada abaixo.

H_0 : A distribuição da variável HEDGES não varia a depender dos níveis da variável independente SEX; $D = 0$.

H_1 : A distribuição da variável HEDGES é diferente a depender dos níveis da variável independente SEX; $D > 0$.

Antes de executarmos o teste propriamente dito, vamos inspecionar os dados graficamente. Primeiramente, carregue os dados que podem ser encontrados em <_inputfiles/04-1-2-1_euf.csv>, verifique a estrutura dos dados e disponibilize os nomes das variáveis.

Código em R

```
>Hedges=read.delim(file.choose())  
>str(Hedges)  
>attach(Hedges)
```

Uma vez que você está interessado na distribuição geral dos dados, você pode criar um gráfico conhecido como `stripchart`. Nesse tipo de gráfico, as frequências das *hedges* são representadas separadamente para cada sexo, mas para evitar que frequências idênticas sejam representadas uma sobre a outra (de maneira que você não consegue vê-las separadamente), utilizaremos o argumento `method="jitter"`, que adiciona um valor pequeno a cada um dos dados de maneira que diminui a proporção de dados sobrepostos (tente também `method="stack"`). Assim, não deixe que o programa R decida sozinho qual será o intervalo do eixo-x, mas inclua o valor $x = 0$ no eixo x. Finalmente, com a função `rug` você adiciona pequenas barras no eixo x (`side=1`), que também sobre a influência do argumento `method="jitter"`. O resultado pode ser visto na Figura 40.

Código em R

```
> stripchart(HEDGES~SEX, method="jitter", xlim=c(0, 25),
  xlab="Number of hedges", ylab="Sex")
> rug(jitter(HEDGES), side=1)
```

Figura 40 – Gráfico do tipo *Stripchart* para HEDGES~SEX

A Figura 40 mostra que, obviamente, os dados estão distribuídos de maneira bem diferente: em média, os valores para as mulheres parecem estar um pouco acima dos valores para os homens. Eles parecem também estar mais homogêneos (pouca variabilidade). Os dados para os homens parecem se dividir em dois grupos. Uma maneira de investigar esse fato mais sistematicamente é através da inspeção de histogramas (veja Figura 41). (Note que os intervalos dos eixos foram definidos de maneira idêntica, de forma a facilitar a comparação dos dados.)

Código em R

```
> par(mfrow=c(1, 2))
> hist(HEDGES[SEX=="M"], xlim=c(0, 25), ylim=c(0, 10), ylab=
  "Frequency", main=)
> hist(HEDGES[SEX=="F"], xlim=c(0, 25), ylim=c(0, 10), ylab=
  "Frequency", main=)
> par(mfrow=c(1, 1))
```

A pressuposição de dados contínuos não é exatamente satisfeita uma vez que as frequências são discretas – não existe frequência 3,3 ou 3,4, etc. – mas HEDGES abrange uma variedade considerável de valores e we could in fact jitter the values to avoid ties. Para testar as diferenças distribucionais utilizando o teste Kolmogorov-Smirnov, o qual envolve a distribuição cumulativa empírica dos dados, precisamos colocar os dados em ordem. Para isso, você coloca os valores da variável SEX na ordem em que vai colocar os valores de HEDGES, e faz o mesmo para os valores de HEDGES:

Código em R

```
> SEX=SEX[order(HEDGES)]
> HEDGES=HEDGES[order(HEDGES)]
```

O próximo passo é um pouco mais complexo. Você deve calcular a maior diferença existente entre os valores das duas distribuições de *hedges*. Você pode utilizar três passos para executar isso. No primeiro passo, você deve criar uma tabela de frequência de maneira que o número de *hedges* esteja na horizontal

Figura 41 – Histogramas com a quantidade de hedges para homens e mulheres

(linha) e os sexos na vertical (coluna). Essa tabela vai servir como argumento para a função `prop.table`, que cria uma tabela com percentuais de ocorrência (`margin=2`; cf. Seção 3.2.1, resultado não mostrado):

Código em R

```
>dists=prop.table(table(HEDGES,SEX), margin=2); dists
```

A partir dessa tabela, podemos ver, por exemplo, que 4 representa 10% de todas as *hedges* produzidas pelos homens. No entanto, esse percentual ainda não é acumulado. O segundo passo é, então, converter esses percentuais em percentuais acumulados. Para isso, você pode utilizar a função `cumsum` para gerar os percentuais cumulativos para ambas as colunas. Podemos também calcular a diferença na mesma linha:

Código em R

```
> differences=cumsum(dists[,1])-cumsum(dists[,2])
```

A função acima significa que você subtrai de cada um dos percentuais acumulados da primeira coluna (os valores para as mulheres) o valor correspondente na segunda coluna (os valores para os homens). O terceiro passo é calcular a maior diferença absoluta, que é o teste estatístico *D*:

Código em R

```
>max(abs(differences))
[1] 0.4666667
```

Basta então utilizar a tabela do teste Kolmogorov-Smirnov para verificar a significância desse valor; para ser significativo, esse valor deve ser maior que o valor tabulado. Para os casos onde as duas amostras

são relativamente numerosas, a Tabela 22 mostra os valores D críticos (limites) para o teste Kolmogorov-Smirnov bilateral (obtido de Sheskin 2011: Tabela A23).

Tabela 22 – Valores D críticos (limite) para o teste Kolmogorov-Smirnov

	p = 0,05	p = 0,01
$n_1 = n_2 = 29$	0,3571535	0,428059
$n_1 = n_2 = 30$	0,3511505	0,4208642
$n_1 = n_2 = 31$	0,3454403	0,4140204

O valor D observado (0,4667) é não somente significativo ($D > 0,3511505$), mas muito significativo ($D > 0,4208642$). Você pode então rejeitar a hipótese H_0 e reportar seus resultados da seguinte maneira: “De acordo com o teste Kolmogorov-Smirnov para duas amostras, existe uma diferença estatisticamente significativa entre as distribuições de *hedges* para homens e mulheres: no geral, as mulheres parecem produzir mais *hedges* e de forma mais homogênea quando comparadas com os homens, os quais usam menos *hedges* e cujos dados parecem pertencer a dois grupos ($D = 0,4667$, $p_{bilateral} < 0,01$).”

A lógica por trás desse teste pode nem sempre ser clara, mas vale a pena explorá-lo. Para isso, vamos verificar os dados graficamente mais uma vez. As linhas de comando abaixo geram um gráfico com as duas funções de distribuições empíricas acumuladas (para homens – preto – e para as mulheres – cinza) bem como uma linha vertical onde está localizada a diferença absoluta ($D = 0,4667$). A Figura 42 abaixo mostra a que o teste de Kolmogorov-Smirnov é sensível: distribuições acumuladas distintas.

Código em R

```
> plot(ecdf(HEDGES[SEX=="M"]), do.points=TRUE, verticals=
TRUE, main="Hedges: men (black) vs. women (grey)",
xlab="Numbers of hedges")
> lines(ecdf(HEDGES[SEX=="F"]), do.points=TRUE, verticals=
TRUE, col="darkgrey")
> abline(v=9, lty=2)
```

Por exemplo, o fato de os valores para as mulheres serem mais alto e mais homogêneos é indicado especialmente na parte esquerda inferior do gráfico, onde as menores frequências estão localizadas e onde as frequências masculinas sobem enquanto que as femininas continuam constante. Mais de 40% dos valores para os homens ocorre em um intervalo onde praticamente não há frequência para as mulheres. Como resultado disso, a maior diferença, localizada na posição $x = 9$ se encontra em um ponto onde a curva para os homens já cresceu de maneira considerável, enquanto que a curva das mulheres apenas começou a subir. Esse gráfico explica também porque a H_0 postula que $D = 0$. Se as curvas forem exatamente iguais, quer dizer que não há diferença entre as distribuições, logo D se torna 0.

A explicação acima simplificou as coisas um pouco. Primeiro, você nem sempre tem testes bicaudais e tamanhos de amostra idênticos. Em segundo lugar, valores idênticos - os chamados laços - podem complicar a computação desse teste (e de outros). Felizmente, você realmente não precisa se preocupar com nada disso porque a função do R `ks.test` faz tudo para você em apenas uma linha. Você só precisa dos seguintes argumentos:³

- x e y : os vetores contendo as distribuições que deseja comparar;
- `alternative="two-sided"` para testes bilaterais (definição padrão) ou `alternative="greater"` ou `alternative="less"` para testes unilaterais depende de H_1 que você gostaria de testar: o argumento `alternative="..."` se refere ao vetor nomeado primeiro, de forma que `alternative="greater"` significa que a distribuição do primeiro vetor é maior que a do segundo vetor.

³ Infelizmente, a função `ks.test` não aceita uma fórmula como *input*.

Figura 42 – Curva das frequências acumuladas de hedger para homens (preto) e mulheres (cinza)

Quando você testa uma H_1 bilateral, como fizemos aqui, a linha de comando que precisa digitar é bem mais simples e se reduz ao comando abaixo, sendo que você obtém o mesmo valor- D e o mesmo valor- p : (Eu omiti o aviso do programa sobre laços. Para não ver esse aviso, e você pode usar `method="jitter"` para se livrar dele; ver o arquivo de códigos.)

Código em R

```
> ks.test(HEDGES[SEX=="M"], HEDGES[SEX=="F"])¶
Two-sample Kolmogorov-Smirnov test
data: HEDGES[SEX == "M"] and HEDGES[SEX == "F"]
D = 0.4667, p-value = 0.002908
alternative hypothesis: two-sided
```

Recomendação para estudo/leitura

- além da função (`plot(ecdf(...))`), você também pode criar gráficos como esse com a função `plot.stepfun`
- [Crawley \(2005, p. 100\)](#), [Crawley \(2007, p. 316\)](#), [Baayen \(2008, Seção 4.2.1\)](#), [Sheskin \(2011, Teste 13\)](#)

4.1.2.2 Uma variável dependente (nominal/categórica) e uma variável independente (nominal/categórica). Amostras independentes

Na Seção 4.1.2.3, discutimos como testar se a distribuição de uma variável categórica é significativamente diferente de uma outra distribuição conhecida. No entanto, uma situação mais comum é testar se a

distribuição de uma variável categórica depende da distribuição de uma outra variável categórica.

Nos exemplos anteriores, investigamos as frequências de duas construções que envolvem a estrutura verbo-partícula. Concluímos que a distribuição dessas construções não é compatível com a H_0 . No entanto, vimos também que existem outras variáveis que se correlacionam com a escolha da construção verbo-partícula. Uma dessas variáveis tem a ver com o fato de o objeto direto fornecer uma informação já dada (i.e. a informação não é nova e sim conhecida). Colocando de maneira mais específica, estudos mostram que objetos diretos que se referem a referentes dados no contexto *discursivo* tendem a ocorrer com mais frequência antes da partícula, ao passo que objetos que introduzem uma informação nova tendem a ocorrer com mais frequência depois da partícula. Vamos investigar essa hipótese (uma hipótese bilateral, no entanto). Essa questão envolve:

- uma variável dependente categórica (ou nominal) Construção: Verbo-Partícula-Objeto vs. Construção: Verbo-Objeto-Partícula;
- uma variável independente categórica (ou nominal): a natureza da informação fornecida pelo objeto direto: GIVENNESS: GIVEN vs. GIVENNESS: NEW;
- as amostras são independentes pois estamos assumindo que o fato de um objeto fornecer informação dada ou nova não se relaciona com a natureza da informação fornecida por um outro objeto.

Assim como antes, essa questão é investigada com o teste qui-quadrado: você vai testar se os níveis da sua variável independente influencia a frequência de ocorrência dos níveis da sua variável dependente. O procedimento geral para o teste qui-quadrado é muito semelhante ao que vimos anteriormente para o teste de aderência, no entanto o cálculo das frequências esperadas é um pouco (superficialmente) diferente.

Procedimentos

- Formular as hipóteses
- Tabular as frequências observadas e representá-las graficamente
- Calcular as frequências esperadas given H_0
- Checar as especificidades (pressuposições) do teste:
 - todas as observações são independentes uma das outras
 - 80% das frequências esperadas são maiores ou iguais a 5
 - todas as frequências observadas são maiores que 1
- Calcular a contribuição para o qui-quadrado para todas as frequências observadas
- Calcular o teste estatístico χ^2 , df , e p

As hipóteses em formato textual são simples, especialmente porque aplicamos o que aprendemos para o teste de aderência acima:

H_0 : As frequências dos dois níveis da variável CONSTRUCTION não variam em função dos níveis da variável independente GIVENNESS; $\chi^2 = 0$.

H_1 : As frequências dos dois níveis da variável CONSTRUCTION variam em função dos níveis da variável independente GIVENNESS; $\chi^2 > 0$.

Para discutir esse tipo de teste qui-quadrado (teste de independência), vamos voltar aos dados de [Peters \(2001\)](#). Na discussão anterior, não apresentamos todos os dados de Peters, omitimos a variável independente GIVENNESS. [Peters \(2001\)](#) investigou também algo semelhante ao que vamos ver agora: a relação entre GIVENNESS e CONSTRUCTION. No experimento onde as pessoas tinham que descrever uma série de fotos, ela manipulou a variável GIVENNESS e obteve as 397 construções que já conhecemos, tabuladas conforme Tabela 23. (A propósito, as células de tais tabelas 2 por 2 são frequentemente referidas com as letras de a até d, sendo “a” a célula superior esquerda (85), “b” a célula superior direita (65), etc.)

Tabela 23 – Frequências observadas em [Peters \(2001\)](#)

	GIVENNESS: GIVEN	GIVENNESS: NEW	Total (linhas)
CONSTRUCTION:V DO PART	85	65	150
CONSTRUCTION:V PART OD	100	147	247
Total (Colunas)	185	212	397

Antes de discutirmos sobre como fazer o teste de significância, vamos primeiro explorar os dados graficamente. Para isso, carregue os dados de <_inputfiles/04-1-2-2_vpcs.csv>, crie uma tabela de dois fatores, e obtenha uma primeira impressão visual da distribuição dos dados (Figura 43).

Código em R

```
> VPCs=read.delim(file.choose())
> str(VPCs); attach(VPCs)
> Peters.2001=table(CONSTRUCTION, GIVENNESS)
> plot(CONSTRUCTION~GIVENNESS)
```

Vimos que as áreas de cores diferentes possuem tamanhos obviamente diferentes entre linhas/colunas. Para verificar se essas diferenças são significativas, devemos calcular as frequências esperadas pela H_0 . Mas como calcular as frequências previstas pela H_0 ? Uma vez que essa é uma pergunta central, vamos discutir esse ponto com mais detalhes.

Suponhamos que Peter tivesse encontrado os totais da Tabela 24. Qual seria a distribuição das frequências para H_0 ? Na Seção 4.1.2.3 acima, dissemos que a H_0 geralmente postula frequências iguais para as categorias. Assim, podemos assumir que a representação das frequências conforme apresentada na Tabela 24 seria correta. Todos os totais das margens são 100 e cada variável tem dois níveis igualmente frequentes, então temos 50 em cada célula.

Tabela 24 – Totais das frequências observadas em [Peters \(2001\)](#) – dados fictícios

	GIVENNESS: GIVEN	GIVENNESS: NEW	Total (linhas)
CONSTRUCTION:V DO PART			100
CONSTRUCTION:V PART OD			100
Total (colunas)	100	100	200

Tabela 25 – Totais das frequências esperadas em [Peters \(2001\)](#) – dados fictícios

	GIVENNESS: GIVEN	GIVENNESS: NEW	Total (linhas)
CONSTRUCTION:V DO PART	50	50	100
CONSTRUCTION:V PART OD	50	50	100
Total (colunas)	100	100	200

As hipóteses estatísticas que vão além de apenas indicar se $\chi^2 = 0$ ou não seriam então:

$$H_0: n_{V \text{ DO Part} \& \text{ Ref DO}=\text{given}} = n_{V \text{ DO Part} \& \text{ Ref DO}\neq\text{given}} = n_{V \text{ Part DO} \& \text{ Ref DO}=\text{given}} \\ = n_{V \text{ Part DO} \& \text{ Ref DO}\neq\text{given}}$$

Figura 43 – Gráfico do tipo Mosaico para CONSTRUCTION~GIVENNESS

H_1 : o mesmo que H_0 , mas com pelo menos um sinal de diferente (“ $\neq$ ”) no lugar de um sinal de igual (“=”).

No entanto, nem sempre a vida é simples assim. Por exemplo, (a) assim como aconteceu em [Peters \(2001\)](#), nem todos os participantes responderam todas as perguntas, ou (b) dados naturais raramente são bem balanceados como os apresentados acima. Voltando aos dados originais de [Peters \(2001\)](#), percebemos que não faz sentido assumir que as frequências seriam iguais. Falando de uma outra forma: H_1 não pode ser como colocamos na Tabela 24, pois de acordo com a Tabela 23, as frequências para os diferentes níveis da variável GIVENNESS não são igualmente frequentes. Se GIVENNESS não tivesse nenhuma influência no tipo de CONSTRUCTION, então seria natural esperar que as frequências das duas construções para cada um dos níveis de GIVENNESS fossem iguais às frequências das duas construções na amostra como um todo. Isso quer dizer que (i) todos os totais marginais (linhas e colunas) devem ser os mesmos – constantes (já que refletem os números dos elementos investigados), and (ii) as proporções dos totais marginais determinam as frequências de cada célula. Assim, surge um grupo de hipóteses um pouco mais complexo:

$$\begin{aligned}
 H_0: & \quad n_{V \text{ DO Part} \& \text{ Ref DO}=\text{given}} : n_{V \text{ DO Part} \& \text{ Ref DO}\neq\text{given}} \propto \\
 & \quad n_{V \text{ Part DO} \& \text{ Ref DO}=\text{given}} : n_{V \text{ Part DO} \& \text{ Ref DO}\neq\text{given}} \propto \\
 & \quad n_{\text{Ref DO}=\text{given}} : n_{\text{Ref DO}\neq\text{given}} \quad \text{e} \quad n_{V \text{ DO Part} \& \text{ Ref DO}=\text{given}} : n_{V \text{ Part DO} \& \text{ Ref DO}=\text{given}} \propto \\
 & \quad n_{V \text{ DO Part} \& \text{ Ref DO}\neq\text{given}} : n_{V \text{ Part DO} \& \text{ Ref DO}\neq\text{given}} \propto \\
 & \quad n_{V \text{ DO Part}} : n_{V \text{ Part DO}}
 \end{aligned}$$

H_1 : o mesmo que H_0 , mas com pelo menos um sinal de diferente (“ $\neq$ ”) no lugar de um sinal de igual (“=”).

Em outras palavras, você não pode simplesmente dizer, “como existem $2 \times 2 = 4$ células, a frequência esperada será 397 dividido por 4, ou seja, aproximadamente 100 em cada célula”. Se você fizer assim, a primeira linha da tabela terá um total de aproximadamente 200 construções – mas isso não pode ser verdade pois só existem 150 casos da CONSTRUCTION: Verbo-Objeto-Partícula. Assim, você deve incluir em seu

cálculo essa informação de que só existem 150 casos da CONSTRUCTION: Verbo-Objeto-Partícula, no cálculo das frequências esperadas. A maneira mais fácil de fazer isso é utilizando proporções/percentuais: existem $\frac{150}{397}$ casos de CONSTRUCTION: Verbo-Objeto-Partícula (i.e. $0,3778 = 37,78\%$). Assim, existem $\frac{185}{397}$ casos de GIVENNESS: GIVEN (i.e., $0,466 = 46,6\%$). Se essas duas variáveis forem independentes uma da outra, então sua probabilidade conjunta é $0,3778 \times 0,466 = 0,1761$. Uma vez que existem um total de 397 casos, a frequência esperada para essa combinação é $397 \times 0,1761 = 69,91$. Essa lógica pode ser reduzida à fórmula (33).

$$(34) \quad n_{\text{frequência esperada em cada célula}} = \frac{\text{total da linha} \times \text{total da coluna}}{n}$$

Se utilizar a fórmula (33) para cada uma das células, vai obter os dados da Tabela 26.

Tabela 26 – Frequências esperadas em Peters (2001)

	GIVENNESS: GIVEN	GIVENNESS: NEW	Total (linhas)
CONSTRUCTION:V DO PART	69,9	80,1	150
CONSTRUCTION:V PART OD	115,1	131,9	247
Total (colunas)	185	212	397

Você pode ver imediatamente que esta tabela corresponde a H_0 acima: as razões/proporções dos valores em cada linha e coluna são exatamente aquelas dos totais das linhas e totais das colunas, respectivamente. Por exemplo, a razão de 69,9 a 80,1 a 150 é a mesma que a de 115,1 a 131,9 a 247, e como a de 185 a 212 a 397, e o mesmo é verdadeiro na outra dimensão. Assim, H_0 não significa "todas as frequências das células são idênticas- mas sim "as proporções das frequências das células são iguais (entre si e em relação aos respectivos totais marginais)".

Esse método utilizado para calcular frequências esperadas pode ser estendido para tabelas de frequências mais complexas (ver Gries (2009b, Seção 5.1)). Mas, como testar se essas frequências esperadas se diferenciam muito das frequências observadas? Felizmente, não precisamos de hipóteses tão complicadas, mas podemos usar as versões mais simples de $\chi^2 = 0$ e $\chi^2 > 0$ usadas acima, e o teste qui-quadrado para independência é idêntico ao teste de aderência qui-quadrado que você já conhece: para cada célula, você calcula uma contribuição para o qui-quadrado e as soma para obter a estatística de teste qui-quadrado.

Como antes, o teste qui-quadrado só pode ser usado quando suas suposições são atendidas. As frequências esperadas são suficientemente grandes e, por uma questão de simplificação, assumimos aqui que cada participante da pesquisa forneceu apenas uma única frase de forma que as observações sejam independentes umas das outras: por exemplo, o fato de que algum participante produziu uma determinada frase em uma ocasião não afeta então a formulação de frase de nenhum outro participante. Podemos, portanto, proceder como acima e calcular (a soma de) as contribuições para o qui-quadrado com base na mesma fórmula, aqui repetida como (34):

$$(35) \quad \text{Qui-quadrado de Pearson} = \chi^2 = \sum_{i=1}^n \frac{(\text{observado} - \text{esperado})^2}{\text{esperado}}$$

Os resultados estão na Tabela 27 e a soma de todas as contribuições para o qui-quadrado é 9,82. Contudo, precisamos agora do número de graus de liberdade. Para tabelas bidimensionais e quando as frequências esperadas são calculadas com base nas frequências observadas, como é o caso aqui, o número de graus de liberdade é calculado de acordo com a fórmula em (35).⁴

Tabela 27 – Contribuições para o Qui-quadrado nos dados de Peters (2001)

	GIVENNESS: GIVEN	GIVENNESS: NEW	Total (linhas)
CONSTRUCTION:V DO PART	3,26	2,85	
CONSTRUCTION:V PART OD	1,98	1,73	
Total (colunas)			9,82

⁴ No nosso exemplo, as frequências esperadas foram calculadas a partir das frequências observadas nos totais marginais. Se você não calcula as frequências esperadas a partir dos dados observados, assim a partir de alguma outra distribuição, o cálculo dos graus de liberdade g_l muda para $g_l = (\text{número de fileiras} \cdot \text{número de colunas}) - 1$.

$$(36) \quad gl = (\text{número de linhas} - 1) \cdot (\text{número de colunas} - 1) = (2-1) \cdot (2-1) = 1$$

Com os valores do qui-quadrado e do grau de liberdade, podemos olhar o resultado na tabela de valores críticos para o teste qui-quadrado (por exemplo, a Tabela 28 abaixo, que é igual à Tabela 21). Se o valor obtido for maior que o valor tabulado para $p = 0,05$ com $gl = 1$, então podemos rejeitar a H_0 . No caso em questão, o valor qui-quadrado obtido é não apenas maior que o valor crítico para $p = 0,05$ e $gl = 1$ mas também para $p = 0,01$ and $gl = 1$. Porém, uma vez que o valor obtido não é maior que 10,827, o valor- p real está entre 0,01 e 0,001: o resultado é muito, porém não altamente, significativo.

Tabela 28 – Valores de χ^2 críticos para $p_{bilateral} = 0,05; 0,01$ e $0,001$ para $1 \leq gl \leq 3$

	$p = 0,05$	$p = 0,01$	$p = 0,001$
$gl = 1$	3,841	6,635	10,828
$gl = 2$	5,991	9,21	13,816
$gl = 3$	7,815	11,345	16,266

Felizmente, tudo isso é feito de maneira mais fácil no programa R. Você pode calcular o valor- p como fizemos antes,

Código em R

```
> pchisq(9.82, 1, lower.tail=F)
[1] 0.001726243
```

ou, se preferir, pode utilizar a função `chisq.test` e fazer tudo de uma vez só. Os argumentos mais importantes da função para os nossos objetivos são:

- x : uma tabela com duas dimensões contendo os valores para os quais se deseja calcular o qui-quadrado;
- `correct=TRUE` ou `correct=FALSE`; correção de continuidade.⁵

Código em R

```
> test.Peters=chisq.test(Peters.2001, correct=FALSE)
> test.Peters
Pearson's Chi-squared test
data: Peters.2001
X-squared = 9.8191, df = 1, p-value = 0.001727
```

É assim que você obtém as frequências esperadas ou o valor qui-quadrado:

Código em R

```
> test.Peters$expected
GIVENNESS
CONSTRUCTION given new
V_DO_Part 69.89924 80.10076
V_Part_DO 115.10076 131.89924
> test.Peters$statistic
X-squared
9.819132
```

⁵ Para mais opções, digite `?chisq.test`. Veja também o que acontece se digitar `summary(Peters.2001)`.

Agora você sabe que GIVENNESS está correlacionado com CONSTRUCTION, mas você não sabe ainda quão forte esse efeito é e nem quais combinações de nível variável são responsáveis por esse resultado. Quanto ao tamanho do efeito, mesmo que você possa estar inclinado a usar o tamanho do valor do qui-quadrado ou o valor- p para quantificar o efeito, você não deve fazer isso, porque o valor do qui-quadrado depende do tamanho da amostra, como podemos ver facilmente:

Código em R

```
> chisq.test(Peters.2001*10, correct=FALSE)¶
Pearson's Chi-squared test
data: Peters.2001 * 10
X-squared = 98.1913, df = 1, p-value < 2.2e-16
```

Isso pode ser problemático para efeitos de cálculo do tamanho do efeito, uma vez que uma amostra maior não quer dizer que a relação entre as variáveis muda. Isso pode ser notado ao inspecionar os percentuais: eles continuam os mesmos. Por esse motivo, para calcular o tamanho do efeito, geralmente utilizamos o coeficiente de correlação (conhecido como ϕ no caso de tabelas do tipo $k \times m \times 2$ ou V de Cramer para tabelas do tipo $k \times m$ onde k ou $m > 2$). Esse coeficiente de correlação é um valor que se encontra entre 0 e 1 (0 = não existe correlação; 1 = existe uma correlação perfeita) e não varia com a variação no tamanho da amostra. Esse coeficiente de correlação é calculado de acordo com a fórmula (36):

$$\Phi/V \text{ de Cramer/Index I de Cramer} = \sqrt{\frac{\chi^2}{n(\min[n_{\text{linhas}}, n_{\text{colunas}}] - 1)}} \quad (4.1)$$

Esse cálculo no R é:

Código em R

```
> sqrt(testePeters$statistic/sum(Peters.2001)*(min(dim(Peters.2001))-1))¶
X-squared
0.1572683
```

Dado o intervalo de valores teórico, esse é um valor relativamente pequeno.⁶ Assim, a correlação provavelmente não é por um acaso, mas também não é muito forte.

Uma outra medida de tamanho do efeito, que, porém, só pode ser utilizada com tabelas 2×2 , é a chamada razão de chances. Uma razão de chances diz como a probabilidade de uma variável muda em resposta a uma mudança de nível de outra variável. As probabilidades de um evento E correspondem à fração em (37).

$$(37) \quad odds = \frac{p_E}{1-p_E} = \text{para obter probabilidades de } odds = \frac{odds}{1+odds}$$

A razão de chances para uma tabela do tipo 2×2 (tal como a Tabela 23) é a razão entre duas chances (ou 1 dividido por essa chance, dependendo se está verificando a ocorrência de um evento (E) ou sua não ocorrência ($\neg E$), como em (38):

$$(38) \quad \text{Razão de chances para a Tabela 23} = \frac{85+65}{100+147} = 1,9223$$

Em outras palavras, as razões de chance para a CONSTRUCTION: V DO PART são $(\frac{85}{185}) / (1-\frac{85}{185}) = \frac{85}{100} = 0,85$ quando o referente do objeto direto é dado e $\frac{65}{147} = 0,4422$ quando o referente é novo. Isso significa que CONSTRUCTION: V DO PART é $\frac{0,85}{0,4422} \approx 1,9223$ vezes mais provável de acontecer quando a informação

⁶ O intervalo teórico entre 0 e 1 só é possível em algumas situações, mas mesmo assim é uma boa heurística para interpretar esse valor.

do objeto direto é dada do que quando não é. A partir disso, concluímos que a razão de chances de ausência de interação é ≈ 1 .⁷

A Tabela 27 mostra quais níveis de variáveis contribuíram mais para a associação significativa: quanto maior a contribuição da célula para o qui-quadrado, mais essa célula contribui para o qui-quadrado geral; no nosso exemplo, esses valores são bem pequenos – nenhum deles excede o valor- $p = 0,05$ e $g1 = 1$, ou seja, 3,841. No R, você obtém as contribuições para o Qui-quadrado da seguinte maneira:

Código em R

```
> test.Peters$residuals^2
GIVENNESS
CONSTRUCTION given new
V_DO_Part 3.262307 2.846825
V_Part_DO 1.981158 1.728841
```

Ou seja, você deve calcular os resíduos de Pearson. Os resíduos de Pearson, que podem ser calculados da maneira exposta abaixo; revelam a direção do efeito para cada célula: valores negativos e valores positivos significam que os valores observados que são menores ou maiores (respectivamente) do que os valores esperados:

Código em R

```
> test.Peterst$residuals
GIVENNESS CONSTRUCTION given new V_DO_Part 1.806186
-1.687254 V_Part_DO -1.407536 1.314854
```

Assim, dadas as pequenas contribuições para o qui-quadrado, se você quisesse tirar alguma conclusão a partir desses números, a única coisa que poderia dizer é que a combinação que contribuiu mais para o qui-quadrado geral foi a combinação CONSTRUCTION: V DO PART e GIVENNESS:GIVEN, a qual é mais observada do que o esperado, mas os efeitos de cada célula são mesmo muito pequenos.

Uma maneira interessante e reveladora de representar essa interação graficamente pode ser obtida a partir da função `assocplot`, que tem como principal argumento a tabela de contingência que se investiga: nesse gráfico (Figura 44), “a área de cada uma dos retângulos é proporcional ao tamanho da diferença entre o valor obtido e o valor esperado”. Os retângulos pretos acima da linha pontilhada representam as frequências observadas que são maiores que as frequências esperadas; os retângulos cinzas abaixo das linhas pontilhadas representam as frequências observadas que são menores que as frequências esperadas; as alturas dos retângulos são proporcionais aos resíduos de Pearson que calculamos acima e o comprimento deles é proporcional à raiz quadrada das frequências esperadas. Note que eu não gero apenas a tabela, mas a tabela transposta – é isso que o `t()` faz. Isto é para que a organização linha/coluna do gráfico corresponda à da tabela original:

Código em R

```
> assocplot(Peters.2001, col=c("black", "darkgrey"))
```

Uma outra maneira interessante de representar graficamente os dados é através de uma mistura entre gráfico e tabela. A Figura 44 tem a estrutura da Tabela 23, mas (i) o tamanho dos números reflete

⁷ Frequentemente, você pode encontrar o logaritmo da razão de chances (Veja especialmente a seção 5.3). Quando as duas variáveis não são correlacionadas, esse logaritmo da razão de chances é $\log 1 = 0$, e correlações positivas/negativas resultam em logaritmos da razão de chances positivos/negativos, o que é geralmente um pouco mais fácil de interpretar. Por exemplo, se você tem duas razões de chances tais como razão de chance 1 = 0,5 e a razão de chance 2 = 1,5, então você não pode imediatamente ver qual efeito é maior. Os logs das razões de chances – *log razão de chance 1* = -0,693 e *log razão de chance 2* = 0,405 – lhe dizem imediatamente que o primeiro é maior porque seu valor absoluto é maior

Figura 44 – Gráfico de Associação para CONSTRUCTION~GIVENNESS

o tamanho dos resíduos (números maiores representam uma diferença maior entre o valor observado e o valor esperado) e (ii) as cores indicam como é esse desvio: cinza escuro indica desvios positivos e cinza claro indica desvio negativo. (Por falta de um termo melhor, eu me refiro a isso como um gráfico de crosstabulação.)

Figura 45 – Gráfico de Contingência Construção~Natureza

É assim que você resumiria todos os resultados: “Objetos novos são fortemente preferidos na construção Verbo-Partícula-Objeto Direto e ocorrem com menos frequência na construção Verbo-Objeto Direto-Partícula”. O tipo oposto de preferência por uma das construções é encontrado para objetos dados. De acordo com um teste qui-quadrado para a independência, essa correlação é muito significativa. ($\chi^2 = 9,82; df = 1; p_{two-tailed} < 0,002$), mas o efeito não é particularmente forte ($\phi = 0,157$, razão de chances = 1,9223).

Vale a pena ressaltar novamente que o procedimento acima fornece o valor- p para um teste bilateral. No caso de tabelas 2×2 , você pode fazer um teste unilateral, como foi discutido na Seção 4.1.2.3, mas, você não pode fazer um teste unilateral para tabelas com graus de liberdade $gl > 1$.

Recomendação para estudo/leitura

- as funções `dotchart` como `mosaic(library(vcd))` e a função `table.cont` (`library(ade4)`) para outros tipos de gráficos
- a função `assocstats` (`library(vcd)`) para ver maneiras diferentes de calcular os testes qui-quadrado e o tamanho do efeito ao mesmo tempo
- a função `CrossTable` (`library(gmodels)`) para fazer tabelas mais completas
- o argumento `simulate.p.value=TRUE` da função `chisq.test` e a função `fisher.test`, que você pode utilizar no caso de frequências esperadas muito pequenas
- o procedimento Marascuilo que verifica quais pares de frequências são diferentes umas das outras (cf. [Gries \(2014, p. 172\)](#), que também discute como testar uma sub-tabela dentro de uma tabela maior)
- [Crawley \(2005, p. 85\)](#), [Crawley \(2007, p. 301\)](#), [Sheskin \(2011, Teste 16\)](#)

Conselho

Mais uma vez: nunca calcule um teste chi-quadrado em porcentagens - sempre em frequências observadas "reais"! (Confie em mim, há uma razão para eu repetir isso ...)

Existe mais uma aplicação do teste qui-quadrado [Zar \(1999, Seção 23.4\)](#) e [Sheskin \(2011, p. 691\)](#)) Algumas vezes, você terá várias tabelas 2×2 sobre um mesmo fenômeno (talvez por que tenha encontrado uma outra fonte que discute os mesmos tipos de dados). Nesse caso, você pode querer testar se as duas tabelas são semelhantes o suficiente a ponto de você poder juntá-las em uma tabela única. Veja as hipóteses desse tipo de teste:

H_0 : A tendência dos dados é a mesma: heterogeneidade do $\chi^2 = 0$.

H_1 : A tendência dos dados não é a mesma: heterogeneidade $\chi^2 \neq 0$.

Para explorar essa questão, vamos comparar os dados de [Peters \(2001\)](#) com os dados de [Gries \(2003a\)](#). Você pode inserir os últimos no R diretamente usando a função `matrix`, a qual precisa do vetor de frequências observadas (em colunas), o número de colunas e os nomes das dimensões (primeiras linhas, depois colunas):

Código em R

```
> Gries.2003=matrix(c(143, 53, 66, 141), ncol=2, dimnames=list(CONSTRUCTION=
c("V_DO_Part", "V_Part_DO"),GIVENNESS=c("given", "new")))#
> Gries.2003#
given new
V_DO_Part 143 66
V_Part_DO 53 141
```

Por um lado, esses dados são bem diferentes dos dados de [Peters \(2001\)](#), pois quando GIVENNESS é GIVEN, então a CONSTRUCTION: V_DO_PART é quase três vezes mais frequente que CONSTRUCTION: V_PART_DO (e não menos frequente, como nos dados de Peters). Por outro lado, os dados são bem semelhantes, objetos diretos que trazem informação dada ocorrem com mais frequência em CONSTRUCTION: V_DO_PART. Uma comparação direta dos gráficos de associação (não mostrados aqui, mas você pode usar o código a seguir para gerá-los) faz com que os dados fiquem muito parecidos - quão mais semelhantes dois gráficos de associação poderiam ser?

Código em R

```
> par(mfrow=c(1,2))  
> assocplot(Peters.2001)  
> assocplot(Gries.2003)  
> par(mfrow=c(1,1))
```

Porém, mesmo os gráficos sendo muito semelhantes, você nunca deve comparar dados com base apenas nos gráficos – apenas a tendência geral. Para testar a semelhança entre eles, fazemos o teste de heterogeneidade do qui-quadrado. O valor da heterogeneidade do Qui-quadrado é calculado como sendo a diferença entre a soma dos qui-quadrados de cada uma das tabelas e o qui-quadrado que você obtém das duas tabelas juntas. Os graus de liberdade é calculado de maneira semelhante. No programa R, esses cálculos são feitos da seguinte maneira. Primeiramente, você calcula o qui-quadrado para os dados de [Gries \(2003a\)](#):

Código em R

```
> test.Gries=chisq.test(Gries.2003, correct=FALSE)  
> test.Gries  
Pearson's Chi-squared test  
data: Gries.2003  
X-squared = 68.0364, df = 1, p-value < 2.2e-16
```

Depois, você calcula a soma dos valores qui-quadrados de cada uma das tabelas originais:

Código em R

```
> test.Peters$statistic+test.Gries$statistic  
X-squared  
[1] 77.85552
```

Em seguida, você calcula o qui-quadrado das tabelas combinadas ...:

Código em R

```
> chisq.test(Peters.2001+Gries,correct=FALSE)$statistic  
X-squared  
[1] 65.87908
```

... e para obter a heterogeneidade do qui-quadrado e seus graus de liberdade (para obter *gl* utilize \$parameter):

Código em R

```
> het.chisq=77.85552-65.87908 # 11.97644¶
> het.df=1+1-1 # 1¶
```

Como você obtém o valor- p para esses resultados?

Pausa para pensar

Código em R

```
> pchisq(het.chisq, het.df, lower.tail=FALSE)¶
[1] 0.0005387754
```

Os dados dos dois estudos exibem as mesmas tendências gerais (objetos dados aumentam a probabilidade da construção CONSTRUCTION: V_DO_PART, mas eles ainda se diferem significativamente um do outro ($\chi^2_{heterogeneity} = 11,98$; $df = 1$; $p_{two-tailed} < 0,001$). O que é responsável por essa diferença? Os tamanhos de efeitos diferentes: a razão de chances para os dados de Peters foi 1,92, mas nos dados de Gries é quase exatamente três vezes maior, e é isso que você escreveria na seção de resultados do seu artigo. Voltaremos a esse exemplo no Capítulo 5.

Código em R

```
> (143/66)/(53/141)¶
[1] 5.764151
```

4.1.2.3 Uma variável dependente (nominal/categórica). Amostras dependentes

Um dos pré-requisitos básicos do teste de qui-quadrado é que as amostras comparadas sejam independentes uma da outra. No entanto, em algumas situações essa independência não existe. Nessa seção, vamos discutir um método que você pode usar em casos como esse.

Suponhamos que você tenha interesse em investigar se o conhecimento metalinguístico influencia julgamentos de aceitabilidade de sentenças. Uma vez que grande parte das pesquisas que envolve julgamento de aceitabilidade são produzidos por linguistas, é plausível querer saber se o linguista com todo seu conhecimento metalinguístico é uma boa fonte para esse tipo de julgamento (em comparação ao falante que não possui tal conhecimento). Vários estudos mostram, na verdade, que o julgamento de aceitabilidade de linguistas é muito diferente do julgamento do falante comum (veja [Spencer \(1973\)](#), [Labov \(1975\)](#), ou [Greenbaum \(1976\)](#)). Vamos pensar em um cenário bem simples: você pede a 100 pessoas que julguem o grau de aceitabilidade de uma frase qualquer. Depois que eles julgarem a aceitabilidade da frase, você explica a cada um deles qual fenômeno linguístico que está investigando com aquela frase. Após feito isso, você pede a cada um que julgue novamente a aceitabilidade da mesma frase. A pergunta então que você se faria é: será que o conhecimento metalinguístico que você explicou ao participante irá influenciar o novo julgamento que ele fez da frase? As variáveis envolvidas aqui são:

- uma variável dependente (categórica ou nominal) BEFORE: ACCEPTABLE vs. BEFORE: UNACCEPTABLE;
- uma variável dependente (categórica ou nominal) AFTER: ACCEPTABLE vs. AFTER: UNACCEPTABLE;
- as variáveis são dependentes, uma vez que cada participante forneceu dois julgamentos.

tab

Para esse tipo de situação, vamos utilizar o teste de McNemar (ou o teste de Bowker). Esse teste está relacionado com o teste de Qui-quadrado apresentado anteriormente nas Seções 4.1.2.3 e 4.1.2.2 e envolve os seguintes passos:

Procedimentos

- Formulação da hipótese
- Cálculo das frequências que você esperaria para a H_0
- Verificação das pressuposições do teste:
 - os níveis de variáveis observados estão relacionadas de forma pareada
 - as frequências esperadas são maiores que 5
- Cálculo das estatísticas χ^2 , gl e p .

Primeiramente, as hipóteses:

H_0 : As frequências das duas formas possíveis de os sujeitos produzirem um julgamento na segunda tarefa de classificação que se difere do julgamento na primeira tarefa de classificação são iguais (ou menores $\chi^2 = 0$).

H_1 : As frequências das duas formas possíveis de os sujeitos produzirem um julgamento na segunda tarefa de classificação que se difere do julgamento na primeira tarefa de classificação não são iguais (ou menores $\chi^2 \neq 0$).

Para fazer esse teste, vamos trabalhar com dados fictícios resumidos na Tabela 29. Os dados podem ser encontrados em <_inputfiles/04-1-2-3_aceitabilidade.csv>. A Tabela 29 já sugere que parece ter havido uma mudança nos julgamentos: dos 100 julgamentos, apenas $31+17 = 48$ deles – nem a mentade – foram julgados da mesma maneira nos dois julgamentos. Mas, o que você quer mesmo saber é se a forma como os 52 julgamentos mudaram é significativamente diferente do acaso.

Código em R

```
> AccBeforeAfter=read.delim(file.choose())¶
> str(AccBeforeAfter); attach(AccBeforeAfter)¶
```

Tabela 29 – Frequências observadas nos dados fictícios sobre aceitabilidade

		AFTER		
		ACCEPTABLE	UNACCEPTABLE	Total (linhas)
BEFORE	ACCEPTABLE	31	39	70
	UNACCEPTABLE	13	17	30
Total (colunas)		44	56	100

O teste de McNemar envolve apenas os casos onde houve uma mudança de julgamento, isto é, células b e c da tabela de *input*. Se esses casos são distribuídos igualmente, então a distribuição esperada é aquela na Tabela 30.

Tabela 30 – Frequências esperadas nos dados fictícios de aceitabilidade

		AFTER		Total (linhas)
		ACCEPTABLE	UNACCEPTABLE	
BEFORE	ACCEPTABLE		26	
	UNACCEPTABLE	26		
Total (colunas)				

Com base na tabela acima, podemos observar que as frequências esperadas são maiores que 5, então você pode de fato aplicar o teste de McNemar. Assim como fizemos antes, devemos calcular o valor do qui-quadrado – fórmula (39) – e o gl – fórmula (40) – onde k é o número de linhas/colunas:

$$(39) \quad \chi^2 = \sum_{i=1}^n \frac{(\text{observado} - \text{esperado})^2}{\text{esperado}} = 13$$

$$(40) \quad gl = \frac{k(k-1)}{2} = 1$$

Também da mesma forma que fizemos anteriormente, basta procurarmos esse valor qui-quadrado na tabela com os valores críticos do qui-quadrado para verificarmos se o valor encontrado é significativo. Caso seja, rejeitamos a hipótese nula H_0 . Como você pode ver, o valor do qui-quadrado encontrado é maior que o valor crítico a $p = 0,05$ e gl relevante. Assim, rejeitamos a hipótese nula H_0 e aceitamos a hipótese alternativa H_1 .

Tabela 31 – Valores de χ^2 críticos para $p_{bilateral} = 0,05; 0,01$ e $0,001$ para $1 \leq gl \leq 3$

	p = 0,05	p = 0,01	p = 0,001
gl = 1	3,841	6,635	10,828
gl = 2	5,991	9,21	13,816
gl = 3	7,815	11,345	16,266

Os resultados devem ser apresentados da seguinte maneira: “De acordo com o teste de McNemar, a maneira como 52 participantes – de um total de 100 – mudaram o julgamento de aceitabilidade após terem conhecimento do propósito do experimento não é fruto do acaso: no segundo julgamento de aceitabilidade, o número de julgamentos do tipo ‘aceitável’ foi muito menor ($\chi^2 = 13$; $df = 1$; $p_{bilateral} < 0,001$).”

No programa R, o teste é mais fácil. Utilizamos a função `mcnemar.test`, que tipicamente requer dois argumentos:

- `x`: uma tabela bidimensional com os dados a serem testados;
- `correct=FALSE` ou `correct=TRUE` (definição padrão): quando o número de mudanças é menor que 30, uma correção de continuidade é recomendada.

Código em R

```
> mcnemar.test(table(BEFORE, AFTER), correct=FALSE)¶
McNemar's Chi-squared test
data: table(BEFORE, AFTER)
McNemar's chi-squared = 13, df = 1, p-value = 0.0003115
```

O resumo e as conclusões são obviamente as mesmas. Quando fizer esse teste para tabelas $k \times k$ onde $k > 2$, o nome do teste é às vezes chamado de Bowker.

Recomendação para estudo/leitura

- Recommendation(s) for further study [Sheskin \(2011, Teste 20\)](#) sobre o teste de McNemar, sua alternativa exata, a qual você pode calcular com `dbinom`
- [Sheskin \(2011, Teste 26\)](#) para a extensão de Cochran do teste de McNemar para testar três ou mais medições de uma variável dicotômica, a qual leva apenas algumas linhas de código para ser calculada no R - por que você não tenta escrever essa função?
- A função `runs.test (library(tseries))` para testar a aleatoriedade de sequências binárias

4.2 Dispersões

Algumas vezes, é necessário e/ou interessante checar não apenas as características gerais de uma distribuição, mas também características mais minuciosas. As duas características mais óbvias são as medidas de tendência central e as medidas de dispersão. Esta seção é sobre a dispersão – mais especificamente variância e desvio padrão – de uma variável. A seção 4.2 discute as medidas de tendência central.

Para algumas perguntas de pesquisa, é importante saber, por exemplo, se duas distribuições têm dispersões semelhantes ou diferentes. Em outras palavras, fazer duas distribuições se espalham em torno de suas médias de forma semelhante ou de uma forma diferente? Nós tocamos neste tópico um pouco mais cedo na Seção 3.1.3.6, mas para ilustrar essa questão mais uma vez, considere a Figura 46.

Figura 46 – Duas distribuições fictícias

A Figura 46 mostra duas distribuições com 10 pontos cada – a primeira representada por bolinhas e a segunda por “x’s”. A média de cada uma das distribuições está representada por uma linha horizontal (tracejada para o primeiro grupo). As linhas verticais, representam os desvios de cada um dos pontos e sua média. Como vc pode observar, as distribuições se diferem em relação à média ($média_{grupo2} =$

1,99; média_{grupo1} = 5,94) e também com relação a dispersão: as linhas verticais do grupo 1 são bem maiores do que as linhas verticais do grupo 2. A diferença na dispersão dos dados da Figura 46 é óbvia, mas nem sempre esse é o caso. Para isso, precisamos de testes estatísticos. Na seção 4.2.1, vamos discutir como testar se a dispersão de uma variável dependente do tipo intervalar/escala de proporção é significativamente diferente em dois grupos.

4.2.1 Teste de aderência para uma variável dependente (escala de proporção)

Para exemplificar esse teste, vamos voltar aos dados apresentados anteriormente sobre a aquisição de tempo e aspecto em Russo. Na Seção 4.1.1.1, analisamos o desenvolvimento da correlação entre tempo e aspecto em uma criança falante de russo. Dessa vez, iremos verificar se a variabilidade geral do uso de tempo e aspecto dessa criança é significativamente diferente da variabilidade encontrada em outras crianças. Suponhamos que, para essas outras crianças, você calculou uma variância de 0,025.

Para investigar essa questão, utilizamos o teste qui-quadrado. No nosso caso, temos as seguintes variáveis envolvidas:

- uma variável dependente TENSEASPECT com os valores V de Cramer;
- não há uma variável independente, pois não estamos interessados em testar se a distribuição da nossa variável TENSEASPECT é influenciada por (ou está correlacionada com) alguma outra variável.

Procedimentos

- Formular as hipóteses
- Calcular a variância da amostra
- Verificar as pressuposições: a população de onde a amostra foi retirada, ou pelo menos a distribuição da amostra, deve ser distribuída de maneira normal
- Calcular o valor χ^2 , os graus de liberdade gl , e a probabilidade de erro p

Como sempre, começamos pelas hipóteses:

H_0 : A variância dos dados da nossa criança em questão não se difere da variância das crianças investigadas anteriormente; sd^2 TENSEASPECT da nossa criança = sd^2 TENSEASPECT das crianças investigadas anteriormente. Em outras palavras sd^2 da nossa criança = 0,025, ou, o quociente das duas variâncias é 1.

H_1 : A variância dos dados da nossa criança em questão se difere da variância das crianças investigadas anteriormente; sd^2 TENSEASPECT da nossa criança $\neq$ sd^2 TENSEASPECT das crianças investigadas anteriormente. Em outras palavras sd^2 da nossa criança $\neq$ 0,025, ou, o quociente das duas variâncias não é 1.

Importe os dados para o programa R de <_inputfiles/04-2-1_tempo-aspecto.csv>.

Código em R

```
> RussianTensAsp=read.delim(file.choose())
> str(RussianTensAsp); attach(RussianTensAsp)
```


Como um próximo passo, você deve testar se a pressuposição desse teste Qui-quadrado é atendida e se os dados são de fato distribuídos de forma normal. Discutimos essas questões em detalhes acima, então executamos o teste aqui sem mais delongas.

Código em R

```
> shapiro.test(TENSE_ASPECT)¶
Shapiro-Wilk normality test
data:  TENSE_ASPECT
W = 0.9942, p-value = 0.9132
```

Obviamente, idêntico ao que encontramos na Seção 4.1.1.1 acima, o *valor - p* é 0,9132. Isso quer dizer que não podemos rejeitar a H_0 , e devemos considerar os dados distribuídos de maneira normal, e você pode calcular este teste Qui-Quadrado. Vamos agora calcular a variância dos dados:

Código em R

```
> var(TENSE_ASPECT)¶
[1] 0.01687119
```

Para testar se esse valor é significativamente diferente da variância conhecida de (0,025), devemos calcular o qui-quadrado como na fórmula (41).

$$(41) \quad \chi^2 = \frac{(n-1)\text{variância da amostra}}{\text{variância da população}}$$

Os graus de liberdade são $n - 1 = 116$. No programa R:

Código em R

```
> chi.squared=((length(TENSE_ASPECT)-1)*var(TENSE_ASPECT))/0.025¶
> chi.squared¶
[1] 78.28232
```

Como de costume, você mesmo pode criar esses valores críticos ou pode procurar esse valor qui-quadrado na tabela já bem conhecida.

Código em R

```
> qchisq(c(0.05, 0.01, 0.001), 116, lower.tail=FALSE)¶
```

Tabela 32 – Valores de χ^2 críticos para $p_{bilateral} = 0,05; 0,01$ e $0,001$ para $115 \leq gl \leq 117$

	p = 0,05	p = 0,01	p = 0,001
gl = 115	141,03	153,191	167,61
gl = 116	142,138	154,344	168,813
gl = 117	143,246	155,496	170,016

Uma vez que o valor encontrado (78,28) é muito menor que o valor crítico (142,138), concluímos que a diferença entre as variâncias não é estatisticamente significativa. Para calcular o *valor - p* exato, utilizamos:

Código em R

```
> pchisq(chi.squared, (length(TENSE_ASPECT)-1), lower.tail=FALSE)¶
[1] 0.9971612¶
```

Para reportar esse resultado, usaríamos algo do tipo: “De acordo com o teste qui-quadrado, a variância encontrada nos dados da criança recentemente investigada (0,017) não se difere significativamente da variância da outra criança investigada anteriormente (0,025): $\chi^2 = 78,28$; $gl = 116$; $p_{bilateral} > 0,05$.”

4.2.2 Uma variável dependente (escala de proporção) e uma variável independente (nominal)

No que diz respeito à dispersões, provavelmente o caso mais comum é quando queremos testar se a dispersão de dois grupos é a mesma ou não. Uma vez que o conceito de comparar dispersão ou variância não é muito comum, vamos tentar ilustrar essa situação com um exemplo da pesquisa em sócio-fonética. Gaudio (1994) estava interessado em investigar a intensidade sonora de homens homossexuais e heterossexuais. No entanto, ele não estava interessado na média da intensidade sonora, mas sim na variação dela entre homossexuais e heterossexuais. Nesse estudo, quatro homens heterossexuais e quatro homens homossexuais foram gravados enquanto liam dois textos em voz alta. Essas gravações foram tocadas para 14 sujeitos que tinham que adivinhar se o falante era homossexual ou não. Os participantes conseguiram distinguir a orientação sexual das gravações quase que perfeitamente. A única correlação que parecia explicar esse fato foi que os homens homossexuais exibiram uma variância de intensidade muito maior na leitura de um dos textos.

Para explorar o procedimento estatístico que utilizamos para investigar tais casos, vamos ver um exemplo da área de aquisição de segunda língua. Suponhamos que você esteja interessado em investigar como falantes nativos de uma língua e falantes não-nativos (mas altamente fluentes) dessa mesma língua se diferenciam em termos de sua performance na tarefa de “encontrar sinônimos”. Nesse tipo de tarefa, participantes veem palavras na língua-alvo e devem escrever um sinônimo para a palavra apresentada. Suponhamos ainda, que você não esteja interessado no número de sinônimos que cada grupo fornece – uma vez que os falantes não-nativos são altamente proficientes, esses números podem ser muito semelhantes. No entanto, você está interessado no tempo que cada participante levou para pensar no sinônimo da palavra apresentada. Esse tipo de investigação envolve:

- uma variável dependente SYNTIMES que consiste no tempo que cada participante demorou para pensar no sinônimo da palavra apresentada;
- uma variável independente nominal/categórica: SPEAKER: LEARNER e SPEAKER: NATIVE.

Esse tipo de pergunta deve ser investigada com o teste F para homogeneidade da variância entre dois ou mais grupos, o qual envolve os seguintes passos:

Procedimentos

- Formulação da hipótese
- Cálculo da variância da amostra; representação e inspeção do gráfico
- Verificação das pressuposições do teste:
 - a população de onde a amostra foi retirada é distribuída de maneira normal, ou pelo menos a amostra está distribuída de maneira normal
 - as amostras são independentes
- Cálculo da estatística dos teste F, graus de liberdade (gl) e p

Primeiramente, você formula as hipóteses. Observe que a hipótese alternativa não é direcional.

H_0 : Os tempos que os falantes não-nativos demoram para pensar em um sinônimo não variam de maneira diferente dos tempos que os falantes nativos demoram para pensar em um sinônimo; a razão de variâncias $F = 1$.

H_1 : Os tempos que os falantes não-nativos demoram para pensar em um sinônimo variam de maneira diferente dos tempos que os falantes nativos demoram para pensar em um sinônimo; $F \neq 1$.

Como exemplo, vamos utilizar os dados fictícios do arquivo `<_inputfiles/04-2-2_synonymtimes.csv>`:

Código em R

```
> SynonymTimes=read.delim(file.choose())  
> str(SynonymTimes); attach(SynonymTimes)
```

Você agora deve calcular a variância para os dois grupos e gerar o gráfico da Figura 47. A variabilidade dos dois grupos parece muito semelhante: as caixas têm tamanhos bastante semelhantes, mas os intervalos dos *whiskers* são um pouco diferentes; veja o arquivo de códigos para alguma exploração adicional com gráficos *fdce* mais precisos.

Código em R

```
> tapply(SYNTIMES, SPEAKER, var)  
Learner Native  
10.31731 14.15385  
> boxplot(SYNTIMES~SPEAKER, notch=TRUE)  
> rug(jitter(SYNTIMES), side=2)
```

O teste F requer uma distribuição normal da população ou, pelo menos, da amostra. Usamos novamente o teste Shapiro-Wilk, desta vez com `tapply`. Nada com o que se preocupar: ambas as amostras não se desviam significativamente da normalidade e você pode fazer um teste F. Este teste requer que você calcule o quociente das duas variâncias (tradicionalmente, mas não necessariamente - veja abaixo - a variância maior é usada como numerador). Agora calculamos a razão das duas variâncias, que acabou não sendo 1, mas um pouco próximo a esse valor.

Código em R

```
>tapply(SYNTIMES, SPEAKER, shapiro.test)  
$Learner  
Shapiro-Wilk normality test  
data: X[[1L]]  
W = 0.9666, p-value = 0.2791  
$Native  
Shapiro-Wilk normality test  
data: X[[2L]]  
W = 0.9751, p-value = 0.5119  
> F.value=var(SYNTIMES[SPEAKER=="Native"])/  
var(SYNTIMES[SPEAKER=="Learner"]); F.value  
[1] 1.371855
```

Figura 47 – *Boxplot* para SYNTIMES~SPEAKER

Para saber se esse valor é significativamente diferente de 1, você precisa levar em consideração os graus de liberdade, desta vez dois: um para o numerador e outro para o denominador. Ambos podem ser calculados facilmente: basta subtrair 1 do tamanho amostral; veja a fórmula em (42).

$$(42) \quad gl_{\text{numerador}} = n_{\text{amostra do numerador}} - 1; gl_{\text{denominador}} = n_{\text{amostra do denominador}} - 1$$

O resultado é 39 nos dois casos. Agora podemos consultar o resultado na tabela F de valores críticos.

Tabela 33 – Valores críticos de F para $p_{\text{bilateral}} = 0,05$ e $38 \leq gl_{1,2} \leq 40$

	$gl_2 = 38$	$gl_2 = 39$	$gl_2 = 40$
$gl_1 = 38$	1,907	1,8963	1,8862
$gl_1 = 39$	1,9014	1,8907	1,8806
$gl_1 = 40$	1,8961	1,8854	1,8752

Obviamente, o resultado não é estatisticamente significativo: o valor F encontrado é menor que o valor crítico tabulado para $p = 0,05$ (1,8907). Como sempre, você pode calcular o valor crítico utilizando a função `qf`. Para isso, você precisará de quatro argumentos:

- `p`: o valor p do qual pretende determinar o valor crítico de F;
- `df1` e `df2`: os graus de liberdade do numerador e do denominador para determinar o valor F crítico;
- `lower.tail=FALSE`: considerar a área abaixo da curva acima / direita do valor F relevante.

Há um último ponto que deve ser comentado. Quando discutimos os testes unicaudal e bicaudal na Seção 1.3.4 acima, eu mencionei que na representação gráfica dos testes unilaterais (cf. Figura 6 e Figura 8) você adiciona as probabilidades dos eventos que você vê quando você se distancia da expectativa da hipótese nula em uma direção, enquanto que na representação gráfica de testes bicaudais (cf. Figura 7 e Figura 9) você adiciona as probabilidades dos eventos que você vê quando você se distancia da expectativa da hipótese nula em ambas as direções. A consequência disso é que o conhecimento prévio

que lhe permitiu formular a hipótese alternativa direcional foi compensado de forma tal que você precisou de resultados menos extremos para obter um resultado significativo. Isso também quer dizer, porém, que quando você deseja calcular um valor- p bicaudal usando `lower.tail=F`, você precisa do valor- p para $\frac{0,05}{2} = 0,025$. Esse valor lhe diz qual valor- F delimita 0,025 no lado direito do gráfico, mas, uma vez que o teste bicaudal requer também que você reduza a mesma área no lado esquerdo, esse é também o valor- F crítico desejado para valor- $p_{bilateral} = 0,05$. A Figura 48 ilustra essa lógica:

Figura 48 – Função de densidade para uma distribuição F com $gl_1 = gl_2 = 39$, teste bicaudal

Como mencionado acima, a expectativa de H_0 é que $F = 1$. A linha vertical à direita indica o valor- F que você precisa obter para um teste bicaudal significativo com $dl_{1,2} = 39$; esse valor- F é aquele que você já conhece da Tabela 33 – 1,8907 – o que quer dizer que você obtém um resultado bicaudal significativo se uma variância for 1,8907 vezes maior do que a outra. A linha vertical à esquerda indica o valor- F que você precisa obter para um teste unicaudal significativo com $gl_{1,2} = 39$; esse valor- F é 1,7045, o que significa que você obtém um resultado unicaudal significativo se a variância que você prevê for 1,7045 vezes maior do que a outra. Para você mesmo calcular os valores- F para os testes bicaudais, como um iniciante você pode querer apenas executar estas três linhas e prosseguir de forma semelhante para todas as outras células da Tabela 33.

Código em R

```
> qf(0.025, 39, 39, lower.tail=TRUE)¶
[1] 0.5288993
> qf(0.025, 39, 39, lower.tail=FALSE)¶
[1] 1.890719
```

O valor- F observado é obviamente muito pequeno tanto para um resultado direcional quanto não direcional significativo: $1,53 < 1,89$. É mais prático, no entanto, calcular o valor- p específico para o valor- F encontrado. Para isso, utilizamos o contrário da função `qf`, a função `pf`. Assim, devemos multiplicar por 2 e não dividir:

Código em R

```
> 2*pf(F.value, 39, 39, lower.tail=FALSE)¶
[1] 0.3276319
```

Como vimos, com um valor- p de $p = 0.3276$, o valor- F de cerca de 1,37 para $gl_1 = gl_2 = 39$ obviamente não é significativo. A função para o teste F no R que facilmente lida com essas questões é chamada `var.test`, e requer pelo menos dois argumentos, as duas amostras, como muitas outras funções, você pode abordar essa de duas formas: você pode fornecer uma fórmula ao R,

Código em R

```
> var.test(SYNTIMES~SPEAKER)¶
F test to compare two variances
data: SYNTIMES by SPEAKER
F = 0.7289, num df = 39, denom df = 39, p-value = 0.3276
alternative hypothesis: true ratio of variances is not
equal to 1
95 percent confidence interval:
 0.385536 1.378221
sample estimates:
ratio of variances
 0.7289402
```

ou você pode usar uma alternativa baseada em um vetor:

Código em R

```
> var.test(SYNTIMES[SPEAKER=="Learner"], SYNTIMES[SPEAKER=="Native"])¶
```

Não fique confuso se o valor- F que você obtiver no R não for igual ao que você mesmo calculou. Salvo erros, o valor gerado pelo R é o valor- $\{1/F\}$ – o R não coloca a variância maior automaticamente no numerador, mas a variância cujo nome aparece primeiro no alfabeto, que no caso discutido é “LEANER”, (antes de “NATIVE”). O valor- p lhe mostra então que o resultado do R é igual ao seu. Você pode resumir isso da seguinte forma: “Os tempos que os nativos demoram para encontrar sinônimos exibem uma variância que é aproximadamente 40% maior do que os tempos dos aprendizes (14,15 vs. 10,32), mas de acordo com um teste F , essa diferença não é significativa: $F = 0,73$; $gl_{learner} = 39$; $gl_{native} = 39$; valor- $p_{bicaudal} = 0,3276$.”

Recomendação para estudo/leitura

- [Dalgaard \(2002, p. 89\)](#), [Crawley \(2007, p. 289\)](#), [Baayen \(2008, Seção 4.2.3\)](#), [Sheskin \(2011, Teste 3, 11a\)](#)
- a função `fligner.test` para testar a homogeneidade da variância quando os dados violam o pressuposto de normalidade
- [Good \(2005, p. 61, 67-70\)](#) para outras possibilidades para comparar variâncias
- veja o arquivo de códigos para a função `exact.f.test.indep` que eu escrevi para calcular uma versão exata desse teste F , a qual você pode usar quando os tamanhos das suas amostras são muito pequenos (talvez <15); Cuidado, esse teste pode demorar algum tempo

4.3 Médias

Provavelmente, os testes de significância mais utilizados depois dos testes Qui-quadrado são os testes para comparação de médias. Na seção 4.3.1, vamos focar em testes de aderência, ou seja, testes em que verificamos se a medida de tendência central observada se difere estatisticamente de outra medida de tendência central já conhecida (o mesmo tipo de pergunta da seção 3.1.6.1); na seção 4.3.2, iremos falar de testes onde verificamos se duas medidas de tendência central de duas amostras distintas se diferem estatisticamente uma da outra.

4.3.1 Testes de Aderência

4.3.1.1 Uma variável dependente (variável de escala de proporção)

Vamos assumir que estamos novamente interessados no uso de partículas eufemísticas. Estudos mostram que homens e mulheres são diferentes no que diz respeito à frequência de uso de partículas eufemísticas. Vamos assumir que sabemos (com base na literatura sobre o assunto) que em experimentos anteriores, mulheres utilizam, em média, 12 partículas eufemísticas em conversa de dois minutos. Vamos assumir também que sabemos que a distribuição da frequência de partículas eufemísticas é normal. Imagine que você fez um experimento em que gravou 30 conversas de dois minutos entre mulheres e observou que elas, no seu experimento (assumindo que é uma replicação dos experimentos anteriores). Você quer então testar se a média de partículas eufemísticas que encontrou no seu experimento é significativamente diferente da média 12 encontrada na literatura. Essa pergunta envolve:

- uma variável dependente: HEDGES, que corresponde ao número de partículas eufemísticas e que será comparado com o encontrado na literatura;
- nenhuma variável independente, uma vez que não está comparando se o HEDGES é influenciado por alguma outra coisa.

Para tais casos, utilizamos um teste t para uma única amostra. O teste envolve os seguintes passos:

Procedimentos

- Formular hipóteses
- Testar as pressuposições do teste: a população da qual a média amostral foi tirada ou pelo menos a amostra em si tem uma distribuição normal
- Calcular o teste estatístico t , os graus de liberdade gl , e a probabilidade de erro p

Como sempre, começamos com as hipóteses:

H_0 : A média de HEDGES nas conversas observadas não é significativamente diferente da média conhecida: a média de partículas observadas = 12, ou a média de partículas observadas - 12 = 0, ou $t = 0$.

H_1 : A média de HEDGES nas conversas observadas não é significativamente diferente da média conhecida: a média de partículas observadas $\neq$ 12, ou a média de partículas observadas - 12 $\neq$ 0, $t \neq 0$

Vamos importar os dados <_inputfiles/04-3-1-1_hedges.csv>:

Código em R

```
> Hedges=read.delim(file.choose())¶
> str(Hedges); attach(Hedges)¶
```

Em seguida, você calcula a frequência média de partículas eufemistas que você encontrou no seu experimento bem como uma medida de dispersão (veja o arquivo de códigos para um gráfico):

Código em R

```
> mean(HEDGES); sd(HEDGES)¶
[1] 14.83333
[1] 2.506314
```

Uma vez que a literatura relata que a média de partículas eufemísticas é distribuída de maneira normal, você testa para saber se a sua amostra também segue a distribuição normal:

Código em R

```
> shapiro.test(HEDGES)¶
Shapiro-Wilk normality test
data: HEDGES
W = 0.946, p-value = 0.1319
```

Sim, ela está distribuído de maneira normal. Você pode, portanto, proceder imediatamente para o cálculo do valor t de acordo com a fórmula (43).

$$(43) \quad t = \frac{\bar{x}_{amostra} - \bar{x}_{população}}{sd_{amostra} / \sqrt{n_{amostra}}}$$

Código em R

```
> (mean(HEDGES)-12) / (sd(HEDGES)/sqrt(length(HEDGES)))¶
[1] 6.191884
```

Para saber o que esse valor significa, precisamos novamente dos graus de liberdade. Neste caso fica fácil, pois $gl = n-1$, ou seja, $gl = 29$. Se verificamos o valor crítico de t para $gl = 29$ na tabela já conhecida, devemos verificar se o valor- t observado é maior que o valor crítico com $p=0,05$. Para calcular o valor- p crítico, você precisa utilizar a função `qt` com o valor p e os graus de liberdade. Uma vez que o teste é bilateral, o valor- p deve ser reduzido $^{0,05/2} = 2,5\%$ em cada um dos lados da distribuição, o que é ilustrado na 49.

Tabela 34 – Valores críticos de t para $p_{bilateral} = 0,05, 0,01, \text{ e } 0,001$ for $28 \leq gl \leq 30$

	p = 0,05	p = 0,01	p = 0,001
gl = 28	2,0484	2,7633	3,6739
gl = 29	2,0452	2,7564	3,6594
gl = 30	2,0423	2,75	3,646

Portanto, o valor crítico para $p = 0,025$ e $gl = 29$ é:

Figura 49 – Função de Densidade para a distribuição t para $gl = 29$, teste bicaudal

Código em R

```
> qt(c(0.025, 0.975), 29, lower.tail=FALSE)¶
[1] 2.045230 -2.045230
```

O valor- p exato pode ser calculado com a função `pt` e o valor t observado é altamente significativo: 6,1919 não é só maior que 2,0452, mas é também muito maior que o valor crítico para $p = 0,001$ e $gl = 29$. Você também poderia ter inferido isso porque o valor- t de 6,19 está muito à direita na Figura 49.

Código em R

```
> 2*pt(6.191884, 29, lower.tail=FALSE)¶
[1] 9.42153e-07
```

Para resumir: “em média, as mulheres no seu experimento falaram 14,83 partículas eufemísticas (DP = 2,51). De acordo com o teste t para uma única amostra, essa média é estatisticamente muito diferente da média encontrada na literatura: $t = 6,1919$; $gl = 29$; $p_{bilateral} < 0,001$.”

Como sempre, no programa R, o cálculo do teste t pode ser feito de maneira bem mais simples, com apenas uma linha de comando. A função de interesse neste caso é `t.test`, a qual requer os seguintes argumentos:

- `x`: vetor com os dados;
- `mu=ldots` , a média da população com a qual você deseja comparar a média de `x`;
- `alternative="two-sided"` para testes bilaterais (configuração padrão) ou `alternative="greater"` e `alternative="less"` , para testes unilaterais.

Código em R

```
> t.test(HEDGES, mu=12)¶
One Sample t-test
data: HEDGES
t = 6.1919, df = 29, p-value = 9.422e-07
alternative hypothesis: true mean is not equal to 12
95 percent confidence interval:
13.89746 15.76921
sample estimates:
mean of x
14.83333
```

Você obtém a média já conhecida de 14,83 bem como o valor de g_l e o valor- t calculados semi-manualmente. Além disso, você também tem o valor- p exato e também o intervalo de confiança da média que não contém o valor 12.

Recomendação para estudo/leitura

Baayen (2008, Seção 4.1.2), Sheskin (2011, Teste 2)

4.3.1.2 Uma variável dependente (variável ordinal)

Na seção anterior, discutimos como testar se a média de uma amostra retirada de uma população distribuída de maneira normal é estatisticamente diferente de uma média já conhecida. Nessa seção, vamos ver como testar essa mesma hipótese quando os dados não são distribuídos de maneira normal, ou quando os dados não são de variável de escala de proporção (*ratio-scaled*). Vamos discutir esse teste analisando um fenômeno morfológico interessante: quando parte de duas palavras são utilizadas para formar uma terceira palavra. Existem dois tipos básicos que representam esse fenômeno: mesclagens e truncamentos complexos. Em (44a) temos alguns exemplos (do inglês) de mesclagens e (44b) alguns exemplos de truncamentos complexos.

Exemplo(44)

- a *brunch* (breakfast × lunch), *motel* (motor × hotel), *smog* (smoke × fog), *foolosopher* (fool × philosopher)
- b *scifi* (science × fiction), *fedex* (federal × express), *sysadmin* (system × administrator)

Uma pergunta que se faz é: será que esse fenômeno é influenciado pelo grau de similaridade entre as palavras utilizadas como fonte? Existem várias maneiras de medir similaridade entre palavras, mas a que vamos utilizar aqui é uma conhecida como coeficiente Dice (cf. BREW; MCKELVIE, 1976). É fácil calcular o coeficiente Dice para duas palavras. Primeiro, você separa a palavra em letras, fonemas, ou bigramas. Para *motel* (*motor* × *hotel*), você obtém:

- *motor*: *mo*, *ot*, *to*, *or*;
- *hotel*: *ho*, *ot*, *te*, *el*.

Em seguida, você conta o número de bigramas de uma palavra que também ocorre na outra. Nesse caso, há dois: *ot* de *motor* ocorre em *hotel*, e, portanto, *ot* de *hotel* ocorre em *motor*.⁸ Esse número, 2, é então dividido pelo número total de bigramas. O resultado é o coeficiente Dice:

⁸ No R, tais cálculos podem ser facilmente automatizados e feitos para centenas de milhares de palavras. Por exemplo, para cada palavra contida em um vetor *a*, esta linha de comando retorna todos os seus bigramas: `substr(rep(a, nchar(a)-1), 1:(nchar(a)-1), 2:(nchar(a)))`¶; para diversas aplicações, cf. Gries (2009a).

$$(45) \quad Dice_{\text{motor \& hotel}} = \frac{2}{8} = 0,25$$

Em outras palavras, o coeficiente Dice representa o percentual de bigramas comuns dentre todos os bigramas. Vamos então investigar se as palavras fonte envolvidas nos processos de mesclagem e mistura complexa são mais semelhantes entre si do que palavras em geral. Imagine que você sabe que a média do coeficiente Dice para um grupo de palavras escolhidas aleatoriamente é 0,225 (com desvio padrão de 0,0809; mediana 0,151 e intervalo interquartil de 0,125). Esses dados já sugerem que a distribuição não deve ser normal.⁹

O estudo envolve:

- uma variável dependente: Similaridade das palavras-fonte que será comparada com a média/mediana conhecida;
- não há variável independente, uma vez que não está interessado em verificar se Similaridade é influenciado por alguma coisa.

As hipóteses devem ser objetivas:

H_0 : A média de SIMILARITY não é significativamente diferente do coeficiente Dice médio: coeficiente Dice das palavras-fonte = 0,225, ou coeficiente Dice das palavras fonte 0,225 = 0.

H_1 : A média de SIMILARITY não é significativamente diferente do coeficiente Dice médio: coeficiente Dice das palavras fonte $\neq$ 0,225, ou coeficiente Dice das palavras fonte 0,225 $\neq$ 0.

Vamos importar os dados para o R a partir de <_inputfiles/04-3-1-2_dices.csv>; Esses dados são semelhantes aos dados analisados em [Gries \(2006\)](#).

Código em R

```
> Dices=read.table(choose.files())  
> str(Dices); attach(Dices)
```

Com base nos resumos das estatísticas, você já poderia inferir que as semelhanças entre palavras escolhidas aleatoriamente não são normalmente distribuídas. Podemos, portanto, assumir que isso também é verdade para a amostra de palavras-fonte, mas é claro que você também testar essa pressuposição (ver o arquivo de códigos para um gráfico):

Código em R

```
> shapiro.test(DICE)  
Shapiro-Wilk normality test  
data: DICE  
W = 0.9615, p-value = 0.005117
```

Os coeficientes Dice não estão distribuídos de maneira normal, mas estão distribuídos de maneira simétrica (como você pode ver também no gráfico ecdf). Portanto, embora os coeficientes Dice sejam uma variável de escala de proporção e apesar do tamanho da amostra ser relativamente grande (maior que 30), devemos ser cuidadosos ao aplicar o teste t para uma amostra. Podemos usar, por exemplo, o chamado teste do sinal para uma amostra para calcularmos a mediana. O teste envolve os seguintes passos:

⁹ Para dados autênticos, cf. [GRIES, 2006](#), onde eu calculei os coeficientes de Dice para todos os 499,500 pares possíveis de 1,000 palavras escolhidas aleatoriamente.

Procedimentos

Formular as hipóteses
 Calcular as frequências dos sinais das diferenças entre os valores observados e a média esperada
 Calcular a probabilidade de erro p

Você deve primeiramente reformular as hipóteses; Eu apenas fornece novas hipóteses estatísticas:

H_0 : mediana_{Coefficientes Dice das palavras fonte} = 0,151.

H_1 : mediana_{Coefficientes Dice das palavras fonte} $\neq$ 0,151.

Então, você calcula a estatística descritiva: a mediana e seu intervalo interquartil. Obviamente, a mediana dos coeficientes Dice observada é um pouco maior que 0,151, a mediana dos coeficientes Dice das palavras escolhidas de maneira aleatória. No entanto, é impossível supor se essa diferença é significativa ou não.

Código em R

```
> median(DICE); IQR(DICE)¶
[1] 0.1775
[1] 0.10875
```

Por isso, precisamos fazer o teste do sinal. Primeiramente, determinamos quantos coeficientes observados estão acima e quantos estão abaixo da mediana esperada. Se a mediana esperada for mesmo um bom parâmetro para os dados, então devemos esperar que 50% dos coeficientes estejam acima dessa mediana e 50% esteja abaixo dela. (NB: Você deve perceber que isso significa que os tamanhos exatos dos desvios da mediana esperada não são considerados aqui - você só analisa se os valores observados são maiores ou menores do que a mediana esperada, mas não quanto maior ou menor.)

Código em R

```
> sum(DICE>0.151); sum(DICE<0.151)¶ [1] 63
[1] 37
```

63 dos 100 valores observados são maiores do que a mediana esperada (o restante é menor do que a mediana esperada) - uma vez que você esperava 50, parece que os coeficientes Dice observados em suas palavras-fonte são significativamente maiores do que aqueles de palavras escolhidas aleatoriamente. Como antes, essa questão também pode ser abordada graficamente, usando a lógica e a função `dbinom` da Seção 1.3.4.1, Figura 7. A Figura 50 mostra as probabilidades de todos os resultados possíveis que você pode obter em 100 experimentos - porque você analisa os coeficientes Dice de 100 formações subtrativas. Primeiramente, considere o painel esquerdo da Figura 50.

De acordo com H_0 , você esperaria que 50 coeficientes Dice fossem maiores do que a mediana esperada, mas você encontrou 63. Assim, você adiciona a probabilidade do resultado observado (a barra preta para 63 de 100) às probabilidades de todos os coeficientes que se desviam de H_0 ainda mais extremamente, ou seja, as chances de encontrar 64, 65, ..., 99, 100 coeficientes Dice de 100 que são maiores do que a mediana esperada. A soma dessas probabilidades do painel esquerdo é aproximadamente igual a 0,006.

Figura 50 – Distribuições de probabilidade para 100 testes binomiais

Código em R

```
> sum(dbinom(63:100, 100, 0.5))  
[1] 0.006016488
```

Mas você ainda não terminou ... Como você pode ver no painel esquerdo da Figura 50, até agora você só incluiu os desvios de H_0 em uma direção - mas a sua H_1 é não-direcional, ou seja, bicaudal. Portanto, você deve incluir também as probabilidades dos eventos que se desviam o mesmo tanto e mais da H_0 na outra direção: 37, 36, ..., 1, 0 coeficientes Dice de 100 que são menores do que a mediana esperada, como representado no painel direito da Figura 50. A soma das probabilidades é a mesma (porque a distribuição das probabilidades binomiais em torno de $p=0,5$ é simétrica).

Código em R

```
> sum(dbinom(37:0, 100, 0.5))  
[1] 0.006016488
```

Novamente: se você espera 50 de 100, mas observa 63 de 100, e quer fazer um teste bicaudal, você deve adicionar a probabilidade somada de encontrar de 63 a 100 coeficientes Dice maiores (as 38 probabilidades superiores/à direita) à probabilidade somada de encontrar de 0 a 37 coeficientes Dice menores (as 38 probabilidades inferiores/à esquerda). Como resultado, você obtém: $p_{bilateral}$ de 0,01203298, o que é obviamente significativo. Para resumir seu resultado, você pode escrever da seguinte forma: “Em uma investigação de 100 formações de palavra subtrativas, foi encontrada uma média de similaridade de 0,1775 (mediana, $IQR = 0,10875$). 63 das 100 palavras foram mais semelhantes entre si do que o esperado com base em pares aleatórios de palavras, o que, de acordo com o teste de sinal, é um desvio significativo da média de similaridade de pares aleatórios de palavras (mediana = 0,151, $IRQ = 0,125$): $p_{binomial} = 0,012$.”

Lembre-se de que o teste de sinal de uma amostra apenas usa informações nominais, seja cada ponto de dados maior ou menor do que a mediana de referência esperada. Se a distribuição dos dados for bastante simétrica – como é aqui – então existe um teste alternativo que também considera os tamanhos dos desvios, isto é, usa pelo menos informações ordinais. O chamado Teste de Postos Sinalizados para uma amostra pode ser calculado usando-se a função `wilcox.test`. Além do vetor a ser testado, os seguintes argumentos são relevantes:

- `alternativa`: uma sequência de caracteres informando qual hipótese alternativa você quer testar: o padrão é "two.sided", outros valores possíveis para testes unicaudais são "less" ou "greater", os quais especificam como o vetor primeiramente nomeado se relaciona com a mediana de referência especificada.
- `mu=...`: a mediana de referência esperada de acordo com H_0 ;
- `exact=TRUE`, se você quer calcular um teste exato (em vez de uma estimativa; apenas quando o tamanho da sua amostra for menor que 50) ou `exact=FALSE`, se um teste assintótico for suficiente, o último é o padrão;
- `correct=TRUE` para uma correção de continuidade ou `correct=FALSE` para nenhuma correção;
- `conf.level`: um valor entre 0 e 1 especificando o tamanho do intervalo de confiança; o padrão é 0,95.

Uma vez que você tem uma hipótese alternativa não direcional, você faz um teste bicaudal adotando a configuração padrão para alternativa:

Código em R

```
> wilcox.test(DICE, mu=0.151, correct=FALSE)¶
Wilcoxon signed rank test
data: DICE
V = 3454.5, p-value = 0.001393
alternative hypothesis: true location is not equal to 0.151
```

O teste confirma o resultado anterior: tanto o teste de sinal com uma amostra, que apenas se preocupa com as direções dos desvios, quanto o Teste de Postos Sinalizados para uma amostra, que também considera os tamanhos dos desvios, indicam que as palavras fonte das formações de palavras por processos de mesclagem e mistura complexa são mais semelhantes entre si do que palavras fonte aleatórias. Isso deve, entretanto, encorajar você a certificar-se de que formulou exatamente a hipótese pela qual você se interessa (e depois usar o teste adequado).

Recomendação para estudo/leitura

- [Baayen \(2008, Seção 4.1.2\)](#), [Sheskin \(2011, Teste 9b, 6\)](#)

4.3.2 Testes para diferenças/independência

Um cenário particularmente frequente requer que você teste se dois grupos de elementos são diferentes em relação às suas tendências centrais. Como discutido acima, há vários fatores que determinam a escolha do teste:

- o tipo de amostras: dependentes ou independentes (cf. Seção [1.3.4.1](#));

- o nível de mensuração da variável dependente: intervalar/de escala de proporção versus ordinal;
- a distribuição da variável intervalar/de escala de proporção dependente: normal versus não normal;
- os tamanhos das amostras.

Para relembrar o discussão feita no início deste capítulo: se a variável de escala de proporção dependente bem como a distribuição normal ou ambos os tamanhos das amostras forem maiores que 30, ou se as diferenças entre as variáveis forem normalmente distribuídas, então você pode geralmente fazer um teste t (para amostras dependentes ou independentes, conforme necessário) – caso contrário você tem que fazer um teste U (para amostras independentes) ou um teste Wilcoxon (para amostras dependentes). A razão para esse processo de decisão é que, enquanto o teste t para amostras independentes requer, entre outros, amostras de distribuição normal, é possível mostrar que médias de amostras de 30 ou mais elementos apresentam usualmente uma distribuição normal mesmo se as amostras não apresentarem tal distribuição, o que explica porque na Seção 4.3.1.2 nós pelo menos consideramos a opção de um teste t com uma amostra (e depois escolhemos o teste de sinal mais conversador ou o Teste de Postos Sinalizados para uma amostra). Portanto, se os dados satisfazem uma das duas condições, já é o suficiente. Estritamente falando, o teste t para amostras independentes também requer variâncias homogêneas, o que nós também testaremos, mas iremos discutir uma versão o teste t que pode ser usada para lidar com variâncias heterogêneas, o teste t segundo Welch.

4.3.2.1 Uma variável dependente (escala de proporção) e uma variável independente (nominal) (amostras independentes)

O teste t para amostras independentes é um dos testes mais amplamente utilizados. Para explorá-lo, nós usamos um exemplo da área da fonética. Suponhamos que você queira estudar a (bastante trivial) hipótese alternativa não direcional de que as frequências dos primeiros formantes de homens e mulheres são diferentes. Você então planeja um experimento no qual você grava a pronúncia de um conjunto relevante de palavras e/ou sílabas por homens e mulheres para análise posterior. Esse estudo envolve:

- uma variável de escala de proporção dependente, a saber as F1-FREQUENCIES, cujas médias lhe interessam ;
- uma variável nominal independente, a dizer SEX: MALE versus SEX: FEMALE;
- amostras independentes uma vez que, se cada sujeito fornece apenas um ponto de dados, os pontos de dados não são relacionados uns com os outros.

O teste a ser usado em tais situações é o teste t para amostras independentes, o qual envolve os seguintes passos:

Procedimentos

- Formulação das hipóteses
- Cálculo das médias relevantes; exame de um gráfico
- Teste das pressuposições do teste:
 - a população da qual a amostra foi tirada ou pelo menos a amostra apresenta distribuição normal (especialmente com amostras de $n < 30$)
 - as variâncias das populações das quais as amostras foram tiradas ou pelo menos as variâncias são homogêneas
- Cálculo da estatística do teste t , dos graus de liberdade gl , e das probabilidades de erro p

Você deve começar com as hipóteses.

H_0 : A frequência média F1 dos homens é a mesma da frequência média F1 das mulheres:

$\text{média}_{F1 \text{ frequência dos homens}} = \text{média}_{F1 \text{ frequência das mulheres}}$, ou

$\text{média}_{F1 \text{ frequência dos homens}} - \text{média}_{F1 \text{ frequência das mulheres}} = 0$, ou $t = 0$.

H_1 : A frequência média F1 dos homens não é a mesma da frequência média F1 das mulheres:

$\text{média}_{F1 \text{ frequência dos homens}} \neq \text{média}_{F1 \text{ frequência das mulheres}}$, ou

$\text{média}_{F1 \text{ frequência dos homens}} - \text{média}_{F1 \text{ frequência das mulheres}} \neq 0$, ou $t \neq 0$.

Os dados que você investigará aqui fazem parte dos dados de um experimento semelhante sobre as vogais em Apache. Primeiramente, você carrega os dados de `<_inputfiles/04-3-2-1_f1-freq.csv>` no R:

Código em R

```
> Vowels=read.delim(file.choose())
> str(Vowels); attach(Vowels)
```

Em seguida, você calcula as médias relevantes das frequências. Como de costume, a forma menos elegante de proceder é usar o comando `apply`.

Código em R

```
> tapply(HZ_F1, SEX, mean)
F      M
528.8548 484.2740
> tapply(HZ_F1, SEX, sd)
F      M
110.80099 87.90112
```

Para obter uma ideia melhor sobre como os dados são, você também pode imediatamente gerar um *boxplot*. Você estabelece os limites do eixo x de forma que ele varie de 0 a 1.000, para que todos os valores estejam incluídos e a representação seja maximamente imparcial; Além disso, você usa `rug` para gerar os valores das mulheres e dos homens à esquerda e à direita do eixo y, respectivamente; veja a Figura 51 e o arquivo de códigos para uma alternativa que inclui um *stripchart*.

Código em R

```
> boxplot(HZ_F1 ~ SEX, notch=T, ylim=c(0, 1000)); grid()
> rug(HZ_F1[SEX=="F"], side=2); rug(HZ_F1[SEX=="M"], side=4)
```

O próximo passo consiste em testar as pressuposições do teste *t*. A figura 51 sugere que os dados vão ao encontro das pressuposições. Em primeiro lugar, os *boxplots* para os homens e para as mulheres aparecem como se os dados tivessem uma distribuição normal: as medianas estão no meio das caixas e os whiskers estendem-se quase igualmente em ambas as direções. Além disso, as variâncias parecem ser muito semelhantes uma vez que os tamanhos das caixas e dos degraus são muito semelhantes. Porém, claro que você tem que testar isso, e você pode usar o já conhecido teste Shapiro-Wilk:

Figura 51 – *Boxplot* para $HZ_{F1} \sim SEX$

Código em R

```
> tapply(HZ_F1, SEX, shapiro.test)¶
$F
Shapiro-Wilk normality test
data: X[[1L]]
W = 0.987, p-value = 0.7723
$M
Shapiro-Wilk normality test
data: X[[2L]]
W = 0.9724, p-value = 0.1907
```

Os dados não se diferem significativamente da normalidade. Agora você testa a homogeneidade da variância com o teste-F da Seção 4.2.2 (cuja pressuposição de normalidade nós já testamos). As hipóteses desse teste são:

H_0 : A variância da primeira amostra é igual àquela da segunda amostra; $F=1$.

H_1 : A variância de uma das amostras é maior do que a da outra; $F \neq 1$.

O teste-F com o R gera o seguinte resultado:

Código em R

```
> var.test(HZ_F1~SEX) # with a formula
F test to compare two variances
data: HZ_F1 by SEX
F = 1.5889, num df = 59, denom df = 59, p-value = 0.07789
alternative hypothesis: true ratio of variances is not
equal to 1
95 percent confidence interval:
0.949093 2.660040
sample estimates:
ratio of variances
1.588907
```

A segunda pressuposição também é atendida: uma vez que o intervalo de confiança inclui 1 e $p > 0,05$ de forma que as variâncias não são significativamente diferentes entre si, e você pode calcular o teste t para amostras independentes. Esse teste envolve três estatísticas diferentes: a estatística de teste t , o número de graus de liberdade gl , e, claro, o valor- p . No caso do teste t aqui discutido, o teste t segundo Welch, o valor- t é calculado de acordo com a fórmula em (46), onde sd^2 é a variância, n é o tamanho da amostra, e os subscritos 1 e 2 referem-se às duas amostras de homens e mulheres.

$$(40) \quad t = \left| (\bar{x}_1 - \bar{x}_2) \div \sqrt{s_1^2/n_1 + s_2^2/n_2} \right|$$

Código em R

```
> t.numerator=mean(HZ_F1[SEX=="M"])-mean(HZ_F1[SEX=="F"])
> t.denominator=sqrt((var(HZ_F1[SEX=="M"])/
length((HZ_F1[SEX=="M"]))) + (var(HZ_F1[SEX=="F"])/
length((HZ_F1[SEX=="F"]))))
> t.value=abs(t.numerator/t.denominator)
```

Você obtém $t = 2.441581$. A fórmula para os graus de liberdade é um pouco mais complexa. Primeiramente, você precisa calcular um valor chamado c , e com c , você pode então calcular o gl . A fórmula para calcular c é exibida em (47), e o resultado de (47) é inserido em (48).

$$(41) \quad c = \frac{sd_1^2/n_1}{sd_1^2/n_1 + sd_2^2/n_2}$$

$$(42) \quad gl = \left(\frac{c^2}{n_1-1} + \frac{(1-c)^2}{n_2-1} \right)^{-1}$$

Código em R

```
> c.numerator=var(HZ_F1[SEX=="M"])/length((HZ_F1[SEX=="M"]))
> c.denominator=t.denominator^2
> c=c.numerator/c.denominator
> df.summand1=c^2/(length(HZ_F1[SEX=="M"])-1)
> df.summand2=((1-c)^2)/(length(HZ_F1[SEX=="F"])-1)
> df=(df.summand1+df.summand2)^-1
```

Você obtém $c = 0.3862634$ e $df \approx 112.1946$. Você pode então verificar o valor- t no tipo familiar de tabela- t (ver Tabela 34), ou você pode calcular o valor- t crítico no R (com `qt(c(0.025, 0.975), 112, lower.tail= F)`). Como anteriormente, para um teste bicaudal você calcula o valor- t para o valor- p de 0.025).

Tabela 35 – Valores- t críticos para $p_{bicaudal} = 0,05, 0,01, \text{ e } 0,001$ para $111 \leq df \leq 113$

	p = 0,05	p = 0,01	p = 0,001
df = 111	1,9816	2,6208	3,3803
df = 112	1,9814	2,6204	3,3795
df = 113	1,9812	2,62	3,3787

Como você pode ver, o valor- t observado é maior do que aquele tabulado para $p = 0,05$, porém menor do que aquele tabulado para $p = 0,01$: tal diferença entre as médias é significativa. O valor- p exato pode ser calculado com `pt` e para o presente caso bicaudal você simplesmente executa isto:

Código em R

```
> 2*pt(t, 112.195, lower.tail=FALSE)¶
[1] 0.01618811
```

No R, você pode fazer tudo isso com a função `t.test`. Essa função toma vários argumentos, sendo que os dois primeiros – as amostras relevantes – podem ser dados por meio de uma fórmula ou com dois vetores. Estes são os outros argumentos relevantes:

- `alternative`: uma sequência de caracteres que especifica qual hipótese alternativa é testada: o valor padrão, que pode portanto ser omitido, é `"two.sided"`. Outros valores para hipóteses unicaudais são novamente `"less"` ou `"maior"`. Como antes, o R considera o nível variável que vem primeiro em ordem alfabética (isto é, aqui "F") como a categoria de referência, de forma que a hipótese unicaudal de que os valores para homens são menores do que aqueles para as mulheres seria testada com `alternative="greater"`;
- `paired=FALSE` para o teste t com amostras independentes (o padrão) ou `paired=TRUE` para o teste t com amostras dependentes (ver a próxima seção);
- `var.equal=TRUE`, quando as variâncias das duas amostras são iguais, ou `var.equal=FALSE` se não forem iguais; o último é o padrão, e raramente deve ser alterado;
- `conf.level`: um valor entre 0 e 1 que especifica o intervalo de confiança da diferença entre as médias; o padrão é 0.95.

Então, para realizar o teste t com amostras independentes, você pode executar qualquer alternativa listada abaixo. Você obterá o seguinte resultado:

Código em R

```
> t.test(HZ_F1~SEX, paired=FALSE)¶
Welch Two Sample t-test
data: HZ_F1 by SEX
t = 2.4416, df = 112.195, p-value = 0.01619
alternative hypothesis: true difference in means is
not equal to 0
95 percent confidence interval:
 8.403651 80.758016
sample estimates:
mean in group F mean in group M
528.8548          484.2740
> t.test(HZ_F1[SEX=="F"], HZ_F1[SEX=="M"], paired=FALSE)¶
```

As primeiras duas linhas do resultado fornecem o nome do teste e os dados aos quais o teste foi aplicado. A linha 3 lista a estatística do teste t (o sinal é irrelevante porque ele depende apenas de qual média é subtraída de qual, mas é claro que ele deve ser considerado no cálculo manual), o valor gl , e o valor- p . A linha 4 mostra a hipótese alternativa testada. Depois, você obtém o intervalo de confiança para as diferenças entre as médias (e o nosso teste é significativo porque esse intervalo de confiança não inclui 0). Por último, você obtém as médias que você já conhece.

Você pode resumir seus resultados da seguinte maneira: “No experimento, a frequência F1 média das vogais produzidas pelos homens foi de 484,3 Hz (sd = 87,9), a frequência F1 média das vogais produzidas pelas mulheres foi de 528,9 Hz (sd = 110,8). De acordo com um teste t para amostras independentes, a diferença de 44,6 Hz entre as médias é estatisticamente significativa, mas não particularmente forte: $t_{Welch} = 2,4416$; $df = 112,2$; $p_{two-tailed} = 0,0162$.”

Na Seção 5.2.2, nós iremos discutir a extensão desse teste para casos nos quais você tem mais que uma variável independente e/ou nos quais a variável independente tem mais que dois níveis.

Recomendação para estudo/leitura

- [Crawley \(2007, p. 289\)](#), [Baayen \(2008, Seção 4.2.2\)](#), [Sheskin \(2011, Teste 11\)](#)
- veja o arquivo de códigos para a função `exact.t.test.indep` que escrevi para calcular uma versão exata desse teste F , a qual você pode usar quando os tamanhos das amostras são muito pequenos (talvez <15); Cuidado, esse teste pode demorar algum tempo (e requer a biblioteca `combinat`)

4.3.2.2 Uma variável dependente (escala de proporção) e uma variável independente (nominal)(amostras dependentes)

A seção anterior ilustrou um teste para médias de duas amostras independentes. O nome desse teste sugere que há um teste semelhante para amostras dependentes, e é isso que vamos discutir nesta seção com base em um exemplo de estudos de tradução. Suponhamos que você queira comparar as extensões de textos em inglês e em português com as extensões de suas respectivas traduções para o português e para o inglês. Imagine também que você suspeite que as traduções são em média maiores do que os textos originais. Essa questão envolve:

- uma variável de escala de proporção dependente, a dizer a LENGTH (extensão) dos textos, em cuja média estamos interessados;
- uma variável nominal/categorial independente, a dizer TEXTSOURCE: ORIGINAL versus TEXTSOURCE: TRANSLATION;
- amostras dependentes desde que os valores de LENGTH exista apenas uma tradução para cada texto original.

Realizar um teste- t para amostras dependentes requer os seguintes passos:

Procedimentos

- Formulação das hipóteses
- Cálculo de medidas descritivas e visualização de dados
- Teste das pressuposições do teste: as diferenças dos valores pareados são distribuídas de forma normal

- Cálculo da estatística do teste t , dos graus de liberdade gl , e da probabilidade de erro p

Como de costume, você deve formular as hipóteses, mas note que desta vez a hipótese alternativa é direcional: você suspeita que a extensão média dos originais é menor do que aquela das traduções, não apenas diferente (i.e., menor ou maior). Portanto, a forma estatística da hipótese alternativa contém não apenas um $\neq$, mas algo mais específico, “ $<$ ”:

H_0 : A média das diferenças pareadas entre as extensões dos originais e as extensões das traduções é 0; $\text{média}_{\text{diferenças pareadas}} = 0$; $t = 0$.

H_1 : A média das diferenças pareadas entre as extensões dos originais e as extensões das traduções é menor que 0; $\text{média}_{\text{diferenças pareadas}} < 0$; $t < 0$.

Observe em particular (i) que as hipóteses não envolvem os valores das duas amostras, mas a diferenças pareadas entre as amostras e (ii) como essas diferenças são calculadas: original menos tradução, não o contrário (e portanto nós usamos “ < 0 ”). Para ilustrar esse teste, nós iremos analisar dados de [Frankenberg-Garcia \(2004\)](#). Ela comparou as extensões de oito textos em inglês e oito textos em português, os quais foram escolhidos e editados de forma que suas extensões eram de aproximadamente 1.500 palavras, e depois determinou as extensões de suas traduções. Você pode carregar esses dados a partir de `<_inputfiles/04-3-2-2_textlengths.csv>`:

Código em R

```
> Texts=read.table(choose.files())  
> str(Texts);attach(Texts)
```

Observe que os dados são organizados de forma que a ordem dos textos e a ordem de suas traduções são idênticas: o caso 1 é um texto original em inglês (portanto, TEXT é 1, TEXTSOURCE é original, LANGUAGE é English), e o caso 17 é sua tradução (portanto, TEXT é novamente 1, mas TEXTSOURCE é agora a tradução, a LANGUAGE é Portuguese) etc. Primeiramente, você calcula as médias e gera o gráfico (note que esse boxplot não mostra a dependência das amostras).

Código em R

```
> tapply(LENGTH, TEXTSOURCE, mean)  
Original Translation  
1500.062 1579.938  
> boxplot(LENGTH~TEXTSOURCE, notch=T, ylim=c(0, 2000))  
> rug(LENGTH, side=2)
```

A extensão média dos textos traduzidos é um pouco maior do que aquela dos textos originais. Além disso, as duas amostras têm dispersões bem diferentes (apenas porque as extensões dos originais foram definidas para aproximadamente 1.500 palavras e, portanto, exibem pouca variação, enquanto que as extensões das traduções não foram controladas dessa forma).

Agora, este é realmente um gráfico ruim para representar os dados - por quê?

Pausa para pensar

Este gráfico não retrata a informação de que os dados da parte esquerda - as extensões dos textos originais - estão relacionados com aqueles da parte direita - as extensões de suas traduções! Assim,

Figura 52 – *Boxplot* para Length~Textsource

veja o arquivo de códigos para três gráficos melhores (especialmente o terceiro). Dadas as extensões controladas dos textos originais, a diferença aqui não é tão grande, mas em outras aplicações, um *boxplot* para amostras dependentes como o acima pode ser muito enganador.

Diferentemente do teste-*t* para amostras independentes, o teste-*t* para amostras dependentes não pressupõe uma distribuição normal ou homogeneidade de variância dos valores da amostra, mas uma distribuição normal das diferenças entre os pares de valores da amostra. Você pode criar um vetor com essas diferenças e aplicar o teste Shapiro-Wilk ao vetor utilizando uma linha com este atalho:

Código em R

```
> shapiro.test(differences=LENGTH[1:16]-LENGTH[17:32])¶
Shapiro-Wilk normality test
data: differences
W = 0.9569, p-value = 0.6057
```

As diferenças não são significativamente diferentes da normalidade, então você pode de fato realizar o teste-*t* para amostras dependentes. Primeiramente, você calcula o valor-*t* de acordo com a fórmula em (49), onde *n* é o número de pares de valores.

$$(43) \quad t = \frac{|\bar{x}_{diff}| \sqrt{n}}{sd_{diff}}$$

Código em R

```
> t.value=(abs(mean(differences))*sqrt(length(differences)))/sd(differences); t¶
> t.value
[1] 1.927869
```

Em seguida, você calcula os graus de liberdade *gl* (*df*), que é o número de diferenças *n* menos 1:

Código em R

```
> df=length(differences)-1; df
[1] 15
```

Primeiramente, você pode agora calcular os valores críticos para $p=0,05$ – desta vez não para $0,05/2 = 0,025$ – porque você tem uma H_1 direcional - com $gl = 15$ ou, de forma mais sofisticada, criar toda a tabela- t .

Código em R

```
> qt(c(0.05, 0.95), 15, lower.tail=FALSE)
[1] 1.753050 -1.753050
```

Depois, você pode olhar o valor- t em tal tabela- t , repetida aqui como Tabela 36. Uma vez que tais tabelas usualmente listam apenas os valores positivos, você usa o valor absoluto do seu valor- t . Como você pode ver, a diferença entre os textos originais e suas traduções é significativa, mas não muito ou altamente significativa: $1,927869 > 1,7531$, mas $1,927869 < 2,6025$.

Tabela 36 – Valores- t críticos para $p_{unilateral} = 0,05, 0,01, \text{ e } 0,001$ (para $14 \leq df \leq 16$)

	p = 0,05	p = 0,01	p = 0,001
df = 14	1,7613	2,6245	3,7874
df = 15	1,7531	2,6025	3,7328
df = 16	1,7459	2,5835	3,6862

Alternativamente, você pode calcular o valor- p exato. Uma vez que você tem uma hipótese alternativa adicional, você precisa apenas delimitar 5% da área abaixo da curva em um lado da distribuição. O valor- t resultante da hipótese nula é 0, e o valor- t que você calculou é aproximadamente -1,93, então você deve calcular a área abaixo da curva de 1,93 até $+\infty$; veja a Figura 53. Uma vez que você está fazendo um teste unicaudal, você não precisa multiplicar o valor- p por 2.

Código em R

```
> pt(t, 15, lower.tail=FALSE)
[1] 0.03651145
```

Observe que isso também quer dizer que a diferença é apenas significativa porque você fez um teste unicaudal – por causa da multiplicação com 2, um teste bicaudal não teria produzido um resultado significativo, mas $p = 0,07302292$.

Agora o mesmo teste com o R. Uma vez que você já conhece os argumentos da função `t.test`, nós podemos focar nas principais diferenças em relação a antes, nos fatos de que você agora tem uma hipótese alternativa direcional e precisa realizar um teste unicaudal, e que agora você fará um teste pareado. Para fazer isso de forma adequada, você deve primeiramente entender como o R calcula a diferença. Como já mencionado, o R procede seguindo a ordem alfabética e calcula a diferença ‘primeiro nível alfabeticamente falando menos segundo nível alfabeticamente falando’ (é por isso que a hipótese alternativa foi formulada como acima). Uma vez que “Original” vem antes de “Tradução”, e nós levantamos a hipótese de que a média dos textos originais seria menor do que a média das traduções, a diferença é menor que 0. Você, portanto, diz ao R que a diferença é “menor” que zero.

É claro que você pode usar a fórmula ou a anotação baseada no vetor. Eu mostro o resultado da anotação da fórmula, mas ambas as formas produzem o mesmo resultado. Você obtém o valor- t (o nosso foi positivo apenas porque usamos `abs`), o valor dos graus de liberdade, um valor- p , e um intervalo de confiança que, uma vez que ele não inclui 0, também reflete o resultado significativo.

Figura 53 – Função Densidade para uma distribuição t para $gl = 15$, teste unicaudal

Código em R

```
> t.test(LENGTH TEXTSOURCE, paired=TRUE, alternative="less")  
Paired t-test  
data: LENGTH by TEXTSOURCE  
t = -1.9279, df = 15, p-value = 0.03651  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf -7.243041  
sample estimates:  
mean of the differences  
-79.875  
> t.test(LENGTH[TEXTSOURCE=="Original"], LENGTH[TEXTSOURCE=="  
Translation"], paired=TRUE, alternative="less")
```

Resumindo: “Em média, os originais são menores que suas traduções em aproximadamente 80 palavras (o intervalo de confiança de 95% dessa diferença é -Inf, -7,24). De acordo com um teste- t para amostras dependentes, essa diferença é significativa: $t=-1,93$; $df=15$; $p_{unicaudal} = 0,0365$. Entretanto, o efeito é relativamente pequeno: a diferença de 80 palavras corresponde a apenas 5% da extensão dos textos.”

Recomendação para estudo/leitura

- [Crawley \(2007, p. 298\)](#), [Baayen \(2008, Seção 4.3.1\)](#), [Sheskin \(2011, Teste 17\)](#)
- veja o arquivo de códigos para a função `exact.t.test.dep` que eu escrevi para calcular um versão exata do teste- F , a qual você pode usar quando os tamanhos das amostras forem muito pequenos (talvez <15); cuidado, esse teste pode tomar muito tempo (para o exemplo visto, ele retorna quase o mesmo valor- p)

4.3.2.3 Uma variável dependente (ordinal) e uma variável independente (nominal) (amostras independentes)

Nesta seção, nós iremos discutir um teste não-paramétrico para duas amostras independentes de dados ordinais, o teste-U. Uma vez que eu mencionei no início da Seção 4.3.2 que o teste-U é não só usado quando as amostras a serem comparadas consistem em dados ordinais, mas também quando elas violam pressuposições distribucionais, esta seção irá novamente envolver um exemplo no qual apenas um teste desses pressupostos distribucionais lhe permite decidir qual teste usar.

Na Seção 4.3.1.2 acima, você analisou as similaridades de palavras fonte participando de formações de palavras subtrativas e você testou se essas similaridades eram em média diferentes da similaridade média conhecida entre palavras aleatórias. Os dados que você usou foram do tipo estudado em Gries (2006), mas no exemplo acima não foi feita nenhuma distinção entre palavras fonte participando de diferentes tipos de formações de palavras subtrativas. É isso que faremos aqui por meio da comparação de semelhanças de palavras fonte participando de mesclagens com semelhanças de truncamentos complexos. Se ambos os tipos de processos de formação de palavras forem diferentes de acordo com esse parâmetro, isso fornecerá motivação empírica para diferenciar esses processos. Esse exemplo, portanto, envolve

- uma variável de escala de proporção dependente, a dizer a SIMILARITY das palavras fonte em cuja média você está interessado;
- uma variável nominal independente, a dizer PROCESS: BLEND versus PROCESS: COMPLCLIP;
- amostras independentes uma vez que o coeficiente de Dice de qualquer par de palavras da fonte não tem nada a ver com qualquer outro par de palavras fonte.

Este tipo de pergunta seria normalmente investigada com o teste-*t* para amostras independentes discutido acima. De acordo com o procedimento acima, você primeiro formula as hipóteses (não de forma direcional, uma vez que podemos não ter uma razão *a priori* para assumir uma diferença particular):

H_0 : A média dos coeficientes de Dice das palavras fonte de mesclagens é tão grande quanto a média dos coeficientes de Dice das palavras fontes de truncamentos complexos;

$\text{média}_{\text{Coeficientes de Dice de mesclagens}} = \text{média}_{\text{Coeficientes de Dice de truncamentos complexos}}, \text{ ou}$
 $\text{média}_{\text{Dice coefficients of blends}} - \text{média}_{\text{Coeficientes de Dice de truncamentos complexos}} = 0.$

H_1 : A média dos coeficientes de Dice das palavras fonte de mesclagens não é tão grande quanto a média dos coeficientes de Dice das palavras fontes de truncamentos complexos;

$\text{média}_{\text{Coeficientes de Dice de mesclagens}} \neq \text{média}_{\text{Coeficientes de Dice de truncamentos complexos}}, \text{ ou}$
 $\text{média}_{\text{Coeficientes de Dice de mesclagens}} - \text{média}_{\text{Coeficientes de Dice de truncamentos complexos}} \neq 0.$

Você pode carregar os dados do arquivo <_inputfiles/04-3-2-3_dices.csv>. Como antes, esse arquivo contém os coeficientes de Dice, mas agora também, em uma coluna adicional, o processo de formação de palavras para cada coeficiente de Dice.

Código em R

```
> Dices=read.table(choose.files())  
> str(Dices); attach(Dices)
```

Como de costume, você deve começar pela exploração gráfica dos dados:

Código em R

```
> boxplot(DICE ~ PROCESS, notch=T, ylim=c(0, 1))¶
> rug(jitter(DICE[PROCESS=="Blend"]), side=2)¶
> rug(jitter(DICE[PROCESS=="ComplClip"]), side=4)¶
> text(1:2, tapply(DICE, PROCESS, mean), "+")¶
```

Figura 54 – *Boxplot* da Similaridade~Processo

Como de costume, este gráfico já revela informações suficientes para quase excluir a necessidade de análise estatística. O aspecto provavelmente mais óbvio é a diferença entre as duas medianas, mas, uma vez que os dados são de proporção de escala, você também precisa explorar as médias. Essas já estão marcadas no gráfico, e aqui está a linha habitual de código para computá-las diretamente; observe o tamanho da diferença entre elas.

Código em R

```
> tapply(DICE, PROCESS, mean)¶
Blend ComplClip
0.22996 0.12152
> tapply(DICE, PROCESS, sd)¶
Blend ComplClip
0.4274985 0.04296569
```

Para testar se o teste-*t* para amostras independentes pode ser usado aqui, nós precisamos testar os seus dois pressupostos, a normalidade nos grupos e a homogeneidade de variância. Uma vez que o teste-*F* para homogeneidade de variâncias pressupõe normalidade, você começa testando se os dados são distribuídos de forma normal. Os *rugs* da Figura 54 sugerem que eles não são, o que é confirmado pelo teste de Shapiro-Wilks.

Código em R

```
> tapply(DICE, PROCESS, shapiro.test)¶
$Blend
Shapiro-Wilk normality test
data: X[[1L]]
W = 0.9455, p-value = 0.02231
$ComplClip
Shapiro-Wilk normality test
data: X[[2L]]
W = 0.943, p-value = 0.01771
```

Por causa dessas violações de normalidade, você na verdade não pode fazer o teste- F regular para testar o segundo pressuposto do teste- t para amostras independentes. Você, portanto, faz o teste Fligner-Killeen de homogeneidade das variâncias, o qual não exige que os dados sejam distribuídos de forma normal e o qual eu mencionei na Seção 4.2.2 acima.

Código em R

```
> fligner.test(DICE~PROCESS)¶
Fligner-Killeen test of homogeneity of variances
data: DICE by PROCESS
Fligner-Killeen:med chi-squared=3e-04, df=1, p-value=0.9863
```

As variâncias são homogêneas, mas a normalidade ainda é violada. Assim, concluímos que, mesmo que os dados sejam de escala de proporção e mesmo que os tamanhos das amostras sejam maiores que 30, é provavelmente mais seguro calcular um teste que não parte dessas pressuposições, o teste- U .

Procedimentos

- Formulação das hipóteses
- Cálculo das medianas observadas, exame de um gráfico
- Teste das pressuposições do teste:
 - os valores são independentes uns dos outros
 - as populações das quais os valores foram tirados como amostra são distribuídas de forma idêntica¹⁰
- Cálculo das estatísticas do teste U e z , e p

Os dois *boxplots* são relativamente semelhantes e as variâncias dos dois grupos não são significativamente diferentes, e o teste U é relativamente robusto, portanto o utilizamos aqui. Uma vez que o teste U assume apenas dados ordinais, agora você calcula as medianas, não apenas as médias. Você, portanto, ajusta as suas hipóteses, calcula as medianas e os intervalos interquartílicos:

H_0 : A média dos coeficientes de Dice das palavras fonte de mesclagens é tão grande quanto a média dos coeficientes de Dice das palavras fonte de truncamentos complexos;
 $\text{mediana}_{\text{Coeficientes de Dice de mesclagens}} = \text{mediana}_{\text{Coeficientes de Dice de truncamentos complexos}}$, ou
 $\text{mediana}_{\text{Coeficientes de Dice de mesclagens}} - \text{mediana}_{\text{Dice coefficients of complex clippings}} = 0$.

H_1 : A média dos coeficientes de Dice das palavras fonte de mesclagens não é tão grande quanto a média dos coeficientes de Dice das palavras fonte de truncamentos complexos;
 $\text{mediana}_{\text{Dice coefficients of blends}} \neq \text{mediana}_{\text{Coeficientes de Dice de truncamentos complexos}}$, ou
 $\text{mediana}_{\text{Coeficientes de Dice de mesclagens}} - \text{mediana}_{\text{Coeficientes de Dice de truncamentos complexos}} \neq 0$.

Código em R

```
> tapply(DICE, PROCESS, median)¶
Blend ComplClip
0.2300 0.1195
> tapply(DICE, PROCESS, IQR)¶
Blend ComplClip
0.0675 0.0675
```

Aqui, as pressuposições podem ser testadas sem problemas: os valores são independentes uns dos outros, uma vez que nenhuma formação de palavras influencia a outra, que as distribuições dos dados na Figura 54 parecem ser bastante semelhantes, e que um teste de Kolmogorov-Smirnov dos valores de Dice padronizados z para ambos os processos de formação de palavras é completamente insignificante ($p = 0,9972$).

Infelizmente, calcular o teste- U é um mais complicado do que muitos outros testes. Primeiro, você transforma todos os coeficientes de Dice em posições, e então você calcula a soma de todas as posições para cada processo de formação de palavras. Em seguida, ambos os valores- T e os dois tamanhos das amostras são inseridos nas fórmulas em (50) e (51) para calcular dois valores- U , o menor dos quais é a estatística do teste exigida.

Código em R

```
> Ts = tapply(rank(DICE), PROCESS, sum)¶
```

$$(44) \quad U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - T_1$$

$$(45) \quad U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - T_2$$

Código em R

```
> n1=length(DICE[PROCESS=="Blend"])¶
> n2=length(DICE[PROCESS=="ComplClip"])¶
> U1=n1*n2+((n1*(n1+1))/2)-Ts[1]¶
> U2=n1*n2+((n2*(n2+1))/2)-Ts[2]¶
> U=min(U1, U2)¶
```

O valor- U resultante, 84, pode ser consultado em uma tabela U ou, porque existem poucas tabelas U para amostras maiores¹¹, convertido em um *score-z* de distribuição normal. Esse *score-z* é calculado em vários passos. Primeiramente, você usa as fórmulas em (52) e (53) para calcular um valor- U esperado e a sua dispersão.

$$(46) \quad U_{\text{esperado}} = 0.5 n_1 n_2$$

$$(47) \quad \text{Dispersão } U_{\text{esperado}} = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

Em segundo lugar, você insere esses valores juntamente com o valor- U observado na fórmula em (54).

¹¹ Bortz, Lienert e Boehnke (1990, p. 202, Tabela 6) fornecem valores- U críticos para $n \leq 20$ e mencionam referências para as tabelas com valores críticos para $n \leq 40$ – eu pelo menos não conheço nenhuma tabela- U para amostras maiores.

$$(48) \quad z = \frac{U - U_{esperado}}{\text{Dirpersão}U_{esperado}}$$

Código em R

```
> expU=n1*n2/2¶
> dispersion.expU=sqrt(n1*n2*(n1+n2+1)/12)¶
> z=abs((U-expU)/dispersion.expU)¶
```

Para decidir se a hipótese nula pode ser rejeitada, você consulta esse valor, 8,038194, em uma tabela Z como a Tabela 37, ou você calcula um *escores z* crítico para $p_{bicaudal} = 0,05$ com `qnorm` (como foi mencionado na Seção 1.3.4.3 acima). Uma vez que você tem uma hipótese alternativa não-direcional, você aplica a mesma lógica que acima e calcula *escores z* para metade dos $p_{bicaudal}$ - os valores nos quais você está interessado:

Tabela 37 – Valores críticos de Z (escores z) para $p_{bicaudal} = 0,05, 0,01$, and $0,001$

escores z	valor- p
1,96	0,05
2575	0,01
3291	1

Código em R

```
> qnorm(c(0.9995, 0.995, 0.975, 0.025, 0.005, 0.0005),lower.tail=FALSE)¶
[1] -3.290527 -2.575829 -1.959964 1.959964 2.575829 3.290527
```

É óbvio que o escore-z observado é não apenas muito maior do que aquele tabulado para $p_{bicaudal} = 0,001$, mas também está muito nítido na área cinzenta na Figura 55: a diferença entre as medianas é altamente significativa, como os *notches* (degraus) que não se sobrepõem já anteciparam. Além disso, agora você pode também calcular o valor- p exato com a habitual “função de espelho” de `qnorm`:

Figura 55 – Função Densidade da distribuição padrão normal; teste bicaudal

Código em R

```
> pnorm(z, lower.tail=FALSE)¶
[1] 4.558611e-16
```

No R, você calcula o teste- U com a mesma função que no teste de Wilcoxon, `wilcox.test` e, novamente, você pode ou usar uma fórmula ou dois vetores. Além desses argumentos, os seguintes também são úteis:

- `alternative`: uma sequência de caracteres especificando qual hipótese alternativa você deseja testar: o padrão é "two.sided", outros valores possíveis para testes unicaudais são novamente "less" (menos) ou "greater" (maior), os quais especificam como o vetor primeiramente nomeado ou nível de fator se relaciona ao segundo;
- `paired=FALSE` para o teste- U para amostras independentes ou `paired=TRUE` para o teste Wilcoxon para amostras dependentes (cf. próxima seção);
- `exact = T`, se você quiser calcular um teste exato, ou `exact = FALSE` se você não quiser (se você não alterar a definição padrão NULL do exato e seus dados tiverem menos de 50 pontos de dados e nenhum laço, um valor- p exato é calculado automaticamente);
- `correct = T` para uma correção de continuidade (o padrão) e `correct = FALSE` para nenhuma;
- `conf.level`: um valor entre 0 e 1 especificando o tamanho do intervalo de confiança; o padrão é 0.95.

A versão padrão a ser usada aqui é a seguinte:

Código em R

```
> wilcox.test(DICE~PROCESS, paired=FALSE, correct=FALSE)¶
Wilcoxon rank sum test
data: DICE by PROCESS
W = 2416, p-value = 9.072e-16
alternative hypothesis: true location shift is not equal to 0
```

Você obtém um valor- U (aqui referido como W) e um valor- p ; o W não é o mínimo de U_1 e U_2 , mas o máximo aqui. O valor que você obtém depende de qual vetor ou o nível de fator vem em primeiro lugar no alfabeto. O valor- p aqui é um pouco diferente do seu uma vez que o R usa um algoritmo ligeiramente diferente. Agora você pode resumir: "De acordo com um teste- U , o coeficiente de Dice mediano das palavras fonte de mesclagens (0,23, $IQR = 0,0675$) e a mediana dos coeficientes de Dice para truncamentos complexos (0,1195, $IQR = 0,0675$) são muito significativamente diferentes: $U = 84$ (ou $W = 2,416$), $p_{bicaudal} < 0,0001$. Os criadores de mesclagens parecem estar mais preocupados com a seleção de palavras fonte que são semelhantes entre si do que os criadores de truncamentos complexos."

Recomendação para estudo/leitura

Dalgaard (2002, p. 89), Crawley (2007, p. 297), Baayen (2008, Seção 4.3.1), Sheskin (2011, Teste 12)

4.3.2.4 Uma variável dependente (ordinal) e uma variável independente (nominal) (amostras dependentes)

Assim como o teste- U , o teste nesta seção tem duas aplicações principais. Em primeiro lugar, você realmente pode ter duas amostras dependentes de dados ordinais, como no caso em que você pede

que um grupo de indivíduos realize duas tarefas de classificação para testar se as classificações feitas primeiramente pelos sujeitos da pesquisa se diferem das segundas classificações. Em segundo lugar, a aplicação provavelmente mais frequente surge quando você tem duas amostras dependentes de dados de mescla de proporção, mas não pode fazer o teste t para amostras dependentes porque as suposições de distribuição desse teste não são atendidas. Vamos discutir um exemplo deste último tipo nesta seção.

Em uma réplica de [Bencini e Goldberg \(2000\)](#), [Gries e Wulff \(2005\)](#) estudaram a questão sobre quais verbos ou estruturas frasais são mais relevantes para a categorização de sentenças por alemães alunos de inglês como língua estrangeira. Eles combinaram quatro construções sintáticas e quatro verbos para obter 16 frases, de forma que cada verbo fosse usado em cada construção. Cada frase foi impressa em um cartão, e 20 alemães aprendizes avançados de inglês receberam os cartões para classificá-los em quatro pilhas de quatro cartões cada. A pergunta era se os agrupamentos feitos pelos sujeitos seriam baseados nos verbos ou as construções. Para determinar as preferências de classificação, as quatro pilhas de cada sujeito foram analisadas em relação a quantos cartões teriam que ser minimamente movidos para se criar ou quatro classificações completamente baseadas nos verbos ou quatro classificações completamente baseadas nas construções. A investigação sobre essa questão envolve:

- uma variável dependente de escala de proporção, a dizer SHIFT, o número de vezes que um cartão teve que ser movido de uma pilha para a outra para criar classificações perfeitamente limpas (ou apenas baseadas nos verbos ou apenas baseadas nas construções, e estamos interessados na média desses números);
- uma variável nominal independente, a dizer CRITERION: CONSTRUCTION versus CRITERION: VERB;
- amostras dependentes uma vez que cada sujeito “gerou” dois números de mudanças, um para criar a classificação baseada nos verbos, um para criar a classificação baseada nas construções.

Para testar alguns desses resultados para significância, você deve primeiro considerar um teste- t para amostras dependentes uma vez que você tem duas amostras de valores de escala de proporção. Como usual, você começa formulando as hipóteses relevantes:

H_0 : A média das diferenças pareadas entre os números de rearranjos para a obtenção de pilhas unicamente baseadas em verbos e os números de rearranjos para a obtenção de pilhas unicamente baseadas em construções é 0; $\text{média}_{\text{diferenças pareadas}} = 0$.

H_1 : A média das diferenças pareadas entre os números de rearranjos para a obtenção de pilhas unicamente baseadas em verbos e os números de rearranjos para a obtenção de pilhas unicamente baseadas em construções não é 0; $\text{média}_{\text{diferenças pareadas}} \neq 0$.

Depois, você carrega os dados que [Gries e Wulff \(2005\)](#) obtiveram em seu experimento a partir do arquivo `<_inputfiles/04-3-2-4_sortingstyles.csv>`:

Código em R

```
> SortingStyles=read.delim(file.choose())  
> head(SortingStyles, 3); attach(SortingStyles)
```

Como sempre, você calcula as médias e os desvios padrão e gera um gráfico dos resultados.

Código em R

```
> tapply(SHIFTS, CRITERION, mean)¶
Construction      Verb
3.45             8.85
> tapply(SHIFTS, CRITERION, sd)¶
Construction      Verb
4.346505         4.107439
> differences=SHIFTS[CRITERION=="Construction"]-SHIFTS[CRITERION!="Construction"]¶
> stripchart(differences, method="stack", xlim=c(-12, 12), xlab="Differences: -
>construction minus ->verb"); abline(v=0, lty=2, col="grey")¶
```

Importante: uma vez que as duas amostras são dependentes, nós estamos gerando gráficos das diferenças, assim como na seção 4.3.2.2 acima. Você, então, testa a pressuposição do teste-*t* para amostras dependentes, a normalidade das diferenças pareadas. Com base na Figura 56, elas obviamente não são normais:

Código em R

```
> shapiro.test(differences)¶
Shapiro-Wilk normality test
data: differences
W = 0.7825, p-value = 0.0004797
```

Differences: ->construction minus ->verb

Figura 56 – *Stripchart* da diferenças dos *shifts*

Você não pode usar o teste-*t*. ao invés disso, você calcula um teste para duas amostras dependentes de variáveis ordinais, o teste de Wilcoxon.

Procedimentos

- Formulação de hipóteses
- Cálculo das estatísticas descritivas e análise visual do gráfico
- Teste das pressuposições do teste:
 - os pares de valores são independentes uns dos outros
 - as populações das quais as amostras cujas tendências centrais são testadas são distribuídas de formas idênticas
- Cálculo da estatística T do teste e da probabilidade de erro p

Como primeiro passo, você ajusta as suas hipóteses ao nível ordinal de medição, e depois você calcula as medianas e seus intervalos interquartílicos:

$$H_0: \text{média}_{\text{diferença pareada}} = 0$$

$$H_1: \text{média}_{\text{diferença pareada}} \neq 0$$

Código em R

```
> tapply(SHIFTS, CRITERION, median)¶
Construction Verb
1 11
> tapply(SHIFTS, CRITERION, IQR)¶
Construction Verb
6.25 6.25
```

Os pressupostos parecem ser satisfeitos porque os pares de valores são independentes uns dos outros (uma vez que a classificação de qualquer participante da pesquisa não afeta nenhuma classificação de outro participante da pesquisa) e, de certa forma informalmente, há pouca razão para supor que as populações são distribuídas de formas diferentes especialmente porque a maioria dos valores para se alcançar uma classificação unicamente baseada em verbos são exatamente o inverso dos valores para se obter uma classificação unicamente baseada em construções. Assim, você calcula o teste de Wilcoxon; por razões de espaço nós consideramos apenas a variante padrão. Primeiro, você transforma o vetor de diferenças pareadas, que você já calculou para o teste Shapiro-Wilk, em posições:

Código em R

```
> ranks=rank(abs(differences))¶
```

Em segundo lugar, todas as posições cuja diferença foi negativa são somadas a um valor $T-$, e todas as posições cuja diferença foi positiva são somadas a $T+$; o menor dos dois valores é o teste da estatística T necessário:¹².

¹² A forma de cálculo discutida aqui é a descrita em [Bortz \(2005\)](#). Ela desconsidera os laços e casos em que as diferenças são zero; cf. [SHESKIN, 2011](#), p. 812

Código em R

```
> T.minus=sum(ranks[differences<0])¶ > T.plus=sum(ranks[differences>0])¶ >
T=min(T.minus, T.plus)¶
```

Este valor- T de 41,5 pode ser olhado em uma tabela T (Tabela 37), mas note que aqui, para um resultado significativo, o teste estatístico observado deve ser menor do que o tabulado.

Tabela 38 – Valores- T críticos para $p_{bicaudal} = 0,05, 0,01, \text{ e } 0,001$ para $14 \leq df \leq 16$

	p = 0,05	p = 0,01	p = 0,001
n = 19	46	32	18
n = 20	52	37	21
n = 21	58	42	25

O valor- T observado de 41,5 é menor do que o tabulado para $n = 20$ e $p = 0,05$ (mas maior do que o tabulado para $n = 20$ e $p = 0,01$): o resultado é significativo.

Vamos agora fazer esse teste como R: Você já conhece a função para o teste Wilcoxon e, portanto, não precisamos discuti-la novamente em detalhe. A diferença relevante é que agora você instrui o R para tratar as amostras como dependentes/pareadas. Como quase sempre, você pode usar a fórmula ou a função baseada em vetores.

Código em R

```
> wilcox.test(SHIFTS CRITERION, paired=TRUE, exact=FALSE,
correct=FALSE)¶
Wilcoxon signed rank test
data: SHIFTS by CRITERION
V = 36.5, p-value = 0.01527
alternative hypothesis: true location shift is not equal to 0
```

O R calcula a estatística do teste de forma diferente, mas chega ao mesmo tipo de conclusão: o resultado é significativo, mas não muito significativo. Resumindo: “De modo geral, os 20 indivíduos apresentaram uma forte preferência por um estilo de classificação baseado nas construções: o número médio de rearranjos de cartões para se chegar a uma classificação unicamente baseada nas construções foi 1, enquanto que o número médio de rearranjos de cartões para se chegar a uma classificação com base unicamente nos verbos foi 11 (ambos $IQRs = 6,25$). De acordo com um teste de Wilcoxon, essa diferença é significativa: $V = 36,5$, $p_{bicaudal} = 0,0153$. Neste experimento, os padrões sintáticos foram uma característica mais relevante do que os verbos (quando se trata do que desencadeou as preferências de classificação).”

Recomendação para estudo/leitura

- [Dalgaard \(2002, p. 92\)](#), [Sheskin \(2011, Teste 18\)](#)

4.4 Coeficientes de correlação e regressão linear

Nesta seção, nós discutiremos os testes de significância para os coeficientes de correlação discutidos na Seção 3.2.3.

4.4.1 A significância da correlação produto-momento

Enquanto o cálculo manual da correlação produto-momento acima foi um pouco complexo, o seu teste de significância não é. Ele envolve estes passos:

Procedimentos

- Formulação das hipóteses
- Cálculo da correlação observada; exame de um gráfico
- Teste das pressuposições do teste: a população da qual a amostra foi retirada segue uma distribuição normal bivariada. Uma vez que pode ser difícil testar esse critério (cf. [BORTZ, 2005](#), p. 213), nós simplesmente exigimos que ambas as amostras sejam distribuídas de forma normal
- Cálculo da estatística do teste- t , graus de liberdade gl (df), e probabilidade de erro p

Voltemos ao exemplo da Seção 3.2.3, quando você calculou um coeficiente de correlação de 0,9337 para a correlação das extensões de 20 palavras e seus tempos de reação. Você formula as hipóteses e nós assumimos por enquanto que a sua hipótese alternativa é não-direcional.

H_0 : A extensão de uma palavra em letras não se correlaciona com o tempo de reação a essa palavra em uma tarefa de decisão lexical; $r = 0$.

H_1 : A extensão de uma palavra em letras se correlaciona com o tempo de reação a essa palavra em uma tarefa de decisão lexical; $r \neq 0$.

Você carrega os dados já conhecidos de `<_inputfiles/03-2-3_reactiontimes.csv>`:

Código em R

```
> ReactTime=read.table(choose.files())
> str(ReactTime); attach(ReactTime)
```

Uma vez que nós já geramos um gráfico de dispersão acima (ver Figura 35 e Figura 36), nós não fazemos isso novamente. Nós, contudo, temos que testar o pressuposto de normalidade de ambos os vetores. Você pode tanto proceder de uma forma gradual e executar `shapiro.test(LENGTH)` e `shapiro.test(MS_LEARNER)`, ou usar uma alternativa mais curta:

Código em R

```
> apply(ReactTime[,2:3], 2, shapiro.test)
$LENGTH
Shapiro-Wilk normality test
data: newX[, i]
W = 0.9748, p-value = 0.8502
$MS_LEARNER
Shapiro-Wilk normality test
data: newX[, i]
W = 0.9577, p-value = 0.4991
```

Essa linha de código significa “pegue os dados mencionados no primeiro argumento de `apply` (a segunda e a terceira coluna da estrutura de dados `ReactTime`), analise-os coluna por coluna” (o 2 no segundo parâmetro – um 1 significaria “olhe para eles linha a linha”; lembre-se dessa notação de `prop.table` na Seção 3.2.1), e aplique a função `shapiro.test` para cada coluna. Claramente, ambas as variáveis não se diferem significativamente de uma distribuição normal.

Para calcular a estatística t do teste, você insere o coeficiente de correlação r e o número de pares de valores correlacionados n na fórmula em (55):

$$(49) \quad t = \left| \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \right|$$

Código em R

```
> r=cor(LENGTH, MS_LEARNER, method="pearson")
> numerator=r*sqrt(length(LENGTH)-2)
> denominator=sqrt(1-r^2)
> t=numerator/denominator
```

Esse valor- t , 11,06507, tem $df = n - 2 = 18$ graus de liberdade.

Código em R

```
> df=length(LENGTH)-2
```

Assim como com os testes- t de antes, agora você pode olhar este valor- t em uma tabela- t , ou você pode calcular um valor crítico: se o valor- t observado for mais alto do que aquele tabulado/crítico, então r é significativamente diferente de 0. Uma vez que o seu valor- t é muito maior até mesmo do que aquele para $p = 0,001$, a correlação é altamente significativa.

Código em R

```
> qt(c(0.025, 0.975), 18, lower.tail=F)
[1] 2.100922 -2.100922
```

Tabela 39 – Valores- t críticos para $p_{bicaudal} = 0,05, 0,01, \text{ e } 0,001$ para $17 \leq df \leq 19$

	p = 0,05	p = 0,01	p = 0,001
df = 17	2,1098	2,8982	3,9561
df = 18	2,1009	2,8784	3,9216
df = 19	2,093	2,8609	3,8834

O valor- p exato pode ser calculado como mostrado a seguir, e não se esqueça de dobrar o valor- p .

Código em R

```
> 2*pt(t, 18, lower.tail=FALSE)
[1] 1.841060e-09
```

Esse valor- p é obviamente muito menor do que 0,001. No entanto, você já vai suspeitar que há uma maneira mais fácil de fazer tudo isso. Em vez da função `cor`, a qual nós temos utilizado na Seção 3.2.3 acima, você simplesmente usa `cor.test` com os dois vetores em cuja correlação você está interessado (e, se você tem uma hipótese alternativa direcional, você especifica se você espera que a correlação seja menor do que 0 (isto é, negativa) ou maior que 0 (isto é, positiva), utilizando `alternative=...`):

Código em R

```
> cor.test(LENGTH, MS_LEARNER, method="pearson")
Pearson's product-moment correlation
data: LENGTH and MS_LEARNER
t = 11.0651, df = 18, p-value = 1.841e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.8370608 0.9738525
sample estimates:
cor
0.9337171
```

Aqui estão os resultados (editados) da regressão linear correspondente:

Código em R

```
> model<-lm(MS_LEARNER~LENGTH)
> summary(model)
Call:
lm(formula = MS_LEARNER ~ LENGTH)
Residuals:
Min 1Q Median 3Q Max
-22.1368 -7.8109 0.8413 7.9499 18.9501
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 93.6149 9.9169 9.44 2.15e-08 ***
LENGTH 10.3044 0.9313 11.06 1.84e-09 ***
-- [...]
Multiple R-Squared: 0.8718, Adjusted R-squared: 0.8647
F-statistic: 122.4 on 1 and 18 DF, p-value: 1.841e-09
```

Vamos começar pela parte inferior: a última linha contém informações que já conhecemos. O valor- F é o nosso valor- t ao quadrado; nós encontramos os 18 graus de liberdade e o valor- p que calculamos. Na linha logo acima, você encontra o coeficiente de determinação que você conhece, além de uma versão ajustada sobre a qual só vamos falar mais tarde (ver Seção 5.2). Nós ignoramos a linha editada sobre o erro padrão residual e a legenda para os valores- p . A tabela acima mostra a interceptação e a inclinação que nós calculamos na Seção 3.2.3 (na coluna com o título “Estimativa”), seus erros padrão, os valores- t – você reconhece o valor- t acima? – e os valores- p . O valor- p para LENGTH diz se a inclinação da linha de regressão é significativamente diferente de 0; o valor- p para a interceptação diz se a interceptação de 93,6149 é significativamente diferente de 0. Nós pulamos a informação sobre os resíduos porque nós discutimos acima como você mesmo pode investigá-los (com `residuals(model)`).

Há uma última, mas extremamente útil, coisa a ser discutida. Lembre-se que acima usamos a função `predict` para obter os tempos de reação previstos para cada extensão de palavra observada, mas também fizemos previsões sobre tempos de reação para extensões de palavras não observadas. A função `predict`, porém, pode retornar mais do que isso: ela pode também retornar intervalos de confiança para as previsões, o que também permite gerar um gráfico da linha de regressão com seu intervalo de confiança. Uma vez que faremos isso com frequência no Capítulo 5, vamos examinar um exemplo aqui, o qual envolverá três etapas.

O primeiro passo repete o que fizemos acima: geramos uma estrutura de dados `preds.hyp` que contém um intervalo de valores que cobrem as extensões de palavras observadas e que passaremos para `predict`, e fazemos isso como na Seção 3.2.3 com `Expand.grid()`. Eu chamo a estrutura de dados de

`preds.hyp` para indicar que essas são previsões do modelo não para as extensões realmente observadas, mas para uma gama de valores hipotéticos. Note novamente que a coluna em `preds.hyp` tem o mesmo nome que a variável independente em `model`.

Código em R

```
preds.hyp=expand.grid(LENGTH=min(LENGTH):max(LENGTH))
```

A segunda etapa também é similar à Seção 3.2.3 acima, mas com duas pequenas mudanças. Nós não apenas usamos `predict` para gerar as previsões a partir de `model` para essa estrutura de dados, mas (i) também deixamos o R calcular os intervalos de confiança para todas as previsões e (ii) fazemos as previsões e os intervalos de confiança nas colunas de 2 a 4 em `preds.hyp`:

Código em R

```
> preds.hyp[,c("PREDICTIONS", "LOWER", "UPPER")]=predict(model, new-  
data=preds.hyp,interval="confidence")
```

Se você olhar para a estrutura de dados `preds.hyp` agora, você verá que agora temos um resultado muito bom: a variável independente está na coluna `preds.hyp$ LENGTH`, a variável dependente prevista está na coluna `preds.hyp$PREDICTIONS`, e os intervalos de confiança inferior e superior para cada previsão estão nas colunas `preds.hyp$LOWER` e `preds.hyp$UPPER`, respectivamente.

O terceiro passo agora envolve a geração de um bom gráfico. O seguinte código reúne muitas coisas e introduz a função `matlines`:

Código em R

```
> plot(MS_LEARNER~LENGTH, xlab="Word length in letters",ylab="Reaction time of  
learners in ms", pch=16,  
col=rgb(0, 0, 0, 70, maxColorValue=255)); grid()  
> matlines(preds.hyp[,1], preds.hyp[,2:4], lwd=c(2, 1,1), lty=c(1, 2, 2),  
col=c("black", "blue", "blue"))
```

A primeira linha apenas gera um `scatterplot` regular - a única coisa nova é o uso da função `rgb` para usar um tom de cinza semi-transparente para evitar a perda de informações por sobreposição. A segunda linha usa `matlines`: o primeiro argumento é a primeira coluna de `preds.hyp` e fornece os valores `x` para as linhas a serem geradas no gráfico. O segundo argumento corresponde às colunas 2 a 4 do `preds.hyp` e fornece três conjuntos diferentes de valores `y` para serem gerados no gráfico com linhas separadas: primeiro, os valores previstos (= linha de regressão), segundo e terceiro, os limites inferior e superior dos intervalos de confiança. Os argumentos `lwd` (largura da linha), `lty` (tipo de linha) e `col` (cor) descrevem o aspecto das linhas, na ordem em que aparecem em `preds.hyp`. O resultado que você vê quando você executa o código é um `scatterplot` com uma linha de regressão e sua faixa de confiança, e podemos ver novamente porque a correlação é tão alta: não só porque a linha de regressão é um bom resumo dos dados, mas também porque a banda de confiança é bastante estreita em torno da linha e muitos pontos estão bem nela ou muito perto dela.

A descrição acima foi muito detalhada, mas uma vez que vamos usar esse tipo de procedimento muitas vezes no capítulo 5, foi um tempo bem gasto. Para resumir: "As extensões das palavras em letras e os tempos de reação no experimento se correlacionam altamente positivamente uns com os outros: $r = 0,9337$; $R^2_{ajustado} = 0,8647$. Essa correlação é altamente significativa: $t = 11,07$; $gl = 18$; $p < 0,001$. A regressão linear mostra que cada letra adicional aumenta o tempo de reação em aproximadamente 10,3 ms."

Na Seção 5.2, nós tratamos de extensões de regressão linear para casos nos quais incluímos mais de uma variável independente, e nós também discutiremos testes mais completos dos pressupostos da regressão (usando `plot(model)`).

4.4.2 A significância do Tau de Kendall

Se você precisar de um valor- p para o τ de Kendall, siga os seguintes passos:

Procedimentos

- Formular as hipóteses
- Calcular a correlação observada; examinar um gráfico
- Testar o(s) pressuposto(s) do teste: os dados são pelo menos ordinais
- Calcular a estatística do teste z e a probabilidade de erro p

Mais uma vez, nós simplesmente usamos o exemplo da Seção 3.2.3 acima (embora saibamos que podemos na verdade usar a correlação produto-momento; nós usamos esse exemplo novamente para simplificar). Como formular as hipóteses já deve estar óbvio neste ponto:

H_0 : A extensão de uma palavra em números de letras não se correlaciona com o tempo de reação à palavra em uma tarefa de decisão lexical; $\tau = 0$.

H_1 : A extensão de uma palavra em números de letras se correlaciona com o tempo de reação à palavra em uma tarefa de decisão lexical; $\tau \neq 0$.

Em relação aos pressupostos: nós já sabemos que os dados são ordinais – afinal, nós sabemos que eles são até intervalares/de escala de proporção. Você carrega os dados novamente a partir de `<_inputfiles/03-2-3_reactiontimes.csv>` e calcula o τ de Kendall:

Código em R

```
> ReactTime=read.table(choose.files())  
> str(ReactTime);attach(ReactTime)  
> tau=cor(LENGTH, ·MS_LEARNER, ·method="kendall")
```

Para testar a significância de τ de Kendall, você deve calcular um escore- z do tipo com o qual você já deve estar familiarizado. Você insere τ e o número de pares de valores n na fórmula em (56).

$$(50) \quad z = |\tau| \div \sqrt{\frac{2(2n+5)}{9n(n-1)}}$$

No R:

Código em R

```
> numerator.root=2*(2*length(LENGTH)+5)  
> denominator.root=9*length(LENGTH)*(length(LENGTH)-1)  
> z=bs(tau)/sqrt(numerator.root/denominator.root)  
[1] ·5.048596
```

Esse valor pode depois ser procurado em uma tabelaz como a Tabela 36, ou você mesmo gera esses valores. O escore- z para um teste bicaudal significativo deve cortar de pelo menos 2,5% da área sob a distribuição normal padrão:

Código em R

```
> qnorm(c(0.0005, 0.005, 0.025, 0.975, 0.995, 0.9995), lower.tail=TRUE)
[1] -3.290527 -2.575829 -1.959964 1.959964 2.575829 3.290527
```

Para um resultado ser significativo, o escore- z deve ser maior que 1,96. Uma vez que o escore- z observado é até maior que 5, esse resultado é altamente significativo:

Código em R

```
> 2*pnorm(z, lower.tail=F)
[1] 4.450685e-07
```

A função para obter esse resultado de forma muito mais rápida é novamente `cor.test`. Uma vez que o R usa um método de cálculo um pouco diferente, você obtém um escore- z e um valor- p um pouco diferentes, mas os resultados são, para efeitos práticos, idênticos.

Código em R

```
> cor.test(LENGTH, MS_LEARNER, method="kendall")
Kendall's rank correlation tau
data: LENGTH and MS_LEARNER
z = 4.8836, p-value = 1.042e-06
alternative hypothesis: true tau is not equal to 0
sample estimates:
tau
0.8189904
```

(O alerta se refere a laços tais como o fato de valor do comprimento 11 ocorrer mais de uma vez). Para resumir: “As extensões das palavras em termos de números de letras e os tempos de reação no experimento se correlacionam altamente de forma positiva: $\tau = 0,819$, $z = 5,05$; $p < 0,001$.”

4.5 Correlação e causalidade

Especialmente na área de correlações, mas também de forma mais geral, você precisa manter em mente algumas coisas mesmo se a hipótese nula for rejeitada. Primeiro, pode-se ouvir frequentemente uma pessoa A fazendo uma afirmação sobre uma correlação (talvez até uma significativa) dizendo “Quanto mais X, mais Y”, e depois ouvir uma pessoa B contestando tal afirmação com base em uma exceção que conhece. O argumento é falho. A exceção citada por B apenas invalidaria a afirmação de A se A considerasse que a correlação fosse perfeita ($r = 1$ or $r = -1$) – mas se A não quis dizer isso (e A nunca quer!), então pode haver correlação forte e significativa embora haja (pelo menos) uma exceção. A exceção ou exceções são a razão pela qual a correlação não é 1 ou -1, mas “apenas”, digamos, 0,9337. Em segundo lugar, uma correlação desse tipo não implica necessariamente causalidade. Como é dito às vezes, a correlação entre X e Y é uma *condição necessária* para uma relação causal entre X e Y, mas não é *suficiente*, como você pode observar em vários exemplos:

- Existe uma correlação positiva entre o número de bombeiros tentando apagar um incêndio e a quantidade de danos causados no local onde o incêndio foi combatido and the amount of damage that is caused at the site. É claro que isso não significa que os bombeiros chegam no local e destroem tudo que podem – a correlação é resultado de uma terceira variável, o tamanho do incêndio: quanto maior o incêndio, mais bombeiros são chamados para ajudar a combatê-lo e mais danos são causados pelo incêndio.

- Existe uma correlação negativa entre a quantidade de cabelos que os homens têm e suas rendas que é infelizmente causada apenas por uma terceira variável: as idades dos homens.
- Existe uma correlação positiva tal que quanto mais um viciado em drogas tendia a ir ao terapeuta para se livrar de seu vício, mais probabilidade ele tinha de morrer. Isso não ocorre porque a terapia conduz a pessoa à morte - a variável por trás desse contexto correlacionada a ambos é a gravidade do vício: quanto mais viciados os viciados eram, mais propensos eles eram de frequentar a terapia, mas também mais propensos eles eram a morrer.

Portanto, cuidado para não tirar conclusões precipitadas...

Agora você deve fazer os exercícios do Capítulo 4...

Recomendação para estudo/leitura

- Recomendação de estudo as funções κ e 1κ (da biblioteca psy) para calcular o coeficiente kappa e testar quão bem dois ou mais participantes da pesquisa concordam em seus julgamentos de estímulos
- a função cronbach (da biblioteca psy) para calcular o alpha de Cronbach e testar quão consistentemente diversas variáveis medem um construto que as variáveis devem refletir
- [Crawley \(2007, Cap. 10\)](#), [Baayen \(2008, Seção 4.3.2\)](#), [Johnson \(2008, Seção 2.4\)](#), [Sheskin \(2011, Testes 28, 30, 31 e 32\)](#)
- a função hints (da biblioteca hints) para obter ideais sobre o que fazer em seguida com um objeto específico

Seleção de Métodos Multifatoriais e Multivariados

Todos os modelos estão errados, mas alguns são úteis.

George E.P. Box

Até agora temos apenas nos interessado em métodos monofatoriais, isto é, métodos nos quais investigamos como uma variável independente está relacionada ao comportamento de uma variável dependente. Em muitos casos, proceder dessa forma é o início do estudo quantitativo empírico de um fenômeno. No entanto, essa visão dos fenômenos é geralmente uma simplificação: vivemos em um mundo multifatorial no qual provavelmente nenhum fenômeno é realmente monofatorial - provavelmente quase tudo está correlacionado com várias coisas ao mesmo tempo. Isso é especialmente verdadeiro para a linguagem, um dos fenômenos mais complexos resultantes da evolução humana. Nesta seção, discutiremos, portanto, várias técnicas multifatoriais, as quais podem lidar com esse tipo de complexidade de forma melhor do que os métodos monofatoriais discutidos até agora. Você deve saber, no entanto, que o método de cada seção abaixo poderia facilmente ser o assunto de cursos com duração de vários trimestres ou semestres, razão pela qual eu, infelizmente, não posso discutir todos os aspectos ou técnicas dos métodos, e terei que sugerir várias referências e recomendações para um estudo mais aprofundado. Além disso, dada a complexidade desses métodos, não haverá discussão sobre como calculá-los manualmente.

Antes de podermos começar a discutir métodos multifatoriais, no entanto, há muito o que discutir. Em um nível muito abstrato, esta discussão envolve as noções de *interação* e *modelo(seleção)* e será o tema da Seção 5.1. Contudo, como você verá em breve, essas noções levarão rapidamente a uma variedade de conceitos interrelacionados e, em última instância, a estratégias analíticas importantes para as seções mais práticas subsequentes.

5.1 As noções de interação e modelo(seleção)

5.1.1 Interações

Como mencionado no início do capítulo anterior, métodos multifatoriais envolvem uma variável dependente e duas ou mais variáveis independentes, não apenas uma como em todo o Capítulo 4. Esta presença de mais de uma variável independente possibilita resultados potencialmente interessantes, mas também levanta a questão de como as duas ou mais variáveis independentes se relacionam em conjunto à variável dependente.

Existem basicamente duas maneiras diferentes pelas quais diversas variáveis independentes e dependentes podem ser relacionadas, as quais vamos explorar com base no exemplo envolvendo as extensões

dos constituintes discutidas no Capítulo 1. Vamos supor novamente que você quis estudar se as extensões de constituintes - capturados na variável dependente LENGTH - são correlacionadas com duas variáveis independentes, a variável GRMRELATION (com os dois níveis SUBJECT e OBJECT e a variável CLAUSETYPE (com os dois níveis MAIN e SUBORDINATE). Vamos supor ainda que você fez um pequeno estudo piloto no qual você investigou 120 constituintes que são distribuídos como mostra a Tabela 39.

Tabela 40 – Dados fictícios contendo a média do tamanho de sujeitos e objetos

	GRMRELATION: SUBJ	GRMRELATION: OBJ	Totais
CLAUSETYPE: MAIN	30	30	60
CLAUSETYPE: SUBORD	30	30	60
Totais	60	60	120

Para cada um dos 120 constituintes, você mediu o seu tamanho – em sílabas – e calculou a média para cada uma das quatro combinações – sujeitos em orações principais, sujeitos em orações subordinadas, objetos em orações principais e sujeitos em orações subordinadas. Os seguintes resultados foram obtidos:

- em média, sujeitos são mais curtos que objetos diretos (em orações principais e subordinadas);
- em média, os sujeitos e objetos em orações principais são mais curtos do que sujeitos e objetos em orações subordinadas.

O interessante é que esses resultados monofatoriais - lembre-se que vimos na Seção 3.2.2.2 que esses resultados são muitas vezes chamados de efeitos principais - podem vir em diferentes formas. Por um lado, os efeitos das duas variáveis independentes podem ser acumulativos. Isso significa que a combinação das duas variáveis tem o efeito que você esperaria de cada efeito principal. Uma vez que os sujeitos são curtos (mais curtos), assim como são os constituintes nas orações principais, a aditividade prevê que os sujeitos das orações principais devem ser os constituintes mais curtos, e que os objetos das orações subordinadas devem ser os mais longos. Esse resultado, que seria o previsto pela H_0 , é representado na Figura 57: os pontos pretos e cinza indicam as extensões médias dos objetos e dos sujeitos, respectivamente, nas duas relações gramaticais, mas também a média entre ambos, e o "m" e o "s" representam as médias dos constituintes das orações principais e subordinadas nas duas relações gramaticais.

Esse resultado é, de fato, muito acumulativo porque as duas linhas são perfeitamente paralelas.

- a diferença entre a extensão do sujeito da oração principal e da extensão do objeto da oração principal é de -2.5 sílabas;
- a diferença entre a extensão do sujeito da oração principal e da extensão do sujeito da oração subordinada é de -2 sílabas;
- a extensão média do sujeito da oração principal é de 2 sílabas,

então você pode perfeitamente prever a extensão média do objeto da oração subordinada: $2 + 2,5 + 2 = 6,5$.

No entanto, com exatamente os mesmos tipos de efeitos principais, também é possível que as duas variáveis independentes interajam entre si. Duas ou mais variáveis interagem se seu efeito conjunto sobre a variável dependente não for previsível a partir de seus efeitos individuais na mesma variável dependente. Um cenário desse tipo é representado na Figura 58. Considere primeiro o painel esquerdo. Você pode ver que ainda existem os mesmos tipos de efeito principal de GRMRELATION (os sujeitos são novamente mais curtos do que objetos) e CLAUSETYPE (constituintes das orações principais são novamente mais curtos do que os constituintes das orações subordinadas), mas agora as linhas não são mais paralelas, mas se cruzam.

O que isso significa? Isso significa que, se eu lhe disser que

Figura 57 – Gráfico de Interação para $LENGTH \sim GRMRELATION * CLAUSETYPE 1$ Figura 58 – Gráfico de Interação para $LENGTH \sim GRMRELATION * CLAUSETYPE 2$

- a diferença entre a extensão do sujeito da oração principal e a extensão do objeto da oração principal é de 3 sílabas;
- a diferença entre a extensão do sujeito da oração principal e a extensão do sujeito da oração subordinada é de 4 sílabas;
- a extensão média do sujeito da oração principal é de 2 sílabas,

então você definitivamente não pode prever a extensão média do objeto da oração subordinada: você poderia prever $2 + 3 + 4 = 9$ sílabas (como indicado no painel direito com a linha tracejada terminando em um círculo, que é paralela ao cinza), enquanto que a extensão média real do objeto da oração subordinada nos dados é de 4 sílabas. Isso é uma interação: você não pode prever a extensão média do objeto da oração subordinada usando os dois efeitos principais, mas precisa de um termo de interação adicional que “corrige” a sua previsão de 9 para o resultado real de 4. Um teste desse termo de interação testaria se esse termo é significativamente diferente de zero ou não.

Ainda existe outro tipo de interação que é apresentado na Figura 59. Novamente, temos os já conhecidos efeitos principais, e mesmo que as linhas não se cruzem, essa é ainda uma interação pela mesma razão que acima. Se eu lhe disser que

- a diferença entre a extensão do sujeito da oração principal e a extensão do objeto da oração principal é de 2 sílabas;
- a diferença entre a extensão do sujeito da oração principal e a extensão do sujeito da oração subordinada é de 2 sílabas;
- a extensão média do sujeito da oração principal é de 2 sílabas,

então novamente você não pode prever a extensão média do objeto da oração subordinada: você poderia prever $2 + 2 + 2 = 6$ sílabas (como indicado novamente no painel à direita com a linha tracejada, que é paralela à cinza), enquanto que o comprimento real do objeto da oração subordinada média nos dados é de 8 sílabas.

Novamente, há uma interação: você não pode prever a extensão média do objeto da oração subordinada usando os dois efeitos principais, mas precisa de um termo de interação adicional que corrige a sua previsão 6 para o resultado real de 8, o que pode ser novamente um efeito de interação significativa.

Antes de prosseguirmos, deixe-me dar brevemente um segundo exemplo de uma interação, com o qual você já está realmente familiarizado, mesmo que você não tenha pensado sobre ele dessa forma. O exemplo acima envolveu médias, o exemplo de agora envolve frequências. Imagine que você faça um estudo de corpus de 60 genitivos do inglês com a preposição *of* e de 80 casos genitivos do inglês com o *s* no qual você tenta determinar se a escolha do tipo de genitivo está correlacionada com a animacidade do SN possuidor (por exemplo, John na frase “John’s car”). Imagine agora que você apresentou seus resultados a um colega em uma tabela de visão geral, mas que você omitiu algumas informações principais da tabela, como na Tabela 40. Se você agora pedisse ao seu colega para preencher a Tabela 40 sem assumir que há algo em particular acontecendo nos dados, seu colega deve - talvez implicitamente - assumir a H_0 e adotar a lógica do teste de qui-quadrado e calcular as frequências esperadas de H_0 , como na Tabela 41.

Tabela 41 – Um conjunto de dados fictícios de escolhas de genitivos (apenas os totais)

	Possuidor animado	Possuidor inanimado	Totais
<i>of-genitive</i>			60
<i>s-genitive</i>			80
Totais	70	70	140

Isso ocorre porque se seu é dito ao seu colega que ele não deve assumir nada de especial nos dados, é difícil ter algo para motivar qualquer situação diferente da Tabela 41. Sim, seu colega poderia criar algo como a Tabela 42 e dizer: “há sempre um pouco de variação aleatória”, mas ... como ele poderia encontrar

Figura 59 – Gráfico de Interação para $LENGTH \sim GRMRELATION * CLAUSETYPE$ 3

Tabela 42 – Um conjunto 1 de dados fictícios de escolhas de genitivos

	Possuidor animado	Possuidor inanimado	Totais
<i>of-genitive</i>	30	30	60
<i>s-genitive</i>	40	40	80
Totais	70	70	140

motivação para chegar à Tabela 43 sem assumir algo de especial nos dados? Esse “algo especial” seria uma interação.

Tabela 43 – Um conjunto 2 de dados fictícios de escolhas de genitivos

	Possuidor animado	Possuidor inanimado	Totais
<i>of-genitive</i>	33	27	60
<i>s-genitive</i>	37	43	80
Totais	70	70	140

Tabela 44 – Um conjunto 3 de dados fictícios de escolhas de genitivos

	Possuidor animado	Possuidor inanimado	Totais
<i>of-genitive</i>	10	50	60
<i>s-genitive</i>	60	20	80
Totais	70	70	140

Assim, o que no cenário do teste do qui-quadrado corresponde às frequências esperadas da H_0 são de fato as frequências que resultam de assumirmos um comportamento acumulativo das duas variáveis. E o que um teste de qui-quadrado faz é avaliar se os dados se desviam da distribuição assumindo que não há nenhuma interação de tal forma que o valor- p do teste qui-quadrado seja $< 0,05$, o que, por sua vez, significa que você rejeitará a H_0 e assumirá que há um interação.

Essa foi provavelmente uma caracterização minuciosamente detalhada, mas a noção de interação é muito importante (e muitas vezes mal compreendido e/ou subutilizado), e por isso é absolutamente crucial que você a entenda. Isso, porque a presença de uma interação significativa significa que você não pode considerar os efeitos principais das variáveis independentes da interação como verdadeiros! Na Figura 58, embora haja um efeito principal de objetos serem mais longos do que os sujeitos, a interação mostra que isso é apenas verdadeiro em orações principais, mas não em orações subordinadas. Essa propriedade de interações significativas - que elas qualificam os efeitos principais - é uma das razões mais importantes pelas quais sua inclusão em um modelo é muitas vezes essencial, um tópico que iremos abordar agora.

5.1.2 Seleção de modelo

A última seção terminou com uma frase usando a palavra *modelo*, uma palavra que você também foram encontrou quando discutimos modelos lineares e regressão. Eu usei essa palavra sem uma definição formal até agora, mas você provavelmente já tinha uma compreensão intuitiva do que eu estava falando. Agora, mais formalmente, quero definir um modelo como uma caracterização formal da relação entre *preditores* - variáveis independentes e suas interações - e uma ou mais variáveis dependentes. Essa 'caracterização' geralmente vem na forma de uma equação (de regressão) do tipo que você viu nas Seções 3.2.3 e 4.4, e também esquematicamente nas legendas da Figura 57, da Figura 58 e da 59, nas quais a finalidade da equação de regressão é quantificar a relação entre os preditores e as variáveis dependentes e gerar previsões para as variáveis dependentes. O desenvolvimento de um modelo apropriado, ou equação de regressão, é chamado de modelagem ou seleção de modelo, e diferentes tipos de modelagem são o assunto principal deste capítulo.

Uma palavra de cautela: este capítulo, por mais curto que seja, mostrará que, com dados multifatoriais, o tipo de receita pronta do Capítulo 4 não funcionará: a análise de dados multifatoriais requer muitas vezes deixar caminhos bem trilhados e distinções apreciadas (por exemplo, entre abordagens exploratórias e de teste de hipóteses). A análise de um conjunto de dados complexo é muito parecida com o trabalho de detetive ou com descascar uma cebola, onde a cada passo múltiplos caminhos são possíveis, e eu queria poder dizer que eu tenho todas as soluções para todos os conjuntos de dados que eu já explorei ... Ok, vamos começar!

5.1.3 Formulação do primeiro modelo

O primeiro passo na seleção do modelo pode parecer ser a formulação de um primeiro modelo, uma equação que tenta modelar a relação entre *preditores* e, por agora, uma variável dependente. No entanto, há uma variedade de ameaças à modelagem que precisam ser levada em consideração. Uma delas tem a ver com algo tão simples como reconhecer a natureza da variável dependente: ela é binária? Categórica? Numérica? Numérica, mas apenas cobrindo uma gama particular de valores discretos (por exemplo, 0 e números inteiros positivos como com frequências)? Ou apenas positiva, mas com um mínimo como com os tempos de reação?

Nós apenas falamos sobre modelagem em um contexto de modelagem linear (com a função *lm*), a qual é tipicamente usada quando a variável dependente é numérica e abrange uma grande variedade de valores. No entanto, uma vez que uma regressão linear praticamente sempre irá prever valores contínuos, ela não é realmente adequada para ser aplicada a variáveis dependentes binárias (embora isso ainda seja comum) ou categóricas. Além disso, uma vez que uma regressão linear praticamente sempre irá prever valores negativos, ela pode não ser adequada para prever frequências. Abaixo, discutiremos diferentes modelos para diferentes variáveis dependentes. Felizmente, boa parte da lógica dos modelos lineares, que você já conhece, pode ser aplicada à maioria desses casos.

Uma segunda ameaça está relacionada com a utilização de preditores no valor e na escala de informação mais úteis. Em relação ao primeiro aspecto, ainda há muito trabalho sendo feito nos quais preditores contínuos são fatorizados. Isso significa que, em vez de usar o preditor contínuo como ele é, os pesquisadores dividem-no em uma variável categórica com apenas um número pequeno de níveis

(talvez usando cut). Isso pode não apenas causar a perda de várias informações, especialmente se o corte não é feito após uma análise muito cuidadosa, mas também aumenta os graus de liberdade para a análise, potencialmente tornando mais difícil obter resultados significativos. Se possível, manter variáveis numéricas numéricas é provavelmente uma boa ideia.

Em relação ao segundo aspecto, é importante perceber, digamos, que nem todos os preditores numéricos devem ser inseridos no modelo tal como estão. Por exemplo, efeitos de frequência operam frequentemente numa escala logarítmica de forma que, mesmo que a palavra 1 seja dez vezes mais frequente que a palavra 2, o efeito da palavra 1 sobre a variável dependente, por exemplo, tempo de reação, poderia ser $\log(10)$ vezes mais forte. Assim, o que se deve talvez colocar na equação de regressão é $\log(\text{frequência})$ (e para interpretar os resultados mais facilmente, pode ser bom usar um logaritmo na base 2!)

Uma terceira ameaça diz respeito ao fato de que provavelmente a maioria das modelagens estatísticas na linguística corresponde a algum tipo (generalizado) de modelagem linear no qual o efeito de um preditor pode ser resumido com uma linha de regressão direta (em algum espaço numérico). No entanto, as relações entre preditores podem diferir em relação à forma como são melhor caracterizados, como os dois painéis na Figura 60 exemplificam. Não é uma boa ideia simplesmente forçar uma linha de regressão reta pelos dados no painel à direita ...

Figura 60 – Modelo envolvendo regressão linear e regressão curvilínea

Às vezes, os dados também exibem tendências interrompidas que são melhor caracterizadas com duas ou mais linhas/curvas de regressão, etc. Novamente, apenas encaixar uma linha reta através dos dados é arriscado, para dizer o mínimo. A parte boa sobre as três ameaças acima é que, se você seguir

as recomendações do Capítulo 4, você provavelmente não cometerá esses erros. Uma das razões pelas quais quase todas as seções do Capítulo 4 envolveram alguma visualização foi fixar no seu cérebro o fato de que a visualização exploratória deve ser parte integrante do início de qualquer análise estatística, e que uma visualização adequada irá revelar relações logarítmicas, tendências curvilíneas, tendências interrompidas, e assim por diante.

Além desses três riscos, eu quero mencionar outros dois. Um é um tanto trivial, o outro é menos. O primeiro é que seu modelo vai fazer um trabalho pior de representação dos dados (o aspecto preditivo do processo de modelagem) e de explicação dos mesmos (o aspecto explicativo do processo de modelagem) se você deixar de lado preditores importantes. O segundo é menos trivial e nos traz de volta à noção de interação, mais especificamente à questão de incluir ou não interações em seus modelos. Pode-se provavelmente distinguir três posições diferentes sobre essa questão.

Uma delas é que as interações entre variáveis independentes devem ser incluídas desde o início. Isso porque (i) se você não incluir interações na equação do modelo, elas não serão testadas e você não saberá se as interações ajudarão de fato a representar os dados de forma melhor, e (ii) se você incluiu apenas interações para as quais havia uma clara motivação teórica, seria mais difícil encontrar algo inesperado; essa seria uma maneira (não sem controversas) pela qual o trabalho exploratório se infiltra no que normalmente é uma abordagem de teste de hipóteses.

Uma segunda posição é que você só inclui interações para as quais você pode encontrar uma motivação teórica *a priori*. Isso tem a desvantagem acima, mas a vantagem é que fica difícil extrair algo nos dados.

A terceira posição pode ainda ser a mais frequente: as interações não são incluídas porque a importância do conceito não é clara para o usuário ou, tão ruim quanto, porque o software que está sendo usado torna difícil incluir as interações (Varbrul é um dos casos).

Esta questão de se incluir ou não interações é importante o suficiente para merecer um pequeno exemplo (o qual você pode reconhecer como um exercício anterior). Vamos supor que 80 alunos (L1 falantes nativos de alemão de duas turmas A e B de 40 alunos cada, um preditor denominado CLASS) participaram de um ditado em sua L1 (alemão) e um na L2 que eles estão aprendendo (inglês). Em seguida, o número de erros em inglês (ENGLISH) e em alemão (GERMAN) foram contados para determinar se é possível prever os números de erros cometidos na L2 com base no número de erros cometidos na L1 e na turma que os alunos frequentaram. Dois modelos multifatoriais podem ser adaptado aos dados, um com a interação entre GERMAN e CLASS, e um sem (lembre-se que os dois modelos da Seção 3.2.2 mostrado em (58) são meros variantes de anotação):¹

(57) ENGLISH ~ GERMAN + C LASS

(58)

a. ENGLISH ~ GERMAN + CLASS + GERMAN:CLASS

b. ENGLISH ~ GERMAN * CLASS

Os resultados dos modelos em (57) e (58) estão ilustrados na Tabela 44 e na Tabela 45, respectivamente. Ambos os modelos são altamente significativos e explicam os dados realmente bem: veja os grandes e significativos valores de R^2 . No entanto, existem vários problemas importantes e interrelacionados com o modelo sem a interação (em (57)). Primeiro, esse modelo faz um trabalho pior na representação dos dados em comparação ao modelo (em (58)): as figuras em negrito nas linhas chamadas "variância residual" mostram quanta variabilidade nos dados os modelos deixam de representar e você pode ver que esse valor é muito maior na Tabela 44. Um teste de significância mostraria que ele é de fato significativamente maior, o que é outra maneira de dizer que o modelo em (57) é significativamente pior do que o modelo em (58).

Em segundo lugar, os valores- p para os coeficientes de regressão, ou estimativas, são muito diferentes. A principal e crucial diferença é que o modelo que explica os dados de forma melhor ((58)) diz que CLASS não é significativa por si só, mas apenas na interação, enquanto que o que explica os dados de forma pior

¹ Eu não forneço os dados aqui, mas você verá este exemplo novamente em um dos exercícios para o Capítulo 5.

Tabela 45 – Resultados do modelo linear (57)

	Soma de Quadrados	Estimativa	Erro Padrão	t	p
Intercepto	23,61	2,75	1,52	1,8	0,08
GERMAN	2931,69	1,75	0,09	20,1	<0,001
CLASS	3010,30	-8,72	0,43	-20,37	<0,001
Resíduo	558,68				
R^2	R^2 Múltiplo = 0,974		R^2 Ajustado = 0,973		$F_{2,77} = 1416$ $p < 0,001$

Tabela 46 – Resultados do modelo linear (58)

	Soma de Quadrados	Estimativa	Erro Padrão	t	p
Intercepto	24,9	2,82	1,15	2,44	0,017
GERMAN	2461,42	1,64	0,07	24,29	<0,001
CLASS	0,25	-0,28	1,15	-0,25	0,807
GERMAN:CLASS	241,73	-0,515	0,07	-7,61	<0,001
Resíduo	316,75				
R^2	R^2 Múltiplo = 0,985		R^2 Ajustado = 0,984		$F_{3,76} = 1661$ $p < 0,001$

(em (57)) diz que CLASS é um efeito principal significativo. Isso não é um detalhe pouco importante: CLASS é uma variável binária, o que significa que, se for significativa, seu coeficiente é uma diferença de médias entre as duas turmas, e GERMAN é uma variável numérica, o que significa que, se for significativa, seu coeficiente é um declive de uma linha de regressão. Assim, o que o modelo em (57) leva você a acreditar é o seguinte: os alunos das duas turmas são em média diferentemente bons (diferindo em 8,72 erros), mas você pode usar uma e a mesma inclinação para ambas as turmas para prever ENGLISH de GERMAN. Isso é representado no painel esquerdo da Figura 61, com GERMAN e ENGLISH nos eixos x e y, respectivamente, e CLASS é indicado pelas letras.

No entanto, o modelo em (58) propõe algo muito diferente, a dizer, que não há diferença entre as médias das duas turmas (o valor- p de CLASS é enorme). No entanto, GERMAN:CLASS é significativo - mas o que isso significa? Significa que a inclinação entre GERMAN e ENGLISH difere significativamente entre turmas, o que é representado no painel direito da Figura 61, e mesmo se já não soubéssemos que esse modelo é melhor, o ajuste das duas linhas de regressão com suas inclinações separadas certamente parece melhor. Assim, os dois modelos dizem coisas bem diferentes sobre o que CLASS faz ...

Finalmente, e em relação ao ponto anterior, os coeficientes nos dois modelos e, portanto, suas previsões, são muito diferentes. Os resíduos do modelo pior são em média mais de 36% maiores do que aqueles do modelo melhor.

De forma resumida, nesse caso, deixar de fora a interação deixaria você com um modelo que parece ótimo na superfície (grande R^2 e altamente significativo), mas que é significativamente pior do que o modelo com a interação, o qual fornece uma exploração muito diferente sobre os dados, e que é muito pior em 'prever' os dados. Lavando isso em consideração, é surpreendente como muitas vezes as interações ainda não são exploradas (adequadamente).

Uma coisa que eu tenho visto é que os pesquisadores parecem conscientes de tais questões, mas que suas ferramentas não estão equipadas para lidar bem com interações (ou dados contínuos); novamente, Varbrul é um caso exemplar. Então, como eles poderiam tentar abordar as interações? Montando um modelo separado para cada turma:

$$(59) \quad \text{ENGLISH}_{\text{CLASSA}} \sim \text{GERMAN}_{\text{TURMAA}}$$

$$(60) \quad \text{ENGLISH}_{\text{CLASSB}} \sim \text{GERMAN}_{\text{TURMAB}}$$

Se alguém fizer isso, obtém-se de fato duas regressões simples significativas e as inclinações corretas de 1,13 para a CLASS A e de 2,16 para a CLASS B. Mas, por que isso é ainda uma má ideia?

Pausa para pensar

Figura 61 – $\text{ENGLISH} \sim \text{GERMAN} + \text{CLASS} (+ \text{GERMAN}:\text{CLASS})$

Bem, com essa abordagem, como você sabe se a diferença entre estas duas inclinações é significativa ou não? A interação não aparece em ambos os modelos em (59) e (60), então as inclinações nunca são comparadas uma com a outra, então você não obtém um valor- p e assim você não sabe se é uma diferença significativa ou não. Tenho visto plenários e artigos e outros noas quais os pesos de Varbrul para diferentes períodos de tempo foram comparados uns com os outros sem qualquer teste para verificar se a diferença entre os pesos de Varbrul de diferentes períodos de tempo foram significativos e, portanto, indicativos de mudança ou não ao longo do tempo ... Você tem que ou incluir a interação e obter um valor- p para ela, ou pelo menos verificar se os intervalos de confiança das inclinações não se sobrepõem (o que, aqui, não acontece).

Em suma, recomendo que você considere cuidadosamente a natureza das variáveis envolvidas, gaste uma quantidade considerável de tempo explorando seus dados (especialmente visualmente) antes de começar a fazer qualquer outra coisa e esteja muito consciente da importância potencial das interações.

5.1.3.1 Seleção de um modelo final

Uma vez consideradas as questões acima, um primeiro modelo é formulado, e muitas vezes esse modelo é o que é chamado de *modelo máximo*, ou seja, um modelo que inclui todas as variáveis independentes e todas as suas interações (muitas vezes só até as interações de três variáveis independentes, porque as interações de uma ordem ainda maior são extremamente difíceis de entender). No entanto, esse é geralmente apenas o ponto de partida uma vez que o modelo máximo geralmente contém preditores que não contribuem o suficiente para o modelo, e uma vez que a famosa máxima chamada lâmina de Occam (*entia non sunt multiplicanda praeter necessitatem*) essencialmente exige que você descarte preditores que não puxam seu próprio peso ou, mais formalmente, que não contribuem o suficiente para o sucesso do modelo.

A seleção do modelo é então influenciada por dois parâmetros: a *direção* da seleção do modelo e o

critério que determina se um preditor deve fazer parte ou não do modelo. Quanto ao primeiro parâmetro, existem três abordagens:

- *seleção reversa (backward selection)*: aqui, você começa com o modelo máximo como explicado acima e testa sucessivamente se você tem que descartar preditores que não contribuem o suficiente para o modelo. O processo de seleção termina quando nenhum preditor pode ser descartado mais sem tornar o modelo muito pior ou quando não há mais preditores no modelo. A eliminação de preditores começa com o nível mais alto de interatividade e prossegue para baixo na direção dos efeitos principais, e você não pode descartar um preditor que participa de uma interação de ordem superior requerida. Isso significa que você não pode excluir nem mesmo um preditor insignificante B se a interação A:B é significativa.
- *seleção posterior (forward selection)*: aqui, você começa com um modelo muito pequeno (talvez até mesmo apenas um que consiste na média geral) e teste sucessivamente se você pode adicionar preditores. O processo de seleção termina quando nenhuma adição de um preditor melhora mais o modelo ou quando todos os preditores disponíveis já estão no modelo. A adição de preditores começa com os efeitos principais e sobe até as interações de ordem superior (se seus principais efeitos já tiverem sido incluídos, como acima).
- *bidirecional (bidirectional)*: aqui, você começa com algum modelo e permite que um algoritmo normalmente automático adicione e subtraia preditores conforme a necessidade.

Eu acho que a primeira abordagem é a mais amplamente utilizada na linguística, mas também há bons argumentos para não se fazer a seleção do modelo (veja [Harrell \(2001, Seção 4.3\)](#) ou [Faraway \(2005, Seção 8.2\)](#)).

Quanto à última, eu foi intencionalmente vago acima na descrição sobre quando preditores são adicionados ou descartados, porque há novamente pelo menos duas maneiras possíveis de fazer isso (quem gostaria que a vida fosse fácil...):

- *uma abordagem baseada em significância*, segundo a qual um preditor pode ser adicionado a um modelo se ele torna o modelo significativamente melhor, e de acordo com a qual um preditor deve ser descartado se a sua eliminação não torna o modelo significativamente pior.
- *uma abordagem baseada em critérios*: o AIC (Critério de Informação de Akaike), por exemplo, é uma medida que relaciona a qualidade de um modelo com o número de preditores que o modelo contém (e, assim, aplica a lâmina de Occam). Se dois modelos explicam os dados igualmente bem, então o modelo com menos preditores terá um AIC menor. Assim, nessa abordagem, um preditor pode ser adicionado ou excluído de um modelo se isso reduzir o AIC.

Uma vez concluído o processo de seleção do modelo, você tem o que às vezes é chamado de modelo mínimo adequado, o qual pode então ser explorado no que diz respeito (i) a se o modelo como um todo é significativo ou não e a quão bem ele explica os dados e (ii) ao que cada preditor nesse modelo contribui para o modelo: o preditor é significativo, qual é a direção do(s) seu(s) efeito(s) e qual é a força do(s) seu(s) efeito(s)? Após essa longa, mas necessária, introdução teórica, as seções a seguir discutirão todas essas questões - (diferentes tipos de) modelos de regressão, efeitos principais, interações, seleção do modelo, precisão de previsão, etc. - com base em muitos exemplos práticos. A seção 5.2 discute modelos lineares para regressão linear (múltipla), ANOVAs e ANCOVAs.

Recomendação para estudo/leitura

- [Good e Hardin \(2012, Parte III\)](#) e [Crawley \(2007, Cap. 9\)](#)

5.2 Modelo Linear

Nas Seções 3.2.3 e 4.4.1, analisamos como calcular e avaliar a correlação entre uma variável independente de proporção e uma variável dependente de proporção utilizando o coeficiente r de correlação produto-momento de Pearson e a regressão linear. Nesta seção, estenderemos isso para o caso de múltiplas variáveis independentes. Os dados que vamos explorar envolvem a questão de como prever os tempos de reação dos falantes em relação a substantivos em uma tarefa de decisão lexical, e envolvem as seguintes variáveis:²

- uma variável dependente, a dizer, o tempo de reação (RT) a palavras em uma tarefa de decisão lexical REACTTIME, em cuja correlação com as seguintes variáveis independentes você está interessado (neste caso, a variável dependente é uma média dos tempos de reação, o que normalmente não seria o caso; isso é por razões expositivas apenas e não importa aqui);
- uma variável numérica independente FREQUENCY, que corresponde a suas frequências registradas (de acordo com Kučera e Francis (1967));
- uma variável categórica independente FAMILIARITY, que é um índice resumindo o julgamento de familiaridade dos participantes da pesquisa com o referente da palavra;
- uma variável binária independente IMAGEABILITY, que é um índice que resume os julgamentos dos participantes da pesquisa para a imaginabilidade do referente da palavra;
- uma variável numérica independente MEANINGFULNESS, que indica o julgamento médio dos participantes da pesquisa para a significância das palavras apresentadas como estímulos.

Este é o procedimento geral do processo de modelagem linear que iremos usar:

Procedimentos

- Formulação das hipóteses
- Importação dos dados para o R, preparando-os para a modelagem e explorando-os
- Cálculo, seleção e interpretação do modelo linear:
 - obtenção dos valores- p para todos os preditores e para o modelo como um todo
 - interpretação dos coeficientes/estimativas da regressão das bases de (i) valores preditos e (ii) gráficos dos valores preditos/observados
- Teste das principais pressuposições dos testes³
 - as variâncias dos resíduos são homogêneas e normalmente distribuídas na população da qual a amostra foi retirada, ou, pelo menos, nas próprias amostras
 - os resíduos são normalmente distribuídos com média zero na população da qual a amostra foi retirada, ou, pelo menos, nas próprias amostras

² As palavras (mas não os tempos de reação) são emprestadas de um conjunto de dados do livro abrangente de Baayen (2008); as outras características dessas palavras foram tiradas da, ou feitas/modificadas com base na, Base de Dados Psicolinguística do MRC; veja <<http://www.psy.uwa.edu.au/mrcdatabase/mrc2.html>> para explicações mais detalhadas sobre as variáveis em geral.

³ Existem outras pressuposições - como independência dos resíduos, ausência de colinearidade - mas, por razões de espaço, não serão discutidos. Verifique Fox e Weisberg (2011, Cap. 6) and Field, Miles e Field (2012, Seção 7.7) para uma exploração mais ampla.

Como você pode ver, nesta seção, vamos testar algumas pressuposições da modelagem linear somente depois de termos um modelo, pois você só pode verificar resíduos quando você tem um modelo a partir do qual eles podem ser calculados. Além disso, esta seção é bastante diferente daquelas no Capítulo 4. Tenho observado ao longo da minha experiência - tanto no ensino como na minha própria pesquisa - que uma das maiores dificuldades na modelagem linear não é obter um resultado significativo, mas sim entender o que os coeficientes de regressão (ou estimativas, vou usar esses termos alternativamente) dos resultados significam, um problema agravado pelo fato de que poucos livros (i) dizem muito explicitamente em uma linguagem que os iniciantes entendem o que o resultado significa e (ii) discutam o que o resultado significa para duas maneiras diferentes de fazer a modelagem linear. A fim de abordar essas duas questões, esta seção irá explicar detalhadamente seis modelos lineares bastante simples que se diferem em termos dos preditores que envolvem para lhe mostrar exatamente - tanto numericamente quanto visualmente - o que significam os coeficientes de regressão. Além disso, cada um desses modelos lineares será calculado de duas maneiras. Uma é o padrão frequentemente usado em alguns software comerciais que infelizmente ainda estão em uso amplo, a outra é a forma padrão no R.

Antes de começar com a modelagem, é importante que você perceba que, se este fosse um estudo real, você não executaria muitos modelos diferentes nos dados para testar hipóteses diferentes, mas sim sobrepostas, como farei aqui. Eu vou passar com você por esses modelos apenas para que você veja como eles são ajustados, interpretados e visualizados - o que você faria se este fosse um estudo real é um processo de seleção de modelos do tipo discutido na Seção 5.2.7.

Vamos começar formulando as hipóteses, as quais serão aplicáveis a todos os modelos lineares desta seção. Utilizamos uma extensão do coeficiente de determinação R^2 , a dizer, seu múltiplo equivalente de regressão múltipla R^2 :

H_0 : Não há correlação entre REACTTIME e os preditores (variáveis independentes e suas interações): múltiplo $R^2 = 0$.

H_1 : Há uma correlação entre REALTIME e os preditores (variáveis independentes e suas interações): múltiplo $R^2 > 0$.

Vamos agora importar os dados (a partir de <_inputfiles/05-2_reactiontimes.csv>) de forma que as palavras para as quais existem dados se tornem os nomes das linhas, o que é útil para alguns gráficos (resultado não mostrado):

Código em R

```
> RTs<-read.delim(file.choose(), row.names=1)¶
> summary(RTs)¶
```

O resumo mostra que esse é um conjunto de dados muito pequeno - um estudo real deveria ser baseado em mais dados. Além disso, encontramos algo que é muito realista, ou seja, que algumas variáveis têm dados faltando, marcados como NA, como deveriam estar. Por agora, vamos adotar uma solução rápida e fazer uso do fato de que a função de modelagem linear do R, `lm`, descartará automaticamente aqueles casos de variáveis no modelo que têm dados faltando.

Antes de começarmos com a modelagem, existem duas maneiras pelas quais os dados podem ser preparados para uma melhor análise. Uma delas é que às vezes é útil padronizar em z variáveis numéricas (com `scale`, lembre-se da Seção 3.1.4), o que pode auxiliar com o problema da colinearidade (o indesejável fenômeno de que vários de seus preditores estão altamente correlacionados) e ajudar na interpretação dos resultados porque a média dos preditores padronizados é zero, o que, por exemplo, torna os interceptos e coeficientes de regressão fáceis de entender. (Por outro lado, isso também pode tornar os resultados mais difíceis de entender porque perdemos as unidades originais da escala). Portanto, não usaremos isso aqui, mas é bom manter isso em mente para mais tarde.

A segunda coisa que vamos fazer tem a ver com fatores (e agora você verá porque eu falei tanto sobre eles acima). Se você olhar para o resultado do resumo, você verá que o fator FAMILIARITY tem níveis que são ordenados alfabeticamente, mas que essa ordem não é compatível com as informações ordinais que os níveis comunicam. Queremos ou *LO*, *MED*, e *hi*, ou *hi*, *MED*, e *LO*, mas não *hi*, *LO*, *MED*. Assim, tanto para FAMILIARITY quanto para IMAGEABILITY, reordenamos os níveis de forma ordenada, razoável e homogênea:

Código em R

```
> RTs$FAMILIARITY=factor(RTs$FAMILIARITY, levels=levels(RTs$FAMILIARITY)[c(2, 3, 1)])
> RTs$IMAGEABILITY=factor(RTs$IMAGEABILITY, levels=levels(RTs$IMAGEABILITY)[c(2, 1)])
> summary(RTs)
```

Uma vez que a exploração gráfica ou tabular (por exemplo, com *boxplots* ou gráficos *ecdf*), a qual eu recomendo fortemente que você sempre faça com os dados, não produz qualquer outra coisa que necessite de correção/preparação, agora podemos anexar RTs e carregar alguns pacotes que vamos usar. Finalmente, o arquivo de códigos define algumas funções que usaremos algumas vezes - *se.mean*, *ci.mean* e *error.bar* - então basta copiar e colar esses códigos no R para que você possa usar as funções abaixo.

Código em R

```
> attach(RTs)
> library(aod); library(car); library(effects); library(gvlma);
library(multcomp); library(rgl)
```

5.2.1 Um modelo linear com preditor binário

Embora este primeiro modelo linear seja o mais simples de todos, esta seção será um pouco mais longa porque todas as coisas que têm a ver com modelos lineares aparecerão pela primeira vez. Então, não se desespere, tudo o que virá depois será mais curto. Para testar se a IMAGEABILITY está correlacionada com REALTIME, ajustamos o modelo linear mais simples possível. No entanto, para obtermos resultados que são comparáveis com algumas saídas de software comerciais, primeiro definimos a maneira pela qual o R calcula contrastes como mostrado aqui (haverá mais sobre isso em breve), então nós ajustamos um modelo linear (quando também dizemos ao R de qual estrutura de dados as variáveis são com o argumento *data*) e, em seguida, analisamos o resultado:

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))
> model.01=lm(RT~IMAGINABILIDADE, data=RTs)
> summary(model.01)
```

Residuals:

Min	1Q	Median	3Q	Max
-84.629	-40.016	2.145	26.975	160.799

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	620.666	6.998	88.693	<2e-16 ***
IMAGEABILITY1	12.987	6.998	1.856	0.0693 .

--

Residual standard error: 50.5 on 51 degrees of freedom
(24 observations deleted due to missingness)
Multiple R-squared: 0.06326, Adjusted R-squared: 0.0449
F-statistic: 3.444 on 1 and 51 DF, p-value: 0.06925

Vamos começar pela parte inferior: o modelo como um todo não é significativo, como mostra o valor- p , o qual, por sua vez, é calculado a partir do valor- F com $gl = 1, 51$ (aqui é: `pf(3.444, 1, 51, lower.tail = FALSE)`). A correlação múltipla entre IMAGEABILITY e REALTIME, R^2 múltiplo, varia teoricamente de 0 a 1 e quantifica a variabilidade representada, então um valor de 0,06326 é realmente pequeno. Além disso, o valor geralmente relatado é o R^2 ajustado. Este valor R^2 é ajustado de tal forma que você assume uma pequena penalidade para cada preditor incluído em seu modelo. Assim, se, numa tentativa desesperada de explicar mais variabilidade, você acrescentasse uma variável inútil ao modelo, então é muito provável que qualquer quantidade pequena de variação aleatória que a variável inútil explique seja consumida pela penalidade. Assim, esse ajuste traz a lâmina de Occam para a modelagem. Obviamente, o R^2 ajustado também é muito pequeno. Então, há um aviso de que o R apagou 24 observações, porque esses casos tinham um NA nas variáveis do modelo.

Ignoramos o erro padrão residual e rapidamente pulamos para o topo do resultado,⁴ onde obtemos um resultado de resumo sobre os resíduos e já podemos ver que esses são quase distribuídos de forma normal - tudo o que aprendemos aqui deve ser interpretado cautelosamente (Voltaremos a essa questão).

Embora não tenhamos falado sobre o que os coeficientes significam, deixe-me apontar desde já o óbvio: eles são apenas estimativas, e é assim que o R os rotula, o que significa que você pode obter intervalos de confiança para eles, e o fato de que o intervalo de confiança para IMAGEABILITY inclui 0 já sugere que, seja ele qual for - a ser discutido em breve - ele não é significativo:

Código em R

```
> confint(model.01)
2.5%      97.5%
(Intercept) 606.617138 634.71498
IMAGEABILITY1 -1.061602 27.03624
```

Antes de passarmos para os coeficientes e seus valores- p , vamos executar mais duas linhas de código que são muito úteis para preditores com mais de um grau de liberdade (gl), isto é, preditores que não são nem binários nem numéricos (isso não se aplica aqui, mas eu menciono isso de qualquer forma por uma questão de coerência).

⁴ O erro padrão residual é a raiz do quociente das somas residuais dos quadrados dividido pelo graus de liberdade do resíduo (no R: `sqrt(sum(residuals(model.01)^2)/51)`).

Código em R

```

> drop1(model.01, test="F")
Single term deletions

Model:
RT ~ IMAGEABILITY
Df Sum of Sq RSS AIC F value Pr(>F)
<none>                130059 417.69
IMAGEABILITY 1 8783.6 138843 419.15 3.4443 0.06925 .
--
[...]
```

```

> Anova(model.01, type="III")
Anova Table (Type III tests)
Response: RT
Sum Sq Df F value Pr(>F)
(Intercept) 20060844 1 7866.4268 < 2e-16 ***
IMAGEABILITY 8784 1 3.4443 0.06925 .
Residuals 130059 51
[...]
```

Essas funções são maneiras importantes para obter valores- p para preditores. A primeira, `drop1`, olha para todos os preditores no modelo e verifica qual preditor poderia teoricamente ser excluído do modelo nesta fase do processo de seleção do modelo, e para os preditores que puderam ser deletados neste ponto, ela retorna um valor- p para o teste do modelo original, `model.01`, em contraste com o modelo que você obteria sem esse preditor. A segunda, `Anova`, está disponível na biblioteca `car`. Ela calcula um valor- p para preditores que é o mesmo que software comerciais retornam como padrão.⁵

Com esse resultado, passemos agora aos coeficientes. Primeiro, a parte mais simples, os valores- p , em seguida, os coeficientes. O valor- p para a interceptação é geralmente desconsiderado: ele testa a H_0 de que a interceptação é 0, mas há poucas aplicações nas quais isso é relevante. Mais interessante é o valor- p para o preditor `IMAGEABILITY`. (Na verdade, o R escreve `IMAGEABILITY1`, vou explicar isso em breve.) Neste caso mais simples, onde o nosso modelo tem apenas um preditor binário, o valor- p é o mesmo que o valor- p do modelo todo, e o mesmo daquele preditor no resultado de `drop1` e da saída da `Anova`: 0,06925. Portanto, o preditor não tem um efeito significativo e, de certa forma, o resultado de `drop1` diz isso mais intuitivamente porque o que `drop1` está dizendo essencialmente é “se você tirar `IMAGEABILITY` de `model.01`, então o modelo resultante não é significativamente pior ($p=0.06925$)”. Uma maneira diferente de ver isso é mostrar que o coeficiente de regressão não é significativamente diferente de 0. Tudo isso é idêntico ao que você obtém por meio de um teste- t .

Embora este modelo/preditor não seja significativo, prosseguiremos com a discussão e geraremos o gráfico como se ele fosse, porque a essa altura eu quero lhe mostrar como o resultado de tal modelo é interpretado e gerado em gráfico. Um processo de seleção de modelos mais realista é apresentado na Seção 5.2.7.

Então - finalmente - o que significam as estimativas, os 620.666 de intercepto (Intercept) e os 12,987 para `IMAGEABILITY1`? Eu recomendo abordar essa questão com base nos valores que o modelo prevê como na Seção 4.4.1:

⁵ Essa questão de soma de quadrados (o argumento `type="III"`) é altamente debatida. Eu não vou me envolver aqui na discussão sobre qual abordagem é melhor, mas vou usar `type="III"` por razões de comparabilidade com outros software mesmo que `type="II"` muitas vezes seja mais útil; veja [Crawley \(2002, Cap. 18\)](#), [Crawley \(2007, p. 368\)](#), [Larson-Hall \(2010, p. 311-313\)](#), [Fox e Weisberg \(2011, Seção 4.4 e 4.6\)](#), [Field, Miles e Field \(2012, p; 475\)](#) e a lista de ajuda do R. Como você pode notar, ambos retornam o valor- p já conhecido para o único preditor.

Código em R

```
> preds.hyp=expand.grid(IMAGEABILITY=levels(IMAGEABILITY));
preds.hyp[c("PREDICTIONS", "LOWER", "UPPER")]=predict(
model.01, newdata= preds.hyp, interval="confidence");
preds.hyp
IMAGEABILITY PREDICTIONS LOWER UPPER
1          lo 633.6534 612.5138 654.7929
2          hi 607.6787 589.1691 626.1884
```

Assim, `model.01` prevê que, quando a palavra é de baixa imaginabilidade, os tempos de reação das pessoas serão cerca de 26 ms mais lentos do que quando a palavra é de alta imaginabilidade. Apenas para deixar isso claro: isso significa que o modelo faz apenas duas previsões diferentes: quando a `IMAGEABILITY` é baixa, ela sempre prediz um RT de 633,6534, e quando a `IMAGEABILITY` é alta, ela sempre prediz um RT de 607,6787, e esses dois valores previstos também são as médias observadas: tente usar `tapply(RT, IMAGEABILITY, mean)`. Observe também o tanto que os intervalos de confiança das duas previsões se sobrepõem.

Se olharmos para `preds.hyp`, você já pode inferir o que a regressão de estimativas significa. Quando você calcula o modelo linear como fizemos aqui, isto é, com contrastes de soma!, então esses dois valores significam o seguinte:

- a intercepção, 620,666, é a média não ponderada(!) das médias da variável dependente, quando é agrupada pela variável independente. Ou seja, 620,666 é a média de 633,6534 e 607,6787, e é uma média não ponderada porque não leva em consideração que os dois níveis de `IMAGEABILITY` não são igualmente frequentes. Item [-] o coeficiente de `IMAGEABILITY`, 12,987, é o que você tem que adicionar à intercepção para obter o RT previsto para o primeiro nível de `IMAGEABILITY` (daí o 1): $620,666 + 12,987 = 633,653$. (E, uma vez que a intercepção é a média das médias, se você subtrair o coeficiente da intercepção, você obtém o RT previsto para o segundo nível de `IMAGEABILITY`: $620,666 - 12,987 = 607,679$.)

Isso é representado visualmente na Figura 62.

Agora, você pode se perguntar por que razão está escrito “declive” na Figura 62. A resposta é porque você pode conceituar o intercepto como um valor do eixo x igual a 0, e `IMAGEABILITY:1` como um valor do eixo y igual a 1, o que é basicamente o processo subjacente à modelagem linear: para as variáveis numéricas, os efeitos são dados como declives que representam o quanto o valor de y previsto muda para cada mudança de unidade no eixo x, mas com a perspectiva acima você pode também entender os coeficientes para os níveis de fatores (por exemplo, 12,987) como declives.

Finalmente, embora esse modelo particular seja tão simples que os coeficientes etc. podem ser entendidos sem qualquer visualização, isso pode mudar rapidamente, apresentarei aqui mesmo duas formas através das quais os dados podem ser visualizados. O código para gerar os gráficos na Figura 63 está no arquivo de códigos. À esquerda temos um gráfico de barras comum das médias, a única coisa que eu adicionei são os intervalos de confiança para as médias. O gráfico à direita é um gráfico de efeito muito fácil de ser gerado.

Tudo o que foi dito acima corresponde ao que software comerciais reportariam nos resultados. No entanto, a maneira padrão no R é realmente um pouco diferente, felizmente é apenas um pouco ... Uma vez que eu quero que você conheça a abordagem padrão do R e uma vez que essa abordagem irá ajudá-lo a compreender a regressão logística mais tarde, vou agora discuti-la muito brevemente. A única diferença real na execução da abordagem padrão do R é que agora você usa os contrastes padrões do R, contrastes de tratamento. Se em seguida você gerar o modelo novamente, os valores de R^2 , o valor-p geral, a maioria é igual, mas não os coeficientes:

Figura 62 – Estimativas da regressão do modelo.01 com soma de constrastes

Código em R

```
> options(contrasts=c("contr.treatment", "contr.poly"))¶
> model.01=lm(RT~IMAGEABILITY, data=RTs)¶
> summary(model.01)¶
[...]
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	633.65	10.53	60.177	<2e-16 ***
IMAGEABILITYhi	-25.97	14.00	-1.856	0.0693 .

```
--
> confint(model.01)¶
[...]
```

Depois do que fizemos acima, você provavelmente verá imediatamente o que o intercepto e o coeficiente de IMAGEABILITY representam:

- o intercepto, 633,65, é a média observada/prevista da variável dependente, quando a variável independente IMAGEABILITY é seu primeiro nível, I_L .
- o coeficiente para IMAGEABILITY, -25,97 é o que você adiciona ao intercepto para obter o TR previsto para o segundo nível de IMAGEABILITY (daí o HI): $633,65 + (-25,97) = 607,68$; o valor- p mostra que a diferença entre o intercepto (representando IMAGEABILITY:LO) e seu tempo de reação previsto para IMAGEABILITY:HI não é significativo.

Isso também é representado na Figura 64, onde, como discutido acima, as anotações de $x=0$ e $x=1$ motivam o uso da palavra *declive* no gráfico.

Como você pode ver, neste caso simples ambas as abordagens produzem coeficientes diferentes, mas equivalem aos mesmos testes de significância (com `drop1` novamente, veja o arquivo de códigos) e às mesmas previsões (no novo `preds.hyp`; veja o arquivo de códigos). Além disso, observe que eu forneço

Gráfico do efeito da IMAGEABILITY

Figura 63 – Efeitos do modelo.01: gráfico de barras com médias observadas/preditas e seus intervalos de 95% de confiança (esquerda); gráfico dos efeitos da biblioteca effects (direita)

alguns códigos extras para obter os valores- p para os coeficientes usando `wald.test` e `glht` no arquivo de códigos. Você deve sempre executar esses códigos também, uma vez que serão muito úteis mais tarde. Futuramente você pode querer explorar [Bretz, Hothorn e Westfall \(2011\)](#).

Você pode resumir os resultados da seguinte maneira: “Um modelo linear foi ajustado com REACTTIME como variável dependente e IMAGEABILITY (baixa vs. alta) como a variável independente. O modelo não foi significativo ($F=3,444$, $gl_1 = 1$, $gl_2 = 51$, $p=0,069$). Houve apenas uma tendência marginalmente significativa de que a imaginabilidade baixa e imaginabilidade alta correlacionaram-se com a reação mais lenta e a reação mais rápida, respectivamente. [Mostrar gráficos].”

5.2.2 Um modelo linear com um preditor categórico

Nesta seção, ainda consideramos apenas um preditor - então, na verdade, ainda não estamos fazendo análise multifatorial - mas tornamos o modelo um pouco mais complexo estudando um preditor com três

Figura 64 – Estimativas de regressão do `model.01` com contrastes de tratamento

níveis (FAMILIARITY, o que significa que você não poderia fazer um teste-*t* como antes.⁶ Primeiramente, vamos usar a abordagem sem contrastes de somas (de agora em diante, não mostrarei mais todos os resultados):

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))¶
> model.01<-lm(RT~FAMILIARITY, data=RTs)¶
> summary(model.01)¶
> confint(model.01)¶
```

Esse modelo é significativo: o valor-*p* global é <0,001. Uma vez que esse é também um modelo com somente um preditor, você sabe que esse é também o valor-*p* para esse preditor. No entanto, se você olhar para a tabela de coeficientes, você não o encontra lá. Em vez disso, você tem um intercepto e, então, dois valores-*p* diferentes. Como você obtém um valor-*p* para FAMILIARITY de outra forma que não seja considerando o valor-*p* global (por exemplo, quando você tem mais de um preditor)? Este é um caso onde `drop1` e `Anova` são necessários porque - lembre-se do supracitado - aqui o (único) preditor tem mais de 1 grau de liberdade porque ele não é nem binário nem numérico. Assim, você usa `drop1` e `Anova`:

Código em R

```
> drop1(model.01, test="F")¶
> Anova(model.01, type="III")¶
```

Agora você tem o valor-*p* para FAMILIARITY, e desta vez você pode ver como a abordagem baseada em significância e a abordagem baseada em critérios concordam: FAMILIARITY é significativo e tirá-lo do

⁶ A propósito, esta seção, bem como a anterior, aborda modelos lineares que alguns chamam de ANOVAs, análises de variância. Entretanto, uma vez que a abordagem subjacente nas regressões lineares com apenas variáveis independentes numéricas, em ANOVAs com apenas variáveis independentes categóricas e em ANCOVAs com variáveis independentes categóricas e numéricas é a mesma - no R todas elas se encaixam em `lm` - Eu não vou discutir as diferenças entre esses métodos, mas sim me concentrar em seus pontos em comum.

Figura 65 – Estimativas da regressão do model .01 com soma de contrastes

modelo aumenta o AIC (Critério de Informação de Akaike) consideravelmente.

Então, o que significam as estimativas? Novamente, abordamos isso através dos valores previstos. Observamos que há um bom efeito ordinal: à medida que a FAMILIARITY aumenta, os R diminuem, o que faz sentido.

Código em R

```
> preds.hyp=expand.grid(FAMILIARITY=levels(FAMILIARITY));
preds.hyp[c("PREDICTIONS", "LOWER", "UPPER")] =
predict(model.01, newdata=preds.hyp,
interval="confidence"); preds.hyp
```

Com base em `preds.hyp`, você pode novamente inferir o que as estimativas significam, e isso é também visualizado na figura 65:

- o intercepto, 622,774, é a média não ponderada(!) das médias da variável dependente, quando é agrupada pela variável independente. Ou seja, 622,774 é a média de 663,2880, 613,6471, e 591,3879, e é uma média não ponderada porque não leva em consideração que os níveis de FAMILIARIDADE não são igualmente frequentes;
- o coeficiente de FAMILIARITY1, 40,514, é o que você tem que adicionar à interceptação para prever o RT para o primeiro nível de FAMILIARITY (daí o 1);
- o coeficiente para FAMILIARITY2, -9,127, é o que você tem que adicionar à interceptação para prever o RT para o segundo nível de FAMILIARITY;

- e se você subtrair ambos os coeficientes de FAMILIARITY do intercepto, você obtém o RT previsto para o terceiro nível de FAMILIARITY.

Note que você não obtém valores- p para todas as diferenças entre o intercepto e os níveis, e às vezes você pode querer executar uma variedade de testes sobre as diferenças entre as médias. Uma forma (bastante conservadora) de abordar esta questão envolve a função TukeyHSD.

Código em R

```
> TukeyHSD(aov(model.01), ordered=TRUE)¶
```

O argumento principal dessa função é um objeto criado pela função aov (uma alternativa para anova), o qual, por sua vez, requer o modelo linear de interesse como argumento. Como resultado, você obtém uma tabela para as três comparações que você pode fazer entre três médias. Você obtém as diferenças entre as médias, os intervalos de confiança inferior e superior para as diferenças, e os valores- p que foram ajustados para o fato de que você está de repente executando três testes de significância no mesmo conjunto de dados. Por que os valores- p teriam que ser ajustados neste caso?

Pausa para pensar

O objetivo de um nível de significância é certificar-se de que, se você aceitasse uma H_1 , sua probabilidade de fazer isso incorretamente seria $<0,05$. Agora, se você rejeitar duas H_0 independentes com cada $p = 0,05$, qual é a probabilidade de que você faça isso corretamente ambas as vezes? É $0,9025$, ou seja, $90,25\%$. Por quê? Bem, a probabilidade de que você esteja certo em rejeitar a primeira H_0 é $0,95$. Mas a probabilidade de que você esteja sempre correto quando rejeita a H_0 em dois testes independentes é $0,95^2 = 0,9025$. Esta é a mesma lógica de quando se quer saber a probabilidade de se obter dois números seis quando você joga dois dados simultaneamente: $1/6^2 = 1/36$. A probabilidade de que você esteja sempre correto quando rejeita a H_0 em três testes independentes é $0,95^3 = 0,857375$. Na verdade, se você considerar 13 H_0 s, então a probabilidade de que você não erre uma vez se você rejeitar todas elas é, de fato, perigosamente perto de $0,5$: $0,951^3 \approx 0,5133$, também muito longe de $0,95$. Assim, a probabilidade de erro que você usa para avaliar n H_0 s não deve ser $0,05$ - ela deve ser menor que isso. Desta forma, quando você executar todos os n testes, a sua probabilidade geral de estar sempre certo será de $0,95$. Assim, se você quiser testar n H_0 s, você deve usar $p = 1 - 0,95^{1/n}$. Para 13 hipóteses nulas, isso significa $p \approx 0,00394$. Então, a probabilidade de que você esteja correto em qualquer rejeição é $1 - 0,00394 = 0,99606$, e a probabilidade de que você esteja certo em relação a todas as 13 rejeições é $0,99606^{13} \approx 0,95$. Uma heurística mais curta que é tão conservadora quanto essa (na verdade, muito conservadora) é a correção de Bonferroni. A correção de Bonferroni consiste apenas em dividir o nível de significância desejado - isto é, geralmente $0,05$ - pelo número de testes - aqui 13. Você obtém $0,05/13 \approx 0,003846154$, o que é próximo (o suficiente) da probabilidade exata de $0,00394$ calculada acima. Assim, se você fizer vários testes *post hoc* com um conjunto de dados, você geralmente precisará ajustar o nível de significância, o que torna mais difícil de você obter resultados significativos extraídos dos seus dados, o que deve motivá-lo a formular H_1 s sensatas antecipadamente em vez de fazer testes excessivos a posteriori.

Voltando aos dados com os quais estamos lidando: podemos ver que a diferença entre os níveis médio e alto de FAMILIARITY não é significativa, mas as outras duas diferenças são. O que isso significa?

Pausa para pensar

Isso significa que a lâmina de Occams exigiria que você testasse agora se você necessita manter a diferença entre a familiaridade média e alta ou se você deve combinar as duas, e nós faremos isso na seção 5.2.7.

Como último passo para este modelo, você pode gerar alguns gráficos novamente, e o arquivo de códigos lhe mostrará como gerar gráficos como o da Figura 63 para este modelo.

Agora, vamos explorar brevemente este mesmo modelo, mas agora novamente com o padrão do R de tratamento de contrastes:

Código em R

```
> options(contrasts=c("contr.treatment", "contr.poly"))
> model.01=lm(RT~FAMILIARITY, data=RTs)
> summary(model.01)
[...]
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)    663.29   13.23   50.118 < 2e-16 ***
FAMILIARITYmed -49.64   15.59   -3.185  0.002449 **
FAMILIARITYhi  -71.90   18.72   -3.842  0.000334 ***
--                                     [...]
```

Depois do que fizemos acima, as estimativas são provavelmente claras:

- o intercepto, 663,29, é a média observada/revista quando a variável independente FAMILIARITY é seu primeiro nível, L0.
- o coeficiente para FAMILIARITYmed, -49,64 é o que você adiciona ao intercepto para prever o RT para o segundo nível de FAMILIARITY.
- o coeficiente de FAMILIARITYhi, -71,90 é o que você adiciona ao intercepto para prever o RT para o terceiro nível de FAMILIARITY.

Isso também é representado na Figura 66, onde, como discutido acima, a anotação de $x = 0$ e $x = 1$ (duas vezes, uma para cada estimativa) ajuda a motivar o uso da palavra “declive” no gráfico. Assim, de alguma forma, é tudo como antes na Seção 5.2.1, e você pode resumir o modelo desta seção baseado no que já fez acima.

5.2.3 Um modelo linear com um preditor numérico

Estamos ainda apenas nos preparando para modelos multifatoriais. Nos dois últimos modelos monofatoriais, o único preditor foi um fator (binário ou categórico) e seus efeitos foram diferenças entre as médias. No entanto, também começamos a abordar isso como um declive, ao conceituar diferenças entre médias como declives do valor de y em um nível de referência ($x = 0$) para um valor de y em um nível definido como $x = 1$. Nesta seção, vamos revisar brevemente o caso de um preditor numérico, isto é, o que discutimos na Seção 4.4.1. Uma coisa boa é que, com apenas um preditor de escala de intervalo, não temos que cobrir dois tipos de contrastes. Vamos analisar a correlação entre FREQUENCY e REACTTIME.

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))
> model.01=lm(RT~FREQUENCY, data=RTs)
> summary(model.01)
> confint(model.01)
```

Figura 66 – Estimativas da regressão do model.01 com tratamentos de contrastes

Já estudamos tais casos, tanto com `cor.test` quanto com `lm`, então eu não vou passar por todos os resultados em detalhes novamente. Basta dizer que o modelo é significativo, porque seu único preditor é significativo, etc. Uma vez que o preditor é numérico, não precisamos realmente das duas linhas a seguir, mas apenas para fixá-las, aqui estão elas novamente, e elas retornam o mesmo valor-*p*.

Código em R

```
> drop1(model.01, test="F")
> Anova(model.01, type="III")
```

Para determinar o que as estimativas significam, seguimos a mesma estratégia de antes e calculamos as previsões para os valores do intervalo conhecido:

Código em R

```
> preds.hyp=expand.grid(FREQUENCY=floor(min(FREQUENCY)):
ceiling(max(FREQUENCY))); preds.hyp[,c("PREDICTIONS",
"LOWER", "UPPER")]=predict(model.01, newdata=
preds.hyp, interval="confidence"); preds.hyp
```

Você pode reconhecer o que você já deve ter inferido com base no exposto cima:

- a intercepção, 667,03, é o RT previsto quando a variável independente FREQUENCY é 0.
- o coeficiente para FREQUENCY, -24,266 o aumento no RT previsto (i.e., dado ao menos, o decline) para cada aumento de unidade de FREQUENCY.

Como de costume, você deve gerar um gráfico com os dados para obter uma impressão do ajuste, e o arquivo de códigos fornece alguns exemplos de como você pode fazer isso.

Agora que nós cobrimos o básico, podemos finalmente passar para os modelos lineares multifatoriais. Embora esta parte introdutória possa ter parecido longa, ter fornecido todos os detalhes nela presentes tornará as coisas mais fáceis agora.

5.2.4 Um modelo linear com dois preditores categóricos

Começamos com um modelo no qual tentamos prever REACTTIME com base em duas variáveis categóricas independentes, IMAGEABILITY e FAMILIARITY, e sua interação, IMAGEABILITY:FAMILIARITY. Como antes, começamos com um modelo baseado em contrastes de soma. Lembre-se da notação usando o asterisco para dizer ‘todos esses efeitos principais e suas interações’:

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))¶
> model.01=lm(RT~IMAGEABILITY*FAMILIARITY, data=RTs)¶
> summary(model.01)¶
> confint(model.01)¶
```

O resultado se torna mais complexo... O modelo como um todo é significativo ($p = 0,01138$), podemos ver que a IMAGEABILITY não é significativa, mas não temos os valores- p individuais para FAMILIARITY e para a interação IMAGEABILITY:FAMILIARITY. Assim, antes de tentarmos entender os coeficientes/estimativas, vamos dar uma olhada rápida em drop1 e Anova:

Código em R

```
> drop1(model.01, test="F")¶
> Anova(model.01, type="III")¶
```

Desta vez, os resultados dos dois são diferentemente abrangentes. O resultado de drop1 segue a lógica acima da seleção reversa de modelo e retorna valores- p para os preditores que poderiam ser descartados *neste momento*. Uma vez que há uma interação de duas variáveis e nada mais complexo do que isso no modelo, você pode descartar essa interação, mas você não pode, neste ponto, eliminar qualquer variável da interação enquanto a interação ainda estiver no modelo. Assim, drop1 retorna somente o valor- p para o modelo com versus sem a interação e, uma vez que a interação não é significativa, deve-se eliminá-la (em conformidade com a lâmina de Occam).

O resultado de Anova é mais abrangente e retorna valores- p para todos os preditores no modelo. Você pode reconhecer os valores- p para IMAGEABILITY de `summary(lm())`, e o valor- p da interação do resultado de drop1. Não vamos eliminar a interação agora, porque, neste ponto, eu quero lhe mostrar como tal resultado de modelo é interpretado e gerado em gráfico. Novamente, o processo de seleção de modelo mais realista será mostrado na Seção 5.2.7.

Agora, vamos aos preditores e suas estimativas:

Código em R

```
> preds.hyp=expand.grid(IMAGEABILITY=levels(IMAGEABILITY),
FAMILIARITY=levels(FAMILIARITY)); preds.hyp[
c("PREDICTIONS", "LOWER", "UPPER")]=predict(model.01,
newdata=preds.hyp, interval="confidence"); preds.hyp¶
```

Eu não vou explicar cada coeficiente em detalhes aqui - veja o arquivo de códigos para definições detalhadas de cada estimativa - por duas razões. Primeiro, para economizar espaço: você vai ver quão longa e complicada a definição das estimativas no arquivo de códigos pode se tornar. Em segundo lugar, o objetivo principal de gerar `preds.hyp` é não ter que olhar tanto para os coeficientes. Claro que você ainda deve entender a explicação no arquivo de códigos, mas, na prática, entender os coeficientes de um modelo com, digamos, cinco interações significativas de 3 vias e 10 outros preditores com base nos coeficientes é praticamente impossível. Assim, leia a explicação dos coeficientes no arquivo de códigos com cuidado, execute os códigos para verificar as minhas explicações, e tente reconhecer seus efeitos nos dados, mas, por enquanto, vamos explorar o modelo com base em suas previsões, que mostram que

- as média observadas/previstas não mudam muito à medida que IMAGEABILITY muda (com o crescimento de FAMILIARITY);
- as média observadas/previstas diminuem à medida que FAMILIARITY aumenta (com o crescimento de IMAGEABILITY)
- há uma pista de interação (mas sabemos com base no que vimos acima que ela não é significativa) porque, quando FAMILIARITY é LO (baixa, do inglês *low*) ou MED, então uma mudança de IMAGEABILITY LO para HI (alta, do inglês *high*) agiliza os tempos de reação, mas tem o efeito oposto quando FAMILIARITY é HI.

O teste de Tukey mostra que, com uma abordagem de teste *a posteriori* muito conservadora, dificilmente há algo significativo nos dados. Mas, vamos visualizar os dados. O arquivo de códigos mostra diferentes tipos de gráficos, gráficos de interação usando linhas, um gráfico de barras de médias e intervalos de confiança, gráficos de pontos de médias e um *boxplot* colorido das medianas observadas e seus *notches*, bem como as médias e seus intervalos de confiança. Finalmente, o último é um gráfico de efeitos, o qual novamente mostra claramente que essa interação não é significativa: as linhas para as médias são quase paralelas.

Agora, que tal a mesma análise com contrastes de tratamento?

Código em R

```
> options(contrasts=c("contr.treatment", "contr.poly"))¶
> model.01=lm(RT~IMAGEABILITY*FAMILIARITY, data=RTs)¶
> summary(model.01)¶
> confint(model.01)¶
> drop1(model.01, test="F")¶
```

Novamente, tudo é igual ao que já vimos acima, exceto para as estimativas, e eu explicar o que eles significam em detalhes no arquivo de códigos. No entanto, uma vez que a compreensão dos contrastes de tratamento será muito importante para as regressões logísticas, quero comentá-los aqui também. Existem duas regras centrais que, uma vez internalizadas, ajudam a entender facilmente todos os resultados de contraste de tratamento:

(61) Cada coeficiente/estimativa para um preditor X (efeito principal, interação ou nível de fator) é o valor que você deve adicionar à interpretação para,

- a no caso de variáveis categóricas, prever o valor para o nível de X que você está considerando;
- b no caso de variáveis numéricas, prever o valor que resulta de uma mudança de uma unidade de X;

enquanto que, e este é o ponto crucial, todos os preditores categóricos não mencionados em X são definidos para o seu primeiro nível (geralmente o primeiro nível alfabeticamente, mas também pode ser o que você definir primeiro, como neste caso) e todos os preditores numéricos não mencionados em X são 0.

A segunda regra é apenas um caso especial de (61), ou seja, o intercepto:

(62) Portanto, o intercepto, onde nenhum preditor é mencionado, é o valor previsto quando

a todas as variáveis categóricas na equação do modelo são ajustadas para seu primeiro nível;

e/ou(!)

b todas as variáveis numéricas são ajustadas para zero (o que, se você as centralizou ou as padronizou segundo z , corresponde à sua média)

Então,

- o intercepto é o RT previsto, quando ambos os preditores são ajustados para seus primeiros níveis (L0);
- o segundo coeficiente é o que você adiciona à interceptação para prever o RT para quando o preditor mencionado muda para o nível mencionado aqui (isto é, a IMAGEABILITY muda de L0 para HI) e quando o preditor não mencionada aqui permanece no nível da interceptação (isto é, FAMILIARITY permanece L0);
- o terceiro coeficiente é o que você adiciona à interceptação para prever o RT para quando o preditor mencionado muda para o nível mencionado aqui (isto é, a FAMILIARITY muda de L0 para MED) e quando o preditor não mencionada aqui permanece no nível da interceptação (isto é, IMAGEABILITY permanece L0), similarmente para o quarto coeficiente;
- o quinto coeficiente é para um preditor que é uma interação. Assim, para utilizá-lo para uma previsão, você não apenas adiciona esta estimativa à interceptação, mas também as estimativas dos efeitos principais que fazem parte dela. Assim, para prever o RT para quando a IMAGEABILITY é HI e a FAMILIARITY é MED, você adiciona ao intercepto o segundo coeficiente (para quando IMAGEABILITY é HI), ao terceiro coeficiente (para quando a FAMILIARITY é MED), e este quinto (para a interação):

Código em R

```
> 676.30 + (-26.11) + (-58.94) + 18.91¶
[1] 610.16
```

Compare isso com `preds.hyp[4, ]`: o resultado é o mesmo, e a mesma lógica se aplica ao sexto coeficiente. É provavelmente óbvia agora porque inspecionar `preds.hyp` e gerar um gráfico dos valores previstos é mais fácil do que esmiuçar a tabela de coeficientes, especialmente porque `preds.hyp` é a base para a geração do gráfico, o que você fez acima.

Agora você pode resumir o `model.01` como antes: estatísticas globais do modelo, preditores e seus valores- p , e um gráfico.

5.2.5 Um modelo linear com um preditor categórico e um numérico

Na última seção, ambas as variáveis eram categóricas, de modo que todos os efeitos foram (ajustes às) médias. Agora vamos passar para variáveis mistas: uma variável é categórica (FAMILIARITY), uma é numérica (FREQUENCY). Primeiro, contrastes de soma:

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))
> model.01=lm(RT~FAMILIARITY*FREQUENCY, data=RTs)
> summary(model.01)
> confint(model.01)
```

O modelo é muito significativo ($p = 0,001728$), mas como antes você não obtém todos os valores- p para todos os preditores: você pode ver que FREQUENCY é significativo, mas você não obtém um valor- p para FAMILIARITY e para a interação. Portanto:

Código em R

```
> drop1(model.01, test="F")
> Anova(model.01, type="III")
```

Ambos mostram que a interação não é significativa. Sobre as estimativas:

Código em R

```
> preds.hyp=expand.grid(FAMILIARITY=levels(FAMILIARITY),
  FREQUENCY=floor(min(FREQUENCY)):ceiling(max(FREQUENCY)));
preds.hyp[c("PREDICTIONS", "LOWER", "UPPER")]<-
  predict(model.01, newdata=preds.hyp, interval=
    "confidence"); preds.hyp
```

Esta estrutura de dados não é fácil de processar. (Dada a extensão desta tabela, eu mostro como criar uma versão mais fácil de processar no arquivo de códigos.) Pode-se observar que, à medida que FREQUENCY sobe, os RTs previstos caem, mas realmente o que é necessário é um gráfico. Mas primeiro uma pergunta: o que a interação representa e, portanto, como isso tem que ser gerado em um gráfico?

Pausa para pensar

Como na Seção 5.1.2, a interação de uma variável categórica e numérica significa que não há um declive para o efeito da variável numérica no modelo, mas o mesmo número de declives que o número de níveis dessa variável categórica. Isto é, a interação reflete ajustes a declives. Assim, geramos um gráfico que tem diferentes linhas de regressão para FREQUENCY para cada nível de FAMILIARITY. Os níveis de FAMILIARITY são representados pelas suas primeiras letras. O gráfico aqui é bastante minimalista (por exemplo, ele não inclui os pontos de dados originais), mas o arquivo de códigos fornece uma variedade de alternativas; a mais simples de todas é, como de costume, o gráfico de efeitos.

O gráfico mostra os resultados mais eficientemente do que qualquer outra coisa (especialmente quando os intervalos de confiança são adicionados para mostrar que a interação não é significativa). O modelo mostra que, em geral, a FREQUENCY acelera o tempo de resposta dos participantes, mas especialmente quando a FAMILIARITY é LO quando comparada com quando ela não é. Isto, juntamente com os valores- p etc., deve estar no seu resumo do modelo.

Agora, novamente, vamos olhar rapidamente para os contrastes de tratamento:

Figura 67 – A interação FAMILIARITY:FREQUENCY no model.01

Código em R

```
> options(contrasts=c("contr.treatment", "contr.poly"))
> model.01=lm(RT~FAMILIARITY*FREQUENCY, data=RTs)
> summary(model.01)
> confint(model.01)
> drop1(model.01, test="F")
```

Como de costume, são os coeficientes que mudam, e eles mudam de acordo com as regras em (61) e (62):

- a intercepção é o RT previsto quando todos os preditores são ajustados para seu primeiro nível ou 0, isto é, quando FAMILIARITY é LO e FREQUENCY é 0;
- o segundo coeficiente é o que você adiciona à intercepção para prever o RT para quando o preditor mencionado muda para o nível mencionado aqui (isto é, a FAMILIARITY muda de LO para MED) e quando o preditor não mencionada aqui permanece no nível do intercepto (i.e. FREQUENCY permanece 0), similarmente para o terceiro coeficiente;
- o quarto coeficiente é o que você adiciona ao intercepto para prever o RT para quando o preditor mencionado aumenta em uma unidade (uma vez que a FREQUENCY é numérica, muda de 0 para 1) e quando o preditor não mencionada aqui permanece no nível da intercepção (isto é, a FAMILIARITY permanece LO);
- o quinto coeficiente é para um preditor que é uma interação. Assim, para utilizá-lo para uma previsão, você não apenas adiciona esta estimativa à intercepção, mas também as estimativas para os efeitos principais que fazem parte dela. Assim, para prever o RT para quando FAMILIARITY é MED, e para quando FAMILIARITY aumenta em 1, você adiciona ao intercepto o segundo coeficiente (para quando a FAMILIARITY é MED), o quarto coeficiente (para quando a FAMILIARITY aumenta em 1), e este coeficiente (para a interação):

Código em R

```
> 697.96 + (-66.40) + (-32.73) + 21.65¶
[1] 620.48
```

Compare isso com `preds.hyp[5, ]`; o mesmo para o sexto coeficiente.

5.2.6 Um modelo linear com dois preditores numéricos

Agora estamos chegando na parte séria, chega de jogos e diversão. Vamos modelar REACTTIME como uma função de duas variáveis numéricas, FREQUENCY e MEANINGFULNESS (Significado), e sua interação. Isso é um pouco complicado por causa da interação. Uma interação entre duas variáveis categóricas reflete ajustes nas médias, uma interação entre uma variável categórica e uma variável numérica reflete ajustes nas inclinações - mas o que é uma interação entre duas variáveis numéricas? Como você verá, essa interação significa que o efeito da inclinação de uma variável numérica varia ao longo da faixa da outra variável numérica, o que também significa que às vezes teremos que considerar gráficos tridimensionais: um preditor no eixo x, outro preditor no eixo y, a previsão no eixo z.

Com apenas dois preditores numéricos, não precisamos distinguir entre soma e contrastes de tratamento, então vamos começar. (Você também pode colar as linhas para `drop1` e `Anova`, mas elas são desnecessárias: cada preditor tem 1 grau de liberdade.)

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))¶
> model.01=lm(RT~MEANINGFULNESS*FREQUENCY, data=RTs)¶
> summary(model.01)¶
> confint(model.01)¶
```

Um modelo quase significativo - embora nenhum preditor seja significativo, o que é um pouco raro. Geramos `preds.hyp`, o que desta vez é um pouco mais trabalhoso. Como temos duas variáveis numéricas, geramos intervalos de valores para ambas. Para FREQUENCY fazemos isso como antes. Para MEANINGFULNESS eu não uso apenas oito valores (uma escolha arbitrária, poderia também ser 20) do intervalo atestado, mas também 0 e 1 (para que eu possa explicar os coeficientes).

Código em R

```
> preds.hyp=expand.grid(MEANINGFULNESS=c(0:1,
seq(floor(min(MEANINGFULNESS, na.rm=TRUE)),
ceiling(max(MEANINGFULNESS, na.rm=TRUE)), length.out=8)),
FREQUENCY=floor(min(FREQUENCY)):ceiling(max(FREQUENCY)))¶
> preds.hyp[c("PREDICTIONS", "LOWER", "UPPER")]=predict( model.01, new-
data=preds.hyp, interval="confidence")¶
> preds.hyp¶
```

De fato, os coeficientes significam o que eles sempre significam; veja (61) e (62):

- o intercepto é o RT previsto quando tanto MEANINGFULNESS quanto FREQUENCY são 0;
- o segundo coeficiente é o que você adiciona ao intercepto para prever o RT para quando o preditor mencionado aumenta em uma unidade (isto é, quando MEANINGFULNESS aumenta de 0 para 1) e quando o preditor não mencionado aqui permanece no nível do intercepto (isto é, FREQUENCY permanece 0);

- o terceiro coeficiente é o que você adiciona ao intercepto para prever o RT para quando a FREQUENCY aumenta de 0 para 1 e para quando MEANINGFULNESS permanece no nível da intercepção (isto é, permanece 0);
- o quarto coeficiente é para um preditor que é uma interação. Assim, para usá-lo para uma previsão, você não apenas adiciona esta estimativa à intercepção, mas também as estimativas dos efeitos principais que fazem parte dela. Assim, para prever o RT para quando MEANINGFULNESS é 1 e FREQUENCY é 1, você adiciona ao intercepto todos os coeficientes.

Agora, em um caso real, você não teria adicionado às previsões valores que se baseiam em valores de MEANINGFULNESS que fossem tão distantes dos valores reais, o que também afeta o gráfico. Portanto, geramos uma estrutura de dados `preds.hyp.for.plot` com um grande número de previsões, a dizer, todas as previsões baseadas em todas as combinações de 100 valores de MEANINGFULNESS e 100 valores de FREQUENCY, como mostrado no arquivo de códigos (observe o uso de `seq(..., length.out= ...)` e o uso de `na.rm=TRUE` para se certificar de que `min` e `max` não têm problemas com os dados que estão faltando).

Agora você tem várias possibilidades. As duas primeiras mostradas nos códigos envolve algo que eu realmente não posso demonstrar bem em um livro: A função `plot3d` gera gráficos rotativos de 3 dimensões - você pode clicar no gráfico e mover o mouse para mudar o sistema de coordenadas - e o argumento `col` usa a função `grey` (veja `?grey`) para tornar a escuridão dos pontos dependentes da altura do valor previsto. Normalmente, você tem que manipular muito o gráfico antes que você possa ver o que está acontecendo nos dados. No entanto, recomendo que você deixe os valores previstos no eixo vertical na maioria das vezes. (Um gráfico alternativo mostra que você pode usar qualquer escala de cores que desejar.)

Embora sejam muito úteis para interpretar os dados, geralmente não é possível publicar tais gráficos. Assim, às vezes você não pode representar os valores previstos em 3 dimensões, mas usando cores ou símbolos. O seguinte gráfico é um gráfico de dispersão com MEANINGFULNESS e FREQUENCY nos eixos x e y, respectivamente, e o tamanho do valor previsto é representado pela claridade: quanto mais claro for o cinza, mais lentos os participantes da pesquisa são.

No geral, mas especialmente quando MEANINGFULNESS é baixo, à medida que FREQUENCY aumenta, o RT previsto diminui - veja como na metade esquerda do gráfico o cinza fica mais escuro à medida que você sobe. Também no geral, mas especialmente quando FREQUENCY é baixa, à medida que MEANINGFULNESS aumenta, o RT previsto diminui - veja como na parte inferior do gráfico o cinza fica mais escuro à medida que você vai para a direita. No entanto, e esta é a ligeira sugestão de uma interação (e indicada pela leve inclinação para cima no gráfico tridimensional), quando ambos MEANINGFULNESS e FREQUENCY tornam-se muito elevados, não obtemos os RTs mais rápidos:

No canto superior direito, os pontos não são os mais escuros. Mas, o `model.01` mostrou que este pedaço de uma interação não é de fato significativo, o que você também teria que dizer na sua seção de resultados/discussão.

Outras possibilidades gráficas a serem exploradas são exemplificadas no arquivo de códigos, incluindo um gráfico de efeitos. Uma delas usa números como símbolos de gráfico e mostra bem como as previsões mudam no espaço abrangidos por MEANINGFULNESS e FREQUENCY.

5.2.7 Processo de seleção de um modelo linear com múltiplos preditores

Até agora, ignoramos duas coisas. Em primeiro lugar, por razões expositivas, ignoramos a lâmina de Occam: quando um preditor - um efeito principal ou uma interação - não foi significativo, deixamos-o no modelo e o incluímos no gráfico. Nesta Seção, vamos ver como fazer um processo de seleção reverso.

Em segundo lugar, ignoramos os testes dos pressupostos da regressão, então vamos também falar sobre isso um pouco no final. O modelo máximo que vamos explorar aqui envolve todas as variáveis independentes que você já viu até agora incluindo todas as suas interações até (e inclusive) interações de 3 vias; deixe-me esclarecer que isto só pode ser um exemplo didático, já que o número de preditores é muito alto em comparação com o pequeno número de dados:

Figura 68 – A interação MEANINGFULNESS : FREQUENCY no model.01

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))
> model.01=lm(RT~(FREQUENCY+FAMILIARITY+IMAGEABILITY+
MEANINGFULNESS)^3, data=RTs[complete.cases(RTs),])
> summary(model.01)
```

Observe como definimos o argumento data para garantir que somente casos completos sejam inseridos no processo. Observe também a sintaxe para dizer que queremos incluir efeitos principais, interações de 2 vias e de 3 vias: as variáveis estão entre parênteses e depois dizemos ^3.

Os resultados mostram um modelo no geral insignificante, com alguns preditores significantes e muitos preditores insignificantes. Parte da razão pela qual o modelo geral é significativo é porque o grande número de preditores insignificantes aumenta os graus de liberdade, o que torna mais difícil obter um resultado significativo. Observe, neste contexto, a enorme diferença entre R^2 múltiplo e R^2 ajustado. Além disso, temos um problema que é bastante comum com dados reais: uma célula no nosso delineamento tem apenas uma observação - a combinação de FAMILIARIDADE:HI e IMAGINABILIDADE:LO - o que leva a NAs nos coeficientes, o que, por sua vez, faz com que a função Anova não funcione.

Código em R

```
> Anova(model.01, type="III")
```

Há três maneiras de lidar com isso. A maneira provavelmente melhor é usar drop1, que, como de costume, irá testar para cada preditor que poderia ser omitido nesta fase se a sua eliminação tornaria o modelo significativamente pior:

Código em R

```
> drop1(model.01, test="F")
```

Como você pode ver, tal como discutido na Seção 5.1.2.2, `drop1` testa somente as interações de ordens mais altas, e aquela com o maior valor-*p* seria a melhor para ser excluída primeiro: `FREQUENCY:FAMILIARITY:IMAGEABILITY`.

Uma segunda possibilidade é adicionar um argumento em `Anova`, o que fornece o mesmo resultado e conclusão sobre qual interação excluir primeiro:

Código em R

```
> Anova(model.01, type="III", singular.ok=TRUE)
```

A última possibilidade seria a mais trabalhosa. Trata-se de identificar as quatro interações que poderiam ser excluídas, calcular quatro modelos que se diferem do `model.01` apenas por perderem uma dessas interações - ou seja, o modelo menor é um sub-modelo do maior! - e, em seguida, fazer uma comparação entre os modelos para verificar quão pior o modelo menor é. Depois que isso é feito para todas as quatro interações candidatas a serem excluídas, você deleta aquela para a qual o maior valor-*p* não significativo foi obtido.

A primeira dessas etapas, a de gerar um sub-modelo, é feita de forma melhor com a atualização de função `update`. O primeiro argumento dessa função é o modelo que você deseja alterar, seguido de `~.`, seguido do que você deseja fazer, como, por exemplo, “aqui subtraia um preditor”: (Eu só mostro as duas primeiras atualizações, você também deve explorar a ajuda para `update`, o qual pode ser usado de outras maneiras úteis.)

Código em R

```
> model.02a=update(model.01, ~. -FREQUENCY:FAMILIARITY:IMAGEABILITY)
> model.02b=update(model.01, ~. -FREQUENCY:FAMILIARITY:MEANINGFULNESS)
```

Em seguida, você compara o primeiro modelo completo com esses sub-modelos usando a função `anova` (a minúsculo!): (Novamente eu só mostrar os dois primeiros.)

Código em R

```
> anova(model.01, model.02a)
> anova(model.01, model.02b)
```

No final, você obtém a interação que deve ser excluída primeiro. Para deletar essa interação agora, você usa novamente a função `update` e agora define `model.02` como o `model.01` sem `FREQUENCY:FAMILIARITY:IMAGEABILITY`:

Código em R

```
> model.02=update(model.01, ~. -FREQUENCY:FAMILIARITY:IMAGEABILITY)
```

Este processo é agora repetido quantas vezes for necessário e como mostrado no arquivo de códigos. Você, obviamente, só precisa executar uma das alternativas mostradas lá. Um comentário: `drop1` às vezes já retornará valores-*p* para a exclusão de preditores de graus menores de interatividade do que o que você está verificando. Vamos nos ater ao acima e apenas ir para um nível mais baixo de interatividade, ou

para interações de ordem inferior, se não sobrar nenhuma interação de ordem superior para ser apagada; veja a sequência no arquivo de códigos.

Depois de alguns testes, você chega ao `model.14`, o qual, seguindo a lâmina de Occam, contém apenas a FAMILIARITY como preditor - tudo mais teve que ser jogado fora.

Código em R

```
> summary(model.14)
[...]
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	619.049	7.145	86.636	< 2e-16 ***
FAMILIARITY1	33.201	11.619	2.858	0.00644 **
FAMILIARITY2	-5.541	8.445	-0.656	0.51512

```
--
Multiple R-squared: 0.1711, Adjusted R-squared: 0.1343
F-statistic: 4.645 on 2 and 45 DF, p-value: 0.01465
[...]
```

Mas não terminamos. A FAMILIARIDADE tem três níveis, mas talvez não precisarmos de todos eles, algo que já foi sugerido acima por `tukeyHSD(aov(...))`. Continuamos, portanto, com a comparação de modelos - não mais testando para descartar variáveis, mas agora níveis das variáveis. Seguindo a lógica de [Crawley \(2007, p. 563\)](#), criamos dois fatores novos, cada um dos quais combina dois níveis adjacentes, e os adicionamos à nossa estrutura de dados (para garantir que testaremos o mesmo número de casos), e então calculamos dois novos modelos, um com cada versão combinada de FAMILIARITY, e, em seguida, fazemos comparações dos modelos com anova:

Código em R

```
> FAMILIARITY.conflat1=FAMILIARITY.conflat2=FAMILIARITY
> levels(FAMILIARITY.conflat1)=c("lo", "med-hi", "med-hi")
> levels(FAMILIARITY.conflat2)=c("lo-med", "lo-med", "hi")
> RTs<-cbind(RTs, FAMILIARITY.conflat1=FAMILIARITY.conflat1,
FAMILIARITY.conflat2=FAMILIARITY.conflat2)
```

Código em R

```
> model.15a=lm(RT~FAMILIARITY.conflat1, data=RTs[complete.cases(RTs),])
> model.15b=lm(RT~FAMILIARITY.conflat2, data=RTs[complete.cases(RTs),])
> anova(model.14, model.15a)
> anova(model.14, model.15b)
```

Os resultados mostram que a primeira fusão - aquela que também teve o maior valor-*p* no teste TukeyHSD - não torna o modelo significativamente pior, enquanto que a segunda sim. Portanto, a Figura 69 é como o modelo final pode ser resumido agora (veja os códigos e `?plotmath` para saber como o cabeçalho principal pode ter itálico, sobrescritos, etc.).

Permitam-me, neste ponto, interromper brevemente a discussão sobre este modelo e voltar a um ponto mais geral. Neste caso, temos apenas um efeito principal significativo, e nas seções acima discutimos como gerar gráficos com interações entre duas variáveis. Às vezes, usuários do R levantam então a seguinte questão, "ok, mas eu tenho uma interação significativa de três variáveis - como eu consigo gerar um gráfico disso?"

Figura 69 – Único efeito significativo final do `model.15`

A resposta usual é que você deve gerar o gráfico com base nos gráficos acima. Por exemplo, se você tiver uma interação de três variáveis categóricas X, Y e Z, então você faz gráficos para X:Y como discutido na Seção 5.2.4 para cada nível de Z (e idealmente você experimenta diferentes configurações para ver qual gráfico é o mais fácil de interpretar). Por exemplo, imagine uma interação de duas variáveis categóricas X e Y e uma variável numérica Z. Nesse caso, você pode gerar um gráfico de X:Z como discutido na Seção 5.2.5 para cada nível de Y (ou Y:Z para cada nível de X), etc. Isto é, você apenas pega os gráficos discutidos acima e os usa como fundamento para interações de ordens superiores.

Uma questão relacionada e muito importante é como obter algo como `preds.hyp` para um preditor X em um modelo quando X não é o único preditor deixado no modelo. Em tais cenários, a abordagem com `preds.hyp` que vimos acima não é ideal quando outro preditor no modelo, digamos, Y, tem níveis cujas frequências diferem-se enormemente entre si, o que acontece frequentemente com dados reais. Por exemplo, o `model.10` no processo de seleção do modelo envolve a seguinte fórmula:

Código em R

```
> formula(model.10)
RT ~ FREQUENCY + FAMILIARITY + IMAGEABILITY + MEANINGFULNESS + IMAGEABILITY:MEANINGFULNESS
```

Se você deseja extrair os valores previstos para FAMILIARITY, então você pode usar a função `effect` para criar uma lista chamada, digamos, `fam`:

Código em R

```
> fam = effect("FAMILIARITY", model.10); fam
lo      med      hi 653.7191 613.6080 606.0162
```

Embora este resultado seja exatamente o que você precisa, obter esses números (talvez até com intervalos de confiança) não é tão fácil quanto parece. Você tem que saber que

- as variáveis de preditores que criamos com `expand.grid` estão em `fam$x`;
- os valores previstos estão agora em `fam$fit`;
- os limites inferiores do intervalo de confiança estão em `fam$lower`;
- os limites superiores do intervalo de confiança estão em `fam$upper`.

Como você usa isso para criar algo como `preds.hyp` para a interação? Confira o arquivo de códigos para ver como isso é feito.

Voltemos ao `model.15`. A última coisa a ser feita antes de explicar o processo de seleção do modelo que você fez e resumir os resultados é verificar os pressupostos do modelo. Isso pode ser feito de muitas maneiras, mas duas formas práticas são: (i) inspecionar alguns gráficos de diagnóstico de modelo e (ii) usar a função `gv1ma` do pacote com o mesmo nome para obter uma visão geral rápida.

Código em R

```
> par(mfrow=c(2, 2))
> plot(model.15)
> par(mfrow=c(1, 1))
```

Os dois gráficos à esquerda testam as suposições de que as variâncias dos resíduos são constantes. Ambos mostram a proporção dos valores ajustados/previstos no eixo x em relação aos tipos de resíduos no eixo y. Idealmente, ambos os gráficos mostrariam uma nuvem de dispersão sem muita estrutura. Aqui temos apenas dois valores ajustados (um para cada nível de `FAMILIARITY.confla1`), mas nenhuma estrutura tal que a dispersão dos valores aumenta ou diminui da esquerda para a direita: aqui, esses gráficos estão ok.⁷ Várias palavras são marcadas como *outliers* potenciais. Além disso, o gráfico na parte superior esquerda mostra que os resíduos são distribuídos bem em torno da média desejada de 0.

A pressuposição de que os resíduos são distribuídos normalmente também parece ser atendida: Os pontos no gráfico superior direito deveriam estar bastante próximos da linha tracejada, e estão. Novamente, três palavras são marcadas como *outliers* potenciais. Mas você também pode fazer um teste Shapiro-Wilk sobre os resíduos, o que também produz o resultado esperado.

Finalmente, o gráfico do lado inferior direito mostra os resíduos padronizados mediante ao chamado alavancagem. A alavancagem é uma medida de quanto um ponto de dados pode influenciar um modelo (porque está muito distante do centro da variável independente relevante). Como você pode ver, existem algumas palavras com uma alavancagem maior, e estes são todos casos de `FAMILIARITY:L0`, que neste conjunto de dados fictício é um número muito menor de dados. Permita-me também mostrar brevemente um exemplo de gráficos de diagnóstico de modelo que apontam violações dos pressupostos do modelo. A Figura 71 abaixo mostra os dois gráficos de modelo acima que uma vez eu encontrei quando explorava os dados um estudante que tinha sido aconselhado (por um consultor de estatística!) a aplicar um modelo linear do tipo ANOVA em seus dados. No painel à esquerda, você pode ver claramente como o intervalo de resíduos aumenta da esquerda para a direita. No painel à direita, você pode ver quão fortemente os pontos se desviam da linha tracejada, especialmente na parte superior direita do sistema de coordenadas. Tais gráficos são um alerta claro (e a função `gv1ma` mencionada acima mostrou que quatro das cinco

⁷ Você também pode usar `ncvTest` da biblioteca `car`: `ncvTest(model.15)`, que retorna o resultado não significativo desejado.

Figura 70 – Diagnóstico do modelo para model . 15

Figura 71 – Diagnóstico problemático do modelo

pressuposições testadas foram violadas!). Um possível passo seguinte da pesquisa seria verificar se é possível ignorar justificadamente os *outliers* indicados; veja [Fox e Weisberg \(2011, Cap. 6\)](#) para discussão.

Recomendação para estudo/leitura

- para seleção de modelo:
 - a função `step`, para que o R execute a seleção do modelo automaticamente baseado no AIC (Coeficiente de Informação de Akaike) (por padrão) e a função `stepAIC` (da biblioteca MASS). Aviso: processos automáticos de seleção de modelos podem ser perigosos: diferentes algoritmos podem resultar em resultados muito diferentes
- sobre diagnóstico de modelos:
 - as funções `residualPlots` e `marginalModelPlots` e `vif` da biblioteca `car`, as duas primeiras como alternativas para `plot(model)`, a última para testar a colinearidade, a qual é muito importante para explorar; é a ameaça representada por variáveis preditoras altamente intercorrelacionadas; veja Faraway (2005: Seções 5.3 e 9.3, Fox e Weisberg 2011: Capítulo 6)
 - as funções `influence.measures` e outras funções mencionadas neste arquivo de ajuda (especialmente `dfbeta`) para identificar pontos de alavancagem e *outliers*
- as funções `oneway.test` e `kruskal.test` como alternativas a ANOVAs monofatoriais
- as bibliotecas `robust`, `MASS`, e `nls` para regressões robustas e não lineares; veja especialmente [Crawley \(2002\)](#), [Crawley \(2005\)](#) e [Faraway \(2005\)](#), [Faraway \(2006\)](#)
- a função `rpart` da biblioteca `rpart`, e a função `ctree` de biblioteca `party`, para calcular árvores de classificação e de regressão como alternativas a modelos lineares (generalizados)
- [Harrell \(2001\)](#), [Crawley \(2002, Cap. 13-15, 17\)](#), [Faraway \(2005, Cap. 14\)](#), [Crawley \(2007, Cap. 10-12, 14, 16\)](#), [Gelman e Hill \(2007, Cap. 3-4\)](#), [Baayen \(2008, Seção 4.4, Cap. 6-7\)](#), [Johnson \(2008, Seções 2.4 e 3.2, Cap. 4\)](#), [Zuur et al. \(2009, Cap. 2-4, 6-7\)](#), [Fox e Weisberg \(2011\)](#), [Baguley \(2012, Cap. 5, 12-15\)](#)

5.2.8 Modelos de regressão logística binária

Na seção anterior, tratamos de métodos lineares, nos quais a variável dependente é de escala de intervalo/razão e abrange uma ampla gama de valores. No entanto, em muitas situações a variável dependente é binária, categórica ou numérica, mas talvez apenas ≥ 0 e/ou discreto (como para frequências) ou ... Uma vez que o modelo linear “normal” discutido acima prediz valores entre $-\infty$ e $+\infty$, ele irá prever valores que não fazem muito sentido para tais variáveis dependentes - o que um valor previsto de -3,65 significaria quando você tenta prever frequências de algo? Para situações como essas, outros modelos são usados, alguns deles recebendo o título de modelos lineares generalizados, levando a tipos de regressão como as seguintes:

- regressão logística binária para variáveis dependentes binárias;
- regressão logística ordinal e regressão multinomial para variáveis dependentes ordinais e categóricas, respectivamente;
- regressão de Poisson/contagem para frequências como variáveis dependentes.

Para ser possível aplicar uma abordagem de modelagem linear a tais dados, a variável dependente é transformada com uma chamada função de ligação (link function), a qual transforma o intervalo previsto de valores de um modelo linear ($-\infty$ a $+\infty$) em um intervalo mais apropriado para a variável dependente. Para a regressão logística binária, por exemplo, a transformação *logit* inversa em (63a) transforma valores do intervalo de $-\infty$ a $+\infty$ em valores variando de 0 a 1, os quais podem então ser interpretados como probabilidades de um evento previsto. Para a regressão de Poisson, a transformação exponencial em (64a) transforma valores do intervalo de $-\infty$ a $+\infty$ em valores variando de 0 a $+\infty$. As funções em (63b) e (64b) fazem a transformação na direção oposta:

(63)

a logit inverso de x : $\frac{1}{1+e^x}$

b logit de x : $\log\left(\frac{x}{1-x}\right)$

(64)

a função exponencial de x : e^x

b função logarítmica de x : $\ln(x)$

Portanto, há boas notícias e más notícias ... A má notícia é que a regressão logística binária não é fácil de entender, por causa da forma como a função de ligação transforma a variável dependente e também porque, como você verá mais adiante, existem três maneiras diferentes pelas quais se pode relatar resultados de tal regressão, o que torna difícil entender como livros ou artigos explicam métodos/resultados. A boa notícia é que, uma vez que você já tiver passado pela função de ligação, o resto é praticamente igual ao que vimos acima, e nesta seção podemos trabalhar o tempo todo com os contrastes de tratamento padrão do R - não há necessidade de dois tipos de contrastes.

O conjunto de dados que vamos explorar envolve a pergunta sobre como uma oração principal e uma subordinada são ordenadas em uma sentença. Essa pergunta envolve as seguintes variáveis:

- uma variável binária dependente, a dizer ORDER: MC-SC versus SC-MC indicando se a oração principal precede a oração subordinada ou não;
- uma variável binária independente SUBORDTYPE: CAUS vs. TEMP indicando se a oração subordinada é causal ou temporal;
- duas variáveis numéricas independentes LENGTHMC e LENTHSC, representando o número de palavras da oração principal e da subordinada, respectivamente;

- uma variável numérica independente LENGTHDIFF, representando o resultado da diferença entre a extensão da oração principal e a extensão da oração subordinada;
- uma variável independente categórica CONJ, que representa a conjunção usada na oração subordinada. Uma vez que estes dados são de um estudo envolvendo dados de corpus paralelos, estes são os níveis: ALS/WHEN (quando), BEVOR/BEFORE (antes), NACHDEM/AFTER (depois) e WEIL/BECAUSE (porque);
- uma variável binária independente MORETHAN2CL : NO vs. YES indicando se há ou não mais do que a oração principal e a subordinada na sentença. Isto pode ser entendido como uma questão de saber se a sentença envolve mais complexidade do que apenas duas orações.

Uma regressão logística binária envolve o seguinte procedimento:

Procedimentos

- Formular as hipóteses
- Importar os dados, prepará-los para a modelagem e explorá-los
- Calcular, selecionar e interpretar o modelo de regressão logístico
 - obter os valores- p para todos os preditores e para o modelo como um todo
 - interpretar os coeficientes / estimativas da regressão baseando (i) nos valores preditos e (ii) o gráfico das probabilidades preditas
- Testar as pressuposições principais dos testes:
 - independência dos dados e dos resíduos, ausência de pontos excessivamente influentes, ausência de multicolinearidade e sobredispersão
 - menos que 95% dos resíduos padronizados do modelo > 2 ;
 - pouco ou nenhum dos graus de liberdade ($g1betas$ absolutos de qualquer caso e preditor > 1)

Primeiramente, as hipóteses:

H_0 : Não há correlação entre ORDER e os preditores (variáveis independentes e suas interações): R^2 de Nagelkerke = 0.

H_1 : Há correlação entre ORDER e os preditores (variáveis independentes e suas interações): R^2 de Nagelkerke > 0 .

Em seguida você carrega os dados de `<_inputfiles/05-3_clauseorders.csv>`:

Código em R

```
> CLAUSE.ORDERS=read.delim(file=file.choose())¶
> summary(CLAUSE.ORDERS); attach(CLAUSE.ORDERS)¶
```

Neste caso, não será efetuada qualquer outra preparação dos dados, e é por isso que a estrutura de dados já foi anexada. No entanto, queremos escrever duas funções auxiliares (e definir `error.bar` novamente como acima), carregar alguns pacotes, e garantir que estamos usando contrastes de tratamento:

Código em R

```
> logit=function(x) log(x/(1-x)) ¶
> ilogit=function(x) 1/(1+exp(-x)) ¶
> options(contrasts=c("contr.treatment", "contr.poly"))¶
```

A exploração dos dados com tabulações cruzadas e spineplots de variáveis em relação a ORDER não levanta qualquer sinal de alerta, então vamos em frente. Nesta seção, vou mostrar relativamente poucos códigos/gráficos no livro, portanto que siga as explicações com o arquivo de códigos!

5.2.9 Regressão logística com preditor binário

Nesta seção, consideraremos se ORDER está correlacionada com SUBORDTYPE. Como antes, esta primeira seção será mais longa do que aquelas que a seguem para estabelecer as bases para as questões mais complexas que aparecerão mais tarde.

Assim como um modelo linear com um preditor binário se reduz a um teste mais simples que já conhecemos - o teste t - uma regressão logística com um preditor binário se relaciona a um teste mais simples: o teste qui-quadrado, uma vez que só temos de fato duas variáveis binárias (como na Seção 4.1.2.2):

Código em R

```
> orders=table(SUBORDTYPE, ORDER); orders¶
ORDER
SUBORDTYPE mc-sc sc-mc
caus      184    15
temp      91    113
> chi.orders=chisq.test(orders, correct=FALSE); chi.orders¶
Pearson's Chi-squared test
data: orders
X-squared = 106.4365, df = 1, p-value < 2.2e-16
```

Dois breves comentários: Primeiro, lembre-se das noções de probabilidades e de proporções/razões de chances da Seção 4.1.2.2. Veja como a partir desta tabela você calcularia asb probabilidades de MC-SC (do inglês main clause - subordinate clause, oração principal e oração subordinada) primeiro com orações subordinadas causais, e depois com orações subordinadas temporais. Além disso, também dissemos que você pode calcular uma razão de chances a partir disso, e que às vezes você verá o logaritmo da razão de chances.

Código em R

```
> (184/199) / (15/199)¶
[1] 12.26667
> (91/204) / (113/204)¶
[1] 0.8053097
> 12.26667/0.8053097¶
[1] 15.23224
> log(15.23224)¶
[1] 2.723414
```

Finalmente, você pode naturalmente expressar o fato de que, obviamente, as orações subordinadas causais preferem aparecer depois da oração principal, enquanto que as orações subordinadas temporais

preferem preceder a cláusula principal com as seguintes porcentagens: 92,46% de todas as orações subordinadas causais, mas apenas 44,61% das orações subordinadas temporais, são usadas depois da cláusula principal. Em outras palavras, temos três maneiras diferentes, mas naturalmente relacionadas, para falar sobre este resultado - probabilidade/chance, logaritmo das chances, e porcentagens/probabilidades - algo que vou voltar a mencionar brevemente.

O segundo comentário tem a ver com uma alternativa para χ^2 . A regressão logística não usa um valor- χ^2 como o calculado em um teste- χ^2 , mas um teste chamado de teste de razão de verossimilhança que resulta em um valor-G. G é também χ^2 distribuído e, em uma regressão logística envolvendo apenas uma variável binária/categórica, pode ser computado de forma semelhante a χ^2 ; veja (65) e também o arquivo de códigos para uma pequena demonstração mostrando quão semelhantes χ^2 e G são.

$$(65) \quad G = 2 \sum_{i=1}^n \text{observados} \left(\frac{\text{observados}}{\text{esperados}} \right)$$

Código em R

```
> 2*sum(orders*log(orders/chi.orders$expected))¶
[1] 116.9747
```

Com tudo isso em mente, vamos agora executar uma regressão logística. A função principal é `glm`, para o modelo linear generalizado, e requer uma fórmula e um argumento de dados como antes, mas agora também um argumento que permite que o R infira que você quer usar uma função de ligação para regressão logística binária (Estou simplificando um pouco). Também como antes, os coeficientes que a regressão retornará são estimativas, então solicitamos imediatamente intervalos de confiança com `confint`:

Código em R

```
> model.01=glm(ORDER~SUBORDTYPE, data=CLAUSE.ORDERS,family=binomial)¶
> summary(model.01)¶
[...]
```

Deviance Residuals:				
Min	1Q	Median	3Q	Max
-1.2706	-0.3959	-0.3959	1.0870	2.2739

```

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.5069    0.2685  -9.336 <2e-16 ***
SUBORDTYPEtemp 2.7234    0.3032   8.982 <2e-16 ***
[...]
```

Null deviance: 503.80 on 402 degrees of freedom	
Residual deviance: 386.82 on 401 degrees of freedom	
AIC: 390.82	
[...]	

```

> confint(model.01)¶
2.5%    97.5%
(Intercept) -3.076455 -2.016328
SUBORDTYPEtemp 2.156967  3.352559
```

Semelhante ao resultado de `lm`, mas também diferente. Por exemplo, você não obtém um valor- p global. No entanto, você pode inferir que o modelo é significativo com base no fato de que o único preditor é significativo (e que seu intervalo de confiança não inclui 0). Além disso, na parte inferior você encontra o chamado desvio nulo - informalmente falando, a quantidade de variabilidade geral nos dados

– e o desvio residual - informalmente falando, a quantidade de variabilidade restante nos dados após o preditor ter cuidado de parte da variabilidade - e a diferença entre os dois é G . Como mencionado acima, G é χ^2 -distribuído com grau de liberdade como a diferença entre os graus de liberdade dos desvios (*deviances*), isto é, 1. Assim, o valor- p global do modelo pode ser calculado da seguinte forma:

Código em R

```
> pchisq(503.80-386.82, 402-401, lower.tail=FALSE)
[1] 2.899771e-27
```

Novamente, encontramos estatísticas de resumo sobre os resíduos na parte superior e, novamente, antes de discutirmos as estimativas, executamos códigos que nos ajudariam a obter valores- p para preditores com mais de um grau de liberdade:

Código em R

```
> drop1(model.01, test="LR")
Single term deletions

Model:
ORDER ~ SUBORDTYPE
Df Deviance    AIC    LRT Pr(>Chi)
<none>         386.82  390.82
SUBORDTYPE    1 503.80  505.80 116.97 <2.2e-16 ***
--
[...]
```

```
> anova(model.01, glm(ORDER 1, family=binomial), test="LR")
Analysis of Deviance Table
Model 1: ORDER ~ SUBORDTYPE
Model 2: ORDER ~ 1
Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      401 386.82
2      402 503.80    -1 -116.97 < 2.2e-16 ***
[...]
```

A única alteração para `drop1` é que agora não fazemos um teste F , mas o teste de razão de verossimilhança (com LR). A linha com `anova` faz o mesmo tipo de comparação de modelos: compara o `model.01` com um modelo mínimo onde `ORDER` é apenas regredido sobre uma interceptação global (1) e retorna o mesmo teste de razão de verossimilhança. Podemos também usar `Anova` novamente, só precisamos mudar para contrastes de soma apenas para este teste, e novamente obtemos o resultado habitual:

Código em R

```
> options(contrasts=c("contr.sum", "contr.poly"))
> Anova(model.01, type="III", test.statistic="LR")
Analysis of Deviance Table (Type III tests)
Response: ORDER
LR Chisq Df Pr(>Chisq)
SUBORDTYPE    116.97 1 < 2.2e-16 ***
[...]
```

```
> options(contrasts=c("contr.treatment", "contr.poly"))
```

Como antes, no arquivo de códigos eu forneço alguns códigos extras para obter valores-*p* para preditores/coeficientes usando `wald.test eglht`.

Agora, finalmente, sobre os coeficientes, e uma última vez que geramos `preds.hyp` da mesma forma que com modelos lineares. No entanto, para modelos lineares generalizados, `predict` infelizmente não retorna intervalos de confiança para os valores previstos.

Código em R

```
> preds.hyp=expand.grid(SUBORDTYPE=levels(SUBORDTYPE));
preds.hyp["PREDICTIONS"]=predict(model.01, newdata=preds.hyp); preds.hyp
SUBORDTYPE PREDICTIONS
1      caus  -2.5068856
2      temp   0.2165283
```

Agora, o que isso significa? Obviamente, esta não é nem uma escolha de ordenação nem uma escolha de 0 contra 1 ... Para entender o que esses valores significam, você tem que (i) lembrar as três maneiras diferentes de falar sobre os dados acima: probabilidades/chances e razão de chances, logaritmos da razão de chances e probabilidades, e (ii) você tem que perceber que estes valores previstos são o logaritmo das probabilidades para a ordenação prevista e, por padrão, o R prediz o segundo nível da variável dependente, isto é, aqui *SC-MC* (oração subordinada-oração principal). Uma vez que você sabe disso, você pode usar essas informações para também perceber como as três formas de considerar esses dados estão relacionadas, o que é representado na Figura 72.

Este gráfico representa as três perspectivas sobre os resultados uma ao lado da outra, e representa os possíveis intervalos numéricos das três formas nos eixos y: as probabilidades/chances vão de 0 a $+\infty$, logaritmos das chances vão de $-\infty$ a $+\infty$, e probabilidades vão de 0 a 1. Cada uma dessas perspectivas expressa preferência, dispreferência e falta de efeito em diferentes intervalos. No espaço numérico de probabilidades, nenhuma preferência é 1, no espaço do logaritmo das probabilidades, a preferência é 0, e para probabilidades previstas a preferência é naturalmente 0,5 (uma vez que temos duas opções). Para probabilidades (odds), as preferências são refletidas por probabilidades (odds) maiores que 1, por probabilidades de log positivas e por probabilidades previstas de $>0,5$, e os opostos refletem dispreferências.

Agora, se as previsões acima são o logaritmo das probabilidades, podemos transformá-las para nos ajudar a reconhecer o que elas significam. Deixe-me novamente mostrar `orders` primeiro.

Código em R

```
> orders
ORDER
SUBORDTYPE mc-sc sc-mc
caus      184    15
temp       91   113
```

Se `preds.hyp` contém o logaritmo das probabilidades, o inverso do logaritmo/exponenciação então deveria nos dar as probabilidades (veja novamente a Figura 72 e o arquivo de códigos), e é isso que acontece.

Figura 72 – Três formas para checar os resultados de uma regressão logística binária

Código em R

```
> exp(-2.5068856) # odds for SC-MC when SUBORDTYPE=="caus"
[1] 0.08152174
> exp(0.2165283) # odds for SC-MC when SUBORDTYPE=="temp"
[1] 1.241758
```

Isso também significa que dividir as duas nos dá uma razão de chances, o que você também obtém o inverso do logaritmo do coeficiente:

Código em R

```
> exp(-2.5068856)/exp(0.2165283) # 1/odds ratio from above
[1] 0.06565025
> exp(0.2165283)/exp(-2.5068856) # odds ratio from above
[1] 15.23223
> exp(2.7234)
[1] 15.23202
```

Da mesma forma, se `preds.hyp` contém o logaritmo das probabilidades, aplicar `ilogit` deve nos dar probabilidades (veja novamente a Figura 72 e o arquivo de códigos), e é isso que acontece:

Código em R

```
> ilogit(-2.5068856) # prob. of sc-mc when SUBORDTYPE="caus"
[1] 0.07537688
> ilogit(0.2165283) # prob. of sc-mc when SUBORDTYPE="temp"
[1] 0.5539216
```

Finalmente, você também pode obter essas probabilidades a partir das probabilidades usando a equação mostrada na Figura 72 e em (37) na página 186:

Código em R

```
> 0.08152174/(1+0.08152174)
[1] 0.07537689
> 1.241758/(1+1.241758)
[1] 0.5539215
```

Grande parte do que pode ser tão confuso sobre a regressão logística para iniciantes é que os autores de artigos ou livros didáticos podem e usam qualquer uma dessas três perspectivas: elas estão todas certas, mas sem algo como a Figura 72, é difícil ver como elas se relacionam. A questão natural agora é, qual das três escalas é melhor. Como de costume, as pessoas discordam sobre isso, mas vou dizer-lhe qual eu estou usando em meu próprio trabalho e também aqui.

Eu mesmo não gosto da escala de probabilidades à esquerda. O fato de que o espaço numérico para expressar preferência (de, digamos oração subordinada oração-principal, SC-MC) é de 1 a $+\infty$, mas que as preferências correspondentes são “espremidas” no intervalo de 0 a 1 e a natureza multiplicativa desta escala me fazem rejeitá-la fortemente. A escala do logaritmo das probabilidades têm propriedades atraentes: é aditiva e os espaços numéricos de preferência e de rejeição são igualmente grandes e simétricos em torno de 0. O que eu ainda não gosto sobre essa escala é que ela é uma escala de algo tão totalmente não intuitivo quanto aos logaritmos das probabilidades. Portanto, eu prefiro a escala de probabilidade. Posso pensar em termos de probabilidades, e os espaços numéricos para preferência e despreferência são igualmente grandes e simétricos em torno de 0,5. Pode parecer agora que as probabilidades não apresentam nenhuma desvantagem - mas apresentam, o que você aprenderá na Seção 5.3.3 (veja a página 306), mas eu ainda as prefiro. Assim, é a escala mais à direita com a qual vamos trabalhar e plotar aqui.

Vamos voltar às previsões e retrabalhar este exemplo de forma mais simples com probabilidades e também adicionar intervalos de confiança. Primeiro, geramos uma versão de `preds.hyp` como acima, mas desta vez usamos imediatamente a abordagem mais poderosa do pacote `effects`: geramos um objeto com todos os resultados para o efeito relevante (`sot`) e extraímos todas as informações relevantes dele - os níveis do único preditor, os valores previstos e os limites de confiança - e aplicamos `ilogit` aos resultados numéricos:

Código em R

```
> sot=effect("SUBORDTYPE", model.01)
> preds.hyp=data.frame(sotx,PREDICTIONS=ilogit(sotfit),
  LOWER=ilogit(sotlower),UPPER=ilogit(sotupper))
> preds.hyp
  SUBORDTYPE PREDICTIONS    LOWER    UPPER
1      caus  0.07537688 0.0459497 0.1212547
2      temp  0.55392157 0.4851215 0.6207159
```


Novamente, essas são as probabilidades preditas do segundo nível da variável dependente. Assim, o modelo prevê uma baixa probabilidade de *SC-MC*, quando a oração subordinada é causal e uma probabilidade muito maior quando a oração subordinada é temporal. Como temos duas opções, é natural tomar 0,5 como ponto de corte (como na Figura 72) e dizer que quando a probabilidade predita de *SC-MC* é $<0,5$, então o modelo prediz *MC-SC* - caso contrário o modelo prediz *SC-MC*. Podemos usar isso para determinar a eficácia do modelo na previsão das opções de ordem das orações. Primeiro, geramos um vetor que contém uma probabilidade predita de *SC-MC* para cada ponto de dados em nossos dados utilizando `fitted`. Em seguida, usamos `ifelse` para permitir que o R decida para cada probabilidade prevista qual ordem ele prevê. E então tabulamos as escolhas previstas pelo modelo com as escolhas reais e calculamos quantas vezes as duas foram as mesmas:⁸

Código em R

```
> predictions.num=fitted(model.01)
> predictions.cat=ifelse(predictions.num>=0.5,"sc-mc", "mc-sc")
> table(ORDER, predictions.cat)
predictions.cat
ORDER    mc-sc  sc-mc
mc-sc    184    91
sc-mc     15   113
> (184+113)/length(predictions.cat)
[1] 0.7369727
```

Isso parece bom? Com o que você compara isso?

Pausa para pensar

Ao contrário do que você poderia pensar, você não deve compará-lo com uma chance de precisão de 0,5 (porque você tem duas ordens). Por quê? Porque as duas ordens não são igualmente frequentes. A ordem mais frequente, *MC-SC*, representa 68,24% de todos os dados, portanto o simples fato de você inferir isso já lhe garante muito mais do que 50% de acerto. A partir dessa perspectiva (veja o arquivo de códigos para outra), o resultado atual não é grande: *SUBORDTYPE* (tipo de subordinada) só melhora nossa precisão em cerca de 5%.

Vamos agora também visualizar os resultados. Apresento dois gráficos aqui, versões melhores do que você verá quando você executar o código no arquivo de códigos. O painel esquerdo da Figura 73 mostra um gráfico de barras das probabilidades previstas de *SC-MC*. O painel direito mostra um gráfico linear dessas probabilidades.

Código em R

```
> barplot(preds.hyp$PREDICTIONS, ylim=c(0, 1),names.arg=preds.hyp$SUBORDTYPE)
> plot(sot, ylim=c(0, 1), rescale.axis=FALSE)
```

Finalmente, eu quero introduzir uma ferramenta muito útil para todos os tipos de modelagem de regressão, o pacote *rms* e sua função *lrm* (para modelagem de regressão logística). Para usar tudo o que *lrm* tem para oferecer, é útil primeiro executar a primeira linha de código abaixo para que as funções de *rms* possam acessar informações básicas sobre os intervalos de variáveis, etc. A segunda linha usa a função *lrm* para ajustar o modelo à fórmula do `modelo.01`. Você não precisa especificar `family`, mas, para

⁸ Harrell (2001, p. 248) adverte contra usar a precisão de classificação como uma forma de medir quão bom um modelo é. Vamos usar uma medida melhor em breve.

Figura 73 – Efeitos do model.01: gráfico de barras com probabilidades observadas/previstas e suas barras de intervalo de confiança de 95% (painel esquerdo); gráfico de efeitos da biblioteca *effects* (painel direito)

muitas aplicações futuras (mais tarde, quando você se tornar mais proficiente), vários outros argumentos podem ser fornecidos como mostrado abaixo:

Código em R

```
> dd=datadist(CLAUSE.ORDERS); options(datadist="dd")
> model.01.lrm=lrm(formula(model.01), data=CLAUSE.ORDERS,
x=TRUE, y=TRUE, linear.predictors=TRUE, se.fit=TRUE);
model.01.lrm
[...]
```

Model Likelihood		Discrimination			
Ratio Test		Indexes			
Obs	403	LR chi2	116.97	R2	0.353
mc-sc	275	d.f.	1	g	1.365
sc-mc	128	Pr(> chi2)	<0.0001	gr	3.915
max deriv	2e-09			gp	0.240
Brier	0.159				

```
Rank Discrim.
Indexes
C      0.776
Dxy    0.552
gamma  0.877
tau-a  0.240
Brier  0.159
[...]
```

Eu não estou mostrando todo o resultado, mas algumas vantagens de *lrm* devem estar claras: Você obtém o teste de significância do modelo (veja o teste de razão de verossimilhança), você obtém um valor do R^2 (muitas vezes dado como R^2 de Nagelkerke e, como de costume, variando de 0 a 1), e você obtém um valor de C, que pode ser usado como um indicador da qualidade de classificação do modelo. Esse valor varia de 0,5 a 1 e valores acima de 0,8 são considerados bons, o que não conseguimos aqui. (Nota de passagem, $C = 0,5 + (D_{xy}/2)$).

Resumindo: “Uma regressão logística binária mostra que há uma correlação altamente significativa, mas fraca, entre o tipo de oração subordinada e a ordem da oração principal e da oração subordinada

($G = 116,97$; $gl = 1$; $p < 0,001$; R^2 de Nagelkerke = 0,353, $C = 0,776$). 73,7% das ordens são classificadas corretamente (contra uma precisão de 68,24%). O modelo prevê que as orações subordinadas causais preferem seguir as orações principais, enquanto as orações temporais preferem preceder as orações principais. [Adicione um gráfico e talvez coeficientes.]”

5.2.10 Regressão logística com um preditor categórico

Como antes, vamos aumentar a complexidade dos modelos de regressão de forma gradual. Passamos agora, portanto, para um preditor categórico, CONJ. Mais uma vez, vou mostrar muito menos resultados a partir de agora. O modelo é significativo, com um valor de razão de verossimilhança de 123,32 com $gl=3$ (ver acima). Uma vez que FAMILIARITY tem mais de 1 grau de liberdade, você usa drop1 (ou outras funções, veja o arquivo de códigos) para obter um valor- p , e FAMILIARITY é altamente significativo.

Código em R

```
> model.01=glm(ORDER~CONJ, data=CLAUSE.ORDERS,family=binomial)¶
> summary(model.01)¶
> confint(model.01)¶
> drop1(model.01, test="LR")¶
```

Para explorar o que os coeficientes revelam, você recorre às previsões. Estas mostram que *als/when* (quando) e *nachdem/after* (depois) preferem SC-MC, enquanto *bevor/before* (antes) e *weil/because* (porque) preferem MC-SC. Eu não vou explicar o significado do intercepto e de todos os coeficientes em detalhes aqui, mas você vai encontrar todas essas explicações no arquivo de códigos e deve as ler com cuidado! A lógica e todas as outras coisas são as mesmas que as explicadas acima com (61) e (62), a única diferença é que os coeficientes agora representam diferenças entre o intercepto - o primeiro nível de FAMILIARITY - e os outros níveis na escala de logaritmos das probabilidades.

Código em R

```
> conj=effect("CONJ", model.01)¶
> preds.hyp=data.frame(conj$x, PREDICTIONS=ilogit(conj$fit),
  LOWER=ilogit(conj$lower), UPPER=ilogit(conj$upper))¶
> preds.hyp¶
  CONJ      PREDICTIONS      LOWER      UPPER
1 als/when      0.60215054 0.4997995 0.6962850
2 bevor/before 0.39130435 0.2623180 0.5375019
3 nachdem/after 0.60000000 0.4773236 0.7112984
4 weil/because 0.07537688 0.0459497 0.1212547
```

Então vemos quão bem o modelo classifica as ordens. Conseguimos uma melhoria em relação à chance, mas 76,18% não parece um grande avanço.

Código em R

```
> predictions.num=fitted(model.01)¶
> predictions.cat=ifelse(predictions.num>=0.5,"sc-mc", "mc-sc")¶
> table(ORDER, predictions.cat)¶
> (212+95)/length(predictions.cat)¶
```

Finalmente, representamos os dados visualmente como antes e geramos um modelo com `lrm` para obter R^2 e C , que são 0,369 e 0,798, respectivamente. Com esses gráficos e estatísticas de resumo, podemos agora resumir o resultado de nosso modelo de regressão da mesma maneira que acima na página 304. Observe que, dada a sobreposição das conjunções temporais, deve-se, estritamente falando, também testar se a decisão de três conjunções temporais é justificada...

Código em R

```
> barplot(preds.hyp$PREDICTIONS, ylim=c(0, 1), names.arg=preds.hyp$CONJ)¶
> plot(allEffects(model.01), ask=FALSE, ylim=c(0, 1), rescale.axis=FALSE)¶
> dd=datadist(CLAUSE.ORDERS); options(datadist="dd")¶
> model.01.lrm=lrm(formula(model.01), data=CLAUSE.ORDERS); model.01.lrm¶
```

5.2.11 Regressão logística para um preditor numérico

Como a última regressão logística monofactorial, passamos agora para um preditor numérico, `LENGTH_DIFF`.

Código em R

```
> model.01=glm(ORDER~LENGTH_DIFF, data=CLAUSE.ORDERS, family=binomial)¶
> summary(model.01)¶
> confint(model.01)¶
```

Por razões de consistência, você também executa `drop1` (e talvez `Anova/anova`), embora, como `LENGTH_DIFF` tem 1 gl, isso não seja realmente necessário.

Código em R

```
> drop1(model.01, test="LR")¶
```

Observe que existe uma ligeira diferença entre os valores- p do resultado de `summary`, por um lado, e de `drop1`, `anova` e `Anova`, por outro lado. De acordo com [Fox e Weisberg \(2011, p. 239\)](#), o teste de razão de verossimilhança que você obtém com o `drop1` etc. pode ser mais confiável.

O modelo é significativo, mas o efeito realmente parece bastante fraco. Para entender os coeficientes, criamos as previsões, mas desta vez teremos que discutir isso com mais detalhes, pois é aqui que se manifesta uma ligeira desvantagem da perspectiva de probabilidade em resultados de regressão logística. Pelo menos de certa forma, as coisas são como antes: o intercepto ainda representa a probabilidade da ordem prevista quando a variável independente está em seu primeiro nível ou, como aqui, 0, o que você pode ver em `preds.hyp` e comparar com `ilogit(-0.77673)`¶:

Código em R

```
> lendiff=effect("LENGTH_DIFF", model.01, xlevels=list(LENGTH_DIFF=max(abs(range(LENGTH_DIFF))):max(abs(range(LENGTH_DIFF))))); lendiff¶
> preds.hyp=data.frame(lendiff$x, PREDICTIONS=ilogit(lendiff$fit), LOWER=ilogit(lendiff$lower), UPPER=ilogit(lendiff$upper)); preds.hyp¶
```

Similarmente, o coeficiente de `LENGTH_DIFF` ainda representa a mudança da probabilidade da ordem prevista, *SC-MC*, para uma mudança de unidade de `LENGTH_DIFF`. No entanto, essa mudança é linear/-constante somente na escala do logaritmo das probabilidades - uma vez que a verificamos na escala de

probabilidade, você pode ver que uma variação de 1 de `LENGTH_DIFF` não causa a mesma diferença em probabilidades: quando você aumenta `LENGTH_DIFF` em 1

- de -20 a -19, isso faz com que a probabilidade prevista de *SC-MC* cresça 0,006019;
- de -10 a -9, isso faz com que a probabilidade prevista de *SC-MC* cresça 0,0078747;
- de 0 a 1, isso faz com que a probabilidade prevista de *SC-MC* cresça 0,0096109.

Isso ocorre porque a transformação logit inversa não é linear (no espaço de probabilidade). Mas vejamos agora quão bem este modelo prediz as ordens:

Código em R

```
> predictions.num=fitted(model.01)¶
> predictions.cat=ifelse(predictions.num>=0.5, "sc-mc", "mc-sc")¶
> table(ORDER, predictions.cat)¶
> (272+2)/length(predictions.cat)¶
```

De algum modo, o desempenho é péssimo: é pior do que o acaso, mesmo que a direção do efeito faça sentido - você deve se certificar de que isso equivale a “curto antes de tempo”? Além disso, você pode ver a partir da tabela que o modelo quase nunca prevê *SC-MC* - ele faz isso apenas cinco vezes. Para visualizar o efeito de `LENGTH_DIFF` e explorar esse mau desempenho, vamos plotar as probabilidades previstas contra as diferenças de comprimento de `preds.hyp` (ver Figura 74).

Código em R

```
> plot(preds.hyp$LENGTH_DIFF, preds.hyp$PREDICTIONS,xlim=c(-35, 35), ylim=c(0, 1))¶
```

Como você pode ver, você não obtém uma linha reta de regressão, mas uma curva, no espaço de probabilidade. Assim, a mudança de `LENGTH_DIFF` por uma palavra tem efeitos diferentes dependendo de onde ela acontece, e esta é a desvantagem da escala de probabilidade à qual eu aludi anteriormente. No entanto, dada a boa forma pelas quais podemos gerar gráficos de tais curvas no R, esta é uma desvantagem com a qual estou feliz de conviver (em comparação com as desvantagens das escalas de probabilidades/chances ou do logaritmo das probabilidades).

A Figura 74 também ajuda a entender a precisão ruim da classificação. A linha horizontal no ponto de corte de $y=0,5$ só se aplica a poucos pontos (veja os rugs). Uma maneira de tentar forçar a regressão a fazer previsões um pouco mais diversas é escolhendo um ponto de corte diferente de 0,5, e uma possibilidade é usar a mediana de todas as probabilidades previstas (0,3150244; ver [Hilbe \(2009, Seção 7.2.2\)](#) para discussão).

Se você fizer isso aqui, você de fato obterá frequências mais equilibradas de previsões categóricas, mas a precisão diminui ainda mais. O arquivo de códigos mostra como, quando você escolhe 0,42 como um ponto de corte, você obtém uma frequência de previsões ligeiramente mais equilibrada e ainda cerca de 68% correta. Por agora, usaremos esse valor, e deixarei o tópico das curvas ROC e de como elas ajudam a identificar pontos de corte para exploração futura. Note, entretanto, que se você não escolher o ponto de corte padrão de 0,5 para as previsões categóricas, você deve mencionar qual você escolheu e o porquê. Finalmente, fazemos a regressão novamente com `lrm` para obter os realmente pequenos R^2 (0.026) e C (0.603), e então podemos resumir nossos resultados como acima.

Figura 74 – Efeitos do model.01: gráfico de dispersão com probabilidades previstas e sua faixa de intervalo de confiança de 95%

Código em R

```
> dd=datadist(CLAUSE.ORDERS); options(datadist="dd")
> model.01.lrm=lrm(formula(model.01), data=CLAUSE.ORDERS);model.01.lrm
```

5.2.12 Uma regressão logística com dois preditores categóricos

Vamos passar agora para a primeira regressão logística com mais de um preditor: vamos explorar se CONJ (conjunção), MORETHAN2CL (mais que duas orações) e a interação entre elas afetam as ordens da orações. Como um dos preditores tem mais de 1 grau de liberdade, ajustamos o modelo e imediatamente adicionamos drop1 e Anova:

Código em R

```
> model.01=-glm(ORDER~CONJ*MORETHAN2CL, data=CLAUSE.ORDERS,family=binomial)
> summary(model.01)
> confint(model.01)
> drop1(model.01, test="LR")
> options(contrasts=c("contr.sum", "contr.poly"))
> Anova(model.01, type="III", test.statistic="LR")
> options(contrasts=c("contr.treatment", "contr.poly"))
```

Você pode ver que drop1 agora só retorna o valor-*p* para a interação não-significativa, a qual, em um processo normal de seleção de modelo, omitiríamos agora. Deixamos a interação aqui para explorar como entendê-la e visualizá-la - com Anova, no entanto, também obtemos todos os outros valores-*p*; apenas CONJ parece significativo.

Sobre as previsões usando o código, eu só as mostro no arquivo de códigos porque lá eu tenho mais espaço para explicar o código e os significados dos coeficientes, os quais, como de costume, seguem as regras em (61) e (62)). Se você olhar para preds.hyp, *weil/because* (porque) se destaca, mas o resultado

também mostra porque a interação não é significativa: os intervalos de confiança são enormes e, em geral, as conjunções apresentam um padrão semelhante em ambos os níveis de MORETHAN2CL. Quão boas são essas previsões?

Código em R

```
> predictions.num=fitted(model.01)¶
> predictions.cat=ifelse(predictions.num>=0.5,"sc-mc", "mc-sc")¶
> table(ORDER, predictions.cat)¶
> (216+93)/length(predictions.cat)¶
```

Obtemos quase 77,7% corretamente, o que está pelo menos acima de chances do acaso novamente. Finalmente, representamos os dados visualmente como antes e geramos um modelo com `lrm` para o teste de modelo global (razão de verossimilhança $\chi^2 = 132,06$, $gl=7$, $p<0,001$), R^2 (0,392) e C (0,82).

Código em R

```
> dd=datadist(CLAUSE.ORDERS); options(datadist="dd")¶
> model.01.lrm=lrm(formula(model.01), data=CLAUSE.ORDERS);model.01.lrm¶
```

Como os dados envolvem agora uma interação, o código pode se tornar um pouco mais complexo (pelo menos quando você não usa as funções do pacote de efeitos) e eu mostro isso apenas no arquivo de códigos. A interação não significativa é refletida pela grande sobreposição dos intervalos de confiança e os semelhantes valores das (diferenças das) probabilidades previstas, o que é, com tudo o mais, o que você discutiria na seção de resultados.

5.2.13 Uma regressão linear com um preditor categórico e um numérico

Como na Seção 5.2.5, passamos agora a um caso com uma interação entre um preditor categórico e um numérico, o que significa novamente que os coeficientes de interação refletirão ajustes na inclinação do preditor numérico.

Código em R

```
> model.01=glm(ORDER~CONJ*LENGTH_DIFF, data=CLAUSE.ORDERS,family=binomial)¶
> summary(model.01)¶
> confint(model.01)¶
> drop1(model.01, test="LR")¶
> options(contrasts=c("contr.sum", "contr.poly"))¶
> Anova(model.01, type="III", test.statistic="LR")¶
> options(contrasts=c("contr.treatment", "contr.poly"))¶
```

Desta vez, ambos os efeitos principais são significativos, e a interação é quase não significativa. Portanto, seria justificado pelo menos olhar para a interação (como vamos fazer de qualquer forma para fins didáticos).

Geramos as previsões da maneira usual, e é novamente prudente, talvez, também gerar uma tabela ligeiramente mais plana que possa ser inspecionado antes de recorremos aos gráficos, como mostrado no arquivo de códigos:

Código em R

```
> intact=effect("CONJ:LENGTH_DIFF", model.01,xlevels=list(LENGTH_DIFF=seq(-32, 32,
length.out=9)))¶
> preds.hyp=data.frame(intact$x, PREDICTIONS=ilogit(intact$fit),
LOWER=ilogit(intact$lower),UPPER=ilogit(intact$upper)); preds.hyp¶
```

A representação mais plana de `preds.hyp.2` em especial é agora mais fácil de ler. Você pode ver para cada conjunção como a probabilidade predita de *SC-MC* muda conforme *LENGTH_DIFF* muda. Um gráfico tornará isso ainda mais óbvio em breve. Novamente, leia o arquivo de códigos em detalhes para entender como os coeficientes resultam nessas previsões! Quão boas são as previsões?

Código em R

```
> predictions.num=predict(model.01, type="response")¶
> predictions.cat=ifelse(predictions.num>=0.5,"sc-mc", "mc-sc")¶
> table(predictions.cat, ORDER)¶
> (228+83)/length(predictions.cat)¶
```

Aproximadamente 77,2% são classificadas corretamente, e um ponto de corte de 0,57 resulta em um valor ligeiramente melhor (de cerca de 78,2%). Mas agora, a que os resultados equivalem? Neste caso, onde os valores-*p* de todos os efeitos são pelo menos <0,07, faremos bons gráficos, os quais serão parcialmente baseados em `preds.hyp.2`, pois ele tem colunas separadas para as conjunções. A Figura 75 é o que queremos criar (no painel inferior, as letras são as primeiras letras das conjunções alemãs). Lembre-se, porém: a interação é significativa, então é nisso que você deve se concentrar - não nos efeitos principais! O arquivo de códigos mostra como isso é feito exatamente. Eu sei que são muitas linhas, mas você deve investir tempo para entender o que cada linha está fazendo porque, depois de você entender isso, você será capaz de usar essa lógica para muitos exemplos no seu próprio trabalho.

O primeiro efeito principal é um tanto familiar com base no que vimos cima. As três conjunções temporais preferem *SC-MC* enquanto *weil/because* prefere fortemente *MC-SC*. O segundo efeito principal também é familiar: curto antes de longo. Agora a interação é interessante. Três observações podem ser feitas:

- *als/when* e *nachdem/after* apresentam preferências médias semelhantes para *SC-MC* e ambas reagem a *LENGTH_DIFF* de uma forma que é (mais) compatível com “curto antes de longo” do que a média;
- *bevor/before* não apenas tem menos preferência por *SC-MC*, mas também a tendência oposta em comparação com as outras duas conjunções temporais: quando a oração principal se torna mais longa, ela tende a preceder a oração subordinada;
- as orações subordinadas com *weil/because* são praticamente completamente imunes a considerações de comprimento: elas querem vir em segundo lugar de qualquer forma.

(O arquivo de códigos também contém códigos para a interação com os intervalos de confiança, mas eles tornam o gráfico mais difícil de ler, acabando com a sua finalidade; o gráfico de efeitos é mais útil nesse sentido.) Tudo visto acima, juntamente com os valores-*p* para os preditores, o valor-*p* para o modelo global (razão de verossimilhança $\chi^2_2 = 135,21$, *gl*= 7, *p* <0,001), R^2 (0,399) e *C* (0,818) do modelo correspondente com *lrm* (veja abaixo), faria parte da sua Seção de resultados.

5.2.14 Uma regressão logística com dois preditores numéricos

Agora a regressão logística final, com dois preditores numéricos, como anteriormente. Vamos verificar se os comprimentos das duas orações afetam a escolha de suas ordens. Primeiro uma pergunta: como isso é diferente de verificar se LENGTH_DIFF é significativo?

Figura 75 – Todos os efeitos do model.01 (lembre-se, no entanto: se a interação é significativa, sua discussão deve se concentrar nela, não nos efeitos principais)

É diferente, porque leva em conta onde uma diferença de comprimento particular pode ser observada: Se LENGTH_DIFF for 1, esse valor não revela se ele é resultado da oração principal e da subordinada contendo 10 e 9 ou 20 e 19 palavras, respectivamente. Como sempre, ajustamos o modelo; Uma vez que ambas as variáveis são numéricas, drop1 e Anova não são realmente necessárias:

Código em R

```
> model.01=glm(ORDER~LEN_MC*LEN_SC, data=CLAUSE.ORDERS,family=binomial)¶
> summary(model.01)¶
> confint(model.01)¶
```

Isso tudo é tão não-significativo quanto possível; É apenas por razões didáticas que prosseguimos com as previsões, e com dois preditores numéricos abrangendo uma ampla gama de valores, tanto `preds.hyp` quanto o `preds.hyp.2` mais plano (consulte o arquivo de códigos) não são fáceis de processar. (Não se esqueça de ler a explicação do arquivo de códigos sobre os coeficientes.) Em seguida, verificamos a tabela de classificação cruzada usando imediatamente a mediana das probabilidades previstas como o ponto de corte:

Código em R

```
> predictions.num=predict(model.01, type="response")¶
> predictions.cat=ifelse(predictions.num>=median(predictions.num), "sc-mc", "mc-
sc")¶
> table(predictions.cat, ORDER)¶
> (151+82)/length(predictions.cat)¶
```

A precisão é bastante baixa, mesmo além das chances de acaso, o que não é surpreendente considerando-se os valores- p dos preditores. Mas como são os efeitos? Como antes, você tem basicamente duas possibilidades. Na primeira delas, você pode novamente gerar um gráfico rotativo tridimensional (com `plot3d`), e o arquivo de códigos mostra, na minha opinião, uma versão melhor onde diferentes cores e letras (a primeira letra da primeira oração) representam quais ordem é prevista para qual combinação de comprimentos. O fato de que a interação é insignificante é refletido pelo fato de que as letras quase formam um plano reto em espaço tridimensional. A versão mais pronta para publicação é mostrada na Figura 76.

LENGTHMC (comprimento da oração principal) e LENGTHSC (comprimento da oração subordinada) estão nos eixos x e y , respectivamente. Quando a previsão categórica é MC-SC, eu ploto um m , caso contrário um s . Quanto maior e mais escura for a letra, mais "certo" é o modelo sobre a previsão no sentido de que a previsão é baseada em uma probabilidade que está mais distante do ponto de corte. A linha reta cinza é a diagonal principal onde ambas as orações são igualmente longas, e a linha preta curva indica, para cada comprimento da oração principal, onde a previsão muda para a outra ordem da oração, razão pela qual lá as letras são tão claras. Embora a interação não seja significativa, vemos brevemente “curto antes de longo” novamente: quando a oração principal é curta (à esquerda do gráfico), à medida que a oração subordinada se torna mais longa, ela tende a aparecer primeiro, e a mesma lógica se aplica a orações subordinadas. Os gráficos de efeitos mostram o mesmo tipo de resultado, porém dividindo os comprimentos da oração subordinada em dez intervalos diferentes e, em seguida, gerando gráficos com linhas de regressão e seus intervalos de confiança. Embora seja mais fácil de gerar em termos de código, eu considero o gráfico menos fácil de interpretar do que a Figura 76.

Finalmente, você geraria o modelo com `lrm` e resumiria tudo: Aqui, obviamente, todos os preditores são fracos e não significativos, e tanto R^2 como C são realmente baixos (0,028 e 0,606, respectivamente).

Concluimos então a visão geral dos diferentes modelos de regressão logística. Novamente, como mencionado acima na p. 263, você naturalmente não teria feito todos esses modelos separadamente, mas teria utilizado um processo de seleção de modelo como o da Seção 5.2.7. Um dos exercícios para este capítulo possibilitará que você faça isso para os dados atuais, e você verá que os resultados revelam mais sobre o comportamento inesperado de *bevor/before* na Seção 5.3.5.

O que falta abordarmos, contudo, é novamente como testar se as pressuposições do modelo de regressão são atendidas. Acima mencionei três critérios diferentes (mas veja Fox e Weisberg (2011, Cap. 6)). Você já conhece a inspeção de resíduos, mas superdispersão é algo novo. Isso requer que você

A interação do Comprimento da Oração no model.01

Figura 76 – A interação LENGTHMC : LENTHSC no model.01

olhe para a razão/proporção do desvio residual do seu modelo e seu graus de liberdade residuais, que não devem ser muito maiores do que 1. Neste caso, é $\frac{495,58}{399} = 1,242$. Várias referências apenas dizem que, se você obtivesse um valor que é muito maior do que 1, por exemplo, > 2 , então você executaria a análise de glm novamente com o argumento `family=quasibinomial`, and take it from there. Baayen (2008, p. 199) usa como aproximação um teste qui-quadrado do desvio residual com o grau de liberdade residual:

Código em R

```
> pchisq(495.58, 399, lower.tail=FALSE)
[1] 0.0006880771
```

Assim, se esta fosse uma análise real com um resultado significativo, recomenda-se seguir esse

conselho. Os outros critérios que mencionei foram relacionados aos valores absolutos dos resíduos padronizados do modelo e dos dfbetas. Os primeiros são um tipo de residuais corrigidos (ver [Fox e Weisberg \(2011, p. 286\)](#)) e [Field, Miles e Field \(2012, Seções 8.6 e 8.7\)](#)) sugerem que não mais de 5% deve ser > 2 ou < -2 . Isso é fácil de testar:

Código em R

```
> prop.table(table(abs(rstandard(model.01))>2))  
FALSE      TRUE  
0.99751861 0.00248139
```

Nesse caso, nem mesmo 1% é > 2 ou < -2 . Similarmente direto é o teste dos dfbetas, os quais refletem o quanto um coeficiente de regressão muda quando cada caso é removido dos dados. Novamente, testar isso no R é simples:

Código em R

```
> summary(dfbetas<-abs(dfbeta(model.01)))
```

O resultado (não mostrado aqui) indica que, de fato, nenhum dfbeta absoluto é maior que 0,1, portanto esse critério também não apresenta problemas para o nosso modelo. A verificação cuidadosa dos diagnósticos é um componente importante da verificação do modelo, e o R como um todo, e a biblioteca car em particular, tem muitas funções úteis para este propósito.

Conselho

- assim como na Seção 5.2, isso também pode ajudar a interpretar os coeficientes de regressão quando as variáveis de entrada são centradas
- a função `hoslem.test` da biblioteca `ResourceSelection` para o teste de Hosmer-Lemeshow (veja [Hilbe \(2009, Seção 7.2\)](#)) (você quer ver um resultado não significativo)
- [Field, Miles e Field \(2012, Seção 8.8.2\)](#) sobre a pressuposição da linearidade de logit
- [Pampel e Publications \(2000\)](#), [Jaccard, Jaccard e Lewis-Beck \(2001\)](#), [Crawley \(2005, Cap. 16\)](#), [Crawley \(2007, Cap. 17\)](#), [Faraway \(2006, Cap. 2 e 6\)](#), [Zuur, Ieno e Smith \(2007, Seção 6.1\)](#), [Gelman e Hill \(2007, Cap. 5\)](#), [Baayen \(2008, Seção 6.3\)](#), [Baguley \(2012, cap. 17\)](#)

5.3 Outros modelos de regressão

Os dois tipos de modelos de regressão acima têm sido os mais usados na Linguística. Nesta seção, vou apresentar uma variedade de outros modelos de regressão que ainda não estão tão difundidos, mas que certamente serão mais usados no futuro próximo: regressão logística ordinal (onde a variável dependente é ordinal), regressão multinomial (onde a variável dependente é categórica com mais de 3 ou mais níveis) e regressão de Poisson (onde a variável dependente é constituída por frequências). A lógica da exposição será como acima, mas - por razões de espaço - muito abreviada. Especificamente - depois de uma breve introdução sobre cada seção e seus dados, vou apenas discutir um exemplo para cada regressão apresentada no livro, ou seja, o caso de duas variáveis independentes, uma categórica e uma numérica. No entanto, o arquivo de códigos irá discutir seis modelos de regressão para cada, assim como antes, para que você obtenha um bom tratamento homogêneo de todos os modelos. Por isso, recomendo

que você carregue os dados, leia o capítulo no livro, e siga junto com o quinto dos seis exemplos no arquivo de códigos e, em seguida, explore os outros exemplos com base no arquivo de código também.

5.3.1 Uma regressão logística ordinal com um preditor categórico e outro numérico

O exemplo que exploraremos para abordar a regressão logística ordinal diz respeito a qual variável de um conjunto de variáveis independentes nos permite prever qual de três exames ou trabalhos de final de curso os alunos de inglês como língua estrangeira escolherão fazer. Isso envolve as seguintes variáveis:

- uma variável ordinal dependente, a dizer ASSIGNMENT : ORALEXAM vs. LABREPORT vs. THESIS (AVALIAÇÃO: EXAME ORAL vs RELATÓRIO DE LABORATÓRIO vs TESE); basicamente, essas avaliações estão ordenadas em ordem de dificuldade crescente (baseada em estudo anterior);
- uma variável binária independente SEX : FEMALE vs. MALE indicando o sexo do aluno cuja escolha foi registrada;
- uma variável independente categórica REGION (REGIÃO), que representa a região geográfica de origem do estudante: CENTRAL - EUROPEAN, HISPANIC, e MIDDLE - EASTERN (CENTRAL-EUROPÉIA, HISPÂNICA, e ORIENTE MÉDIO);
- uma variável numérica independente WORKHOURS (HORAS DE TRABALHO), representando o número de horas/meses que os alunos alegaram terem investido no curso;
- uma variável numérica independente MISTAKES (ERROS), que representa o número de erros cometidos pelos alunos fizeram na última avaliação do curso.

Aqui estão os passos de uma regressão logística ordinal que vamos seguir:

Procedimentos

- Formulação da hipótese
- Importação, preparação para a modelagem e exploração dos dados
- Cálculo, seleção e interpretação de um modelo de regressão lógica:
 - obtenção dos valores- p para todos os preditores e para o modelo como um todo
 - interpretação dos coeficientes/estimativas da regressão com bases (i) nos valores preditos e (ii) nos gráficos das probabilidades previstas
- Teste das principais pressuposições do teste:
 - as suspeitas comuns: independência dos dados e resíduo, ausência de pontos excessivamente influentes e de multicolinearidade
 - a variável dependente “se comporta de forma ordinal em relação a cada preditor” (Harrell (2001, p. 332))

Primeiramente, as hipóteses:

H_0 : Não há correlação entre ASSIGNMENT e os preditores (variáveis independentes e suas interações): $R^2 = 0$.

H_1 : Há correlação entre ASSIGNMENT e os preditores (variáveis independentes e suas interações): $R^2 > 0$.

Em seguida, você carrega os dados de <_inputfiles/05-4-1_assignments.csv>, bem como a biblioteca *rms*, cuja função *lrm* usaremos aqui:

Código em R

```
> rm(list=ls(all=TRUE)); library(rms)¶
> ASSIGNS=read.delim(file=file.choose()); str(ASSIGNS)¶
```

Se você analisar o resumo fornecido por *str*, verá que os níveis do fator *ASSIGNMENT* não estão na ordem correta, e que esse fator não é nem mesmo um fator ordenado, o que significa que o R o trata como uma variável categórica (como todos os fatores neste livro até agora), não como a variável ordinal desejada. Portanto, mudamos isso (verifique *str* novamente), e então podemos carregar na memória (*attach*) e, uma vez que usaremos a função *lrm* novamente, criamos o objeto *datadist* necessário:

Código em R

```
> ASSIGNS$ASSIGNMENT=factor(ASSIGNS$ASSIGNMENT, ordered=
TRUE, levels=levels(ASSIGNS$ASSIGNMENT)[c(2,1,3)])¶
> str(ASSIGNS); attach(ASSIGNS)¶
> ddist<-datadist(ASSIGNS); options(datadist="ddist")¶
```

Como mencionado anteriormente, vou agora pular os primeiros quatro modelos discutidos no arquivo de códigos e ir diretamente para o quinto, onde exploramos a influência conjunta de *REGION* e *WORKHOURS* sobre *ASSIGNMENT*:

Código em R

```
> model.01=lrm(ASSIGNMENT ~ REGION*WORKHOURS, data=ASSIGNS,
x=TRUE, y=TRUE, linear.predictors=TRUE, se.fit=TRUE)¶
> model.01¶
> anova.rms(model.01)¶
```

O modelo é altamente significativo (razão de verossimilhança $\chi^2 = 93.09$, $gl=5$, $p < 0,001$) e mostra que há uma correlação muito forte: $R^2 = 0,908$, $C=0,938$. O resultado de *anova.rms* é um pouco diferente. Em vez de lhe dar um valor-*p* para cada efeito principal e cada interação (como *Anova* da biblioteca *car* fez), você obtém dois valores-*p* para o que cada efeito principal faz sozinho juntamente com o que ele faz na interação, e você obtém um valor-*p* para a interação. Uma vez que a interação é quase significativa, vamos nos concentrar nisso. Mas qual é a sua natureza? Os coeficientes são agora bastante diferentes do que já vi antes: há mais de uma interceptação. Eu explico os significados de cada coeficiente em detalhe no arquivo de códigos, mas a maneira mais fácil de entender os resultados é novamente por meio das probabilidades previstas, as quais vamos gerar usando a mesma lógica, porém um código ligeiramente diferente (a função *effect* não funciona com objetos *lrm*, mas às vezes você pode usar *polr* do pacote *MASS*). Aqui estão as duas primeiras linhas:

Código em R

```
> preds.hyp=expand.grid(REGION=levels(REGION),WORKHOURS=c(0, 1,
floor(min(WORKHOURS)):
ceiling(max(WORKHOURS))))¶
> preds.hyp=data.frame(preds.hyp, predict(model.01,newdata=preds.hyp,
type="fitted.ind"))¶
```

Isso gera uma estrutura de dados `preds.hyp` novamente, a qual contém, para cada combinação de `REGION` e um grande número de valores de `WORKHOURS`, a probabilidade prevista para cada tipo de avaliação. Por exemplo, quando o estudante é da região hispânica e investe 26 horas de trabalho, há uma previsão forte de que ele escolha o relatório de laboratório:

Código em R

```
> preds.hyp[preds.hyp$REGION=="hispanic"& preds.hyp$WORKHOURS==26,]¶
```

Mas queremos isso ainda melhor: não queremos apenas as probabilidades previstas, mas também imediatamente para cada linha qual é a previsão categórica. Como você pode ver, a função `predict` combinou o nome da variável dependente, `ASSIGNMENT`, com os níveis previstos inserindo um ponto entre eles. Não queremos ver isso, então usamos o seguinte:

Código em R

```
> preds.hyp=data.frame(preds.hyp, ASSIGNMENT.pred=
sub("^.*? ", , names(preds.hyp)[- (1:2)][max.col(preds.hyp[, -(1:2)])]))¶
> preds.hyp[38:42,]¶
```

A função `sub` requer três argumentos: o que procurar (e o argumento `".*"` significa "caracteres até e incluindo um ponto final"), o que colocar no lugar disso (e significa "nada", isto é, "apagar"), e onde fazer tudo isso (nos nomes de três colunas de `preds.hyp` que não são as duas primeiras). E, então, esses níveis são selecionados com o vetor de números que resulta do R ter verificado para cada linha onde a probabilidade máxima prevista está (excluindo sempre as duas primeiras colunas de `preds.hyp`, que contêm as variáveis independentes!). Verifique isso observando essas cinco linhas do resultado.

Agora, primeiramente removemos as primeiras linhas de `preds.hyp` porque estas só foram incluídos para explicar os coeficientes, mas não foram representativas dos valores reais de `WORKHOURS`. Em seguida, verificamos a precisão da classificação:

Código em R

```
> preds.hyp=preds.hyp[- (1:6),]¶
> predictions.num=predict(model.01, type="fitted.ind")¶
> predictions.cat=sub("^.*? [
.=]", , colnames(
predictions.num)[max.col(predictions.num)])¶
> table(predictions.cat, ASSIGNMENT)¶
```

Como você pode ver, conseguimos uma boa precisão, quase 85%, o que é significativamente melhor do que o nível de chance ao acaso de 33% (uma vez que as três avaliações são igualmente frequentes). Em seguida, geramos um gráfico das probabilidades previstas, as quais, dada a multiplicidade de resultados que esses tipos de regressões produzem, tornam-se um pouco mais complexas. Uma maneira de representar esses resultados é mostrada na Figura 77. Há um painel para cada região, as horas de trabalho estão em cada eixo x, as probabilidades previstas em cada eixo y e as três avaliações são representadas por linhas e suas primeiras letras. No geral, há um efeito muito forte de `WORKHOURS`: os alunos que declararam menos horas de trabalho são fortemente previstos para escolher a avaliação mais fácil, o exame oral. Aqueles que relatam um número intermediário de horas de trabalho são fortemente previstos para usar a avaliação intermediariamente difícil, o relatório de laboratório. Aqueles que relatam o maior número de horas de trabalho, por sua vez, são previstos para escolher a tese. A interação quase significativa,

porém, indica que esse comportamento não é completamente uniforme nas três regiões: por exemplo, os estudantes hispânicos escolhem as avaliações mais difíceis com um menor número de horas de trabalho do que os estudantes do Oriente Médio. Os europeus centrais preferem mais os exames orais mesmo que eles tenham trabalhado um número de horas quando os outros alunos já começaram a preferir o relatório de laboratório, e apenas os mais trabalhadores do Oriente Médio escolhem a tese. (Veja o arquivo de códigos para outros gráficos.)

Vamos finalmente verificar alguns pressupostos deste tipo de regressão: os primeiros cinco gráficos representam os resíduos, os quais são na sua maioria muito perto de 0, como requerido. A pressuposição de ordinalidade parece um pouco mais problemática, e embora isso exija mais atenção, isso está além do escopo deste livro; veja as recomendações para um estudo mais aprofundado.

Código em R

```
> par(mfrow=c(2, 4))¶
> residuals(model.01, type="score.binary", pl=TRUE)¶
> plot.xmean.ordinaly(ASSIGNMENT ~ REGION*WORKHOURS)¶
> par(mfrow=c(1, 1))¶
```

Deixando essa questão de lado por agora, você seria capaz agora de resumir os resultados de regressão numericamente (razão de verossimilhança χ^2 , gl, p, R^2 , C) e discutir o gráfico e suas implicações em conformidade com o que discutimos acima.

Figura 77 – A interação REGION:WORKHOURS no modelo.01

Recomendação para estudo/leitura

- a função `polr` da biblioteca MASS , para regressões logísticas ordinais
- Harrell (2001, Cap. 13-14), Baayen (2008, Seção 6.3.2), Hilbe (2009, Cap. 10), Agresti (2010), Fox e Weisberg (2011, Seção 5.9))

5.3.2 Um regressão multinomial com um preditor cagetórico e um numérico

Depois da discussão sobre regressão logística ordinal, passamos agora à regressão multinomial. Por uma questão de simplicidade, usaremos o mesmo conjunto de dados e simplesmente não consideraremos ASSIGNMENT uma variável ordinal (e, portanto, um fator ordenado), mas uma variável categórica e, portanto, um fator desordenado “regular”. Este é o procedimento que vamos seguir:

Procedimentos

- Formulação das hipóteses
- Leitura dos dados, preparando-os para modelagem e explorando-os
- Cálculo, seleção e interpretação do modelo linear:
 - obtenção dos valores- p para todos os preditores do modelo como um todo
 - interpretação dos coeficientes/estimativas da regressão com base (i) nos valores preditos e (ii) nos gráficos de probabilidades previstas
- Teste das principais pressuposições dos testes:
 - as suspeitas comuns: independência dos dados e resíduo, ausência de pontos excessivamente influentes e de multicolinearidade
 - independência de alternativas irrelevantes, um teste de Hasuman-McFadden não significativo (o que eu não discutirei aqui, seja referências abaixo)

Considerando que estamos usando o mesmo conjunto de dados, as hipóteses também permanecem as mesmas, ademais você pode carregar o arquivo, alterar os níveis de ASSIGNMENT como acima, (porém sem alterá-lo para um fator ordenado), e carregar várias bibliotecas. Em seguida, ajustamos um modelo de regressão multinomial da seguinte maneira:

Código em R

```
> model.01=multinom(ASSIGNMENT ~ REGION*WORKHOURS, data=ASSIGNS)¶
> summary(model.01, Wald=TRUE)¶
> mlogit.display(model.01)¶
> confint(model.01)¶
```

A saída de `summary` é um pouco “esmagadora” porque novamente obtemos múltiplas intercepções e coeficientes para todos os níveis, exceto o primeiro nível da variável dependente. Esses resultados representam de forma um tanto complicada as diferenças entre o primeiro nível da variável dependente e cada um dos outros. De certa forma, as regressões multinomiais são séries de regressões logísticas

binárias. Também obtemos estatísticas de Wald, as quais são, como de costume, os coeficientes divididos por seus erros-padrão.

Vamos verificar a significância dos preditores. Infelizmente, não podemos usar `drop1`, mas podemos fazer algo que é praticamente equivalente a ele: uma comparação de anova entre o `model.01` e um modelo sem a interação, além de usar `Anova` na maneira agora já familiar. Ambas as formas revelam que a interação não é de forma alguma significativa: $p > 0,9$

Código em R

```
> anova(model.01, multinom(ASSIGNMENT~REGION+WORKHOURS))¶
> options(contrasts=c("contr.sum", "contr.poly"))¶
> Anova(model.01, type="III")¶
> options(contrasts=c("contr.treatment", "contr.poly"))¶
```

Agora, o que os coeficientes significam? Estou quase tentado a dizer "você não quer saber ..." As explicações dos coeficientes são ainda mais evidência de que tentar entender os resultados em termos de coeficientes não é muitas vezes a estratégia melhor/mais intuitiva. Você encontrará explicações detalhadas sobre eles no arquivo de códigos. Basta dizer aqui que, quando você os exponencia, você obtém proporções/razões entre diferentes probabilidades previstas. Uma representação visual que poderíamos usar é o tipo exemplificado pela Figura 77 acima, e o arquivo de códigos mostra como você pode gerar esse gráfico bem como dois outros. Em geral, os resultados são comparáveis aos da Figura 77: baixos números de horas de trabalho levam a exames orais, números intermediários de horas de trabalho levam a relatórios de laboratório, e os números mais elevados de hora de trabalho são mais associados a teses, e essas preferências são, com algumas pequenas diferenças, obtidas para todas as regiões.

Para determinar a precisão da classificação, podemos prosseguir da maneira usual, ou podemos levar as coisas para o próximo nível. Novamente usamos a função `model.statistics` do bom pacote `polymous` de Antti Arppe:

Código em R

```
> model.statistics(ASSIGNMENT, predictions.cat, predictions.num)¶
```

Isso fornece um conjunto imensamente útil de estatísticas de resumo: estatísticas de verossimilhança e desvios (*deviances*) para o nosso modelo.01 (-329,5837 e 143,4688) e para um modelo que consiste apenas o intercepto (-71,73441 e 143,4688), na precisão de classificação (0,8733) e, como nas excurses de antes, Nagelkerke R^2 (0,9233), tudo o que se desejaria saber ...

Recomendação para estudo/leitura

- [Gries \(2009a, Seção 5.1\)](#) sobre análise (hierárquica) de frequência configuracional (e o script `hcfa` com o qual ela pode ser computada interativamente) e [Field, Miles e Field \(2012, Seções 18.7-18.12\)](#) sobre análise loglinear loglinear; veja também as funções `loglin` e a função `loglm` (da biblioteca MASS) para computar análises loglineares.
- a função `hmftest` da biblioteca `mlogit` to compute o teste de Hausman-McFadden
- [Agresti \(2002, Cap. 7\)](#), [Faraway \(2006, Cap. 5\)](#), [Fox e Weisberg \(2011, Seção 5.7\)](#), [Field, Miles e Field \(2012, Seção 8.9\)](#)

5.3.3 Regressão de Poisson com um preditor categórico e um numérico

Nesta seção, discutirei outro tipo de modelo linear generalizado, a regressão de Poisson, a qual é usada para modelar contagens/frequências. Como discutido acima na página 294, assim como a regressão logística binária, esta abordagem também requer uma função de ligação - desta vez a função exponencial - para garantir que uma abordagem do tipo modelo linear possa ser aplicada a uma variável dependente que nunca é negativa. Como nas duas últimas seções, discuti apenas uma regressão com um preditor categórico e numérico aqui no livro, e incentivarei você a explorar os outros cinco exemplos no arquivo de código. O exemplo que utilizarei para explicar a regressão de Poisson diz respeito a fatores que levam a um número maior/menor de disfluências nas conversas de falantes bilíngues e/ou de falantes altamente avançados de uma segunda língua e envolve as seguintes variáveis:

- uma variável dependente DISFLUENCY (DISFLUÊNCIA), a qual representa o número de disfluências que cada um de 300 falantes produz em 20 minutos de conversação;
- uma variável independente binária SEX: FEMALE vs. SEX: MALE, o sexo dos falantes;
- uma variável independente categórica a categorical MOVEDWHEN, a qual indica quando o falante se mudou para os Estados Unidos: quando adulto, durante o Ensino Médio ou durante o Ensino Fundamental;
- uma variável independente numérica REALITYTV, representando o número de horas/meses que os falantes declararam assistir *reality shows*;
- uma variável independente numérica SOCIALNETWORK, representando o número de horas/meses que os falantes declararam passar tempo nas redes sociais.

Aqui estão os passos de uma regressão de Poisson que vamos seguir:

Procedimentos

- Formulação da hipótese
- Leitura dos dados, preparação dos mesmos para modelagem e exploração dos dados
- Cálculo, seleção e interpretação do modelo linear:
 - obtenção dos valores- p para todos os preditores e para o modelo como um todo
 - interpretação dos coeficientes/estimativas da regressão com base (i) nos valores preditos e (ii) nos gráficos das probabilidades previstas
- Teste das principais pressuposições do teste:
 - as suspeitas comuns: independência dos dados e resíduos, ausência de pontos excessivamente influentes e de multicolinearidade
 - o modelo não pode ter influência de sobredispersão

Primeiramente, as hipóteses, depois carregamos algumas bibliotecas (veja o arquivo de códigos) e também os dados de <_inputfiles/05-4-3_disfluencies.csv>.

H_0 : Não há nenhuma correlação entre DISFLUENCY e os preditores (variáveis independentes e suas interações): $R^2 = 0$.

H_1 : Há uma correlação entre DISFLUENCY e os preditores (variáveis independentes e suas interações): $R^2 > 0$.

Código em R

```
> DISFL=read.delim(file=file.choose())
> str(DISFL); attach(DISFL)
```

O modelo que vamos discutir aqui testa a hipótese de que a frequência de disfluências é correlacionada com a época em que o falante se mudou para os Estados Unidos e com o tempo passado em redes sociais:

Código em R

```
> summary(model.01=glm(FREQDISFL~MOVEDWHEN*SOCNETWORK,data=DISFL, family=poisson))
```

O resultado desse modelo já indica um primeiro problema: sobredispersão. A proporção de desvio residual (3532,6) e de graus de liberdade residuais (294) é muito maior do que 1, e significativo (`pchisq(3532.6, 294, lower.tail=FALSE)`), e, por isso, adequamos o modelo novamente com `family=quasipoisson`, que corrige os erros padrão dos preditores e, assim, os valores-*p*, e calculamos o que foi proposto como um R^2 :

Código em R

```
> summary(model.01=glm(FREQDISFL~MOVEDWHEN*SOCNETWORK,data=DISFL, family=quasipoisson))
> 1-(model.01$deviance/model.01$null.deviance)
[1] 0.2558909
```

Uma vez que estamos usando `glm`, grande parte dos códigos para regressões logísticas também se aplica aqui. Por exemplo, com `drop1` e `Anova` obtemos os valores-*p* para preditores. Obviamente, a interação não é de forma alguma significativa, então normalmente atualizaríamos o modelo deletando a interação, e obviamente, `MOVEDWHEN` não parece exercer influência nenhuma, enquanto `SOCIALNETWORK` sim.

Código em R

```
> drop1(model.01, test="LR")
Single term deletions

Model:
FREQDISFL ~ MOVEDWHEN * SOCNETWORK
Df Deviance scaled dev. Pr(>Chi)
<none>                3532.6
MOVEDWHEN:SOCNETWORK  2 3538.0      0.46628    0.792

> options(contrasts=c("contr.sum", "contr.poly"))
> Anova(model.01, type="III", test="LR")
Analysis of Deviance Table (Type III tests)
Response: FREQDISFL
LR Chisq Df Pr(>Chisq)
MOVEDWHEN          2.7684 2 0.2505
SOCNETWORK         19.3469 1 1.09e-05 ***
MOVEDWHEN:SOCNETWORK 0.4663 2 0.7920
[...]
```

Por razões de exposição apenas, continuamos com a interação. Acima, usamos a função `effect` para obtermos as probabilidades previstas – aqui, utilizaremos essa função para obtermos as frequências previstas, e podemos reusar muito do que já vimos sobre o uso de `effect`. A única diferença real é que acima aplicamos `ilogit` ao resultado do efeito, porque a regressão logística binária usa `logit` como uma função de ligação – uma vez que a regressão de Poisson usa `log` como uma função de ligação, agora aplicamos `exp`. No arquivo de códigos, eu novamente explico os significados dos coeficientes e como eles dão origem às frequências previstas em mais detalhes. Pressigamos, então, para o gráfico. A Figura 78 mostra DISFLUENCY em contraste com SOCIALNETWORK e depois adiciona três linhas de regressão, uma para cada nível de MOVEDWHEN. (Eu omiti as faixas de confiança aqui, o que desorganiza o gráfico a menos que se use cores.)

É fácil ver porque a interação não é significativa. A correlação positiva entre DISFLUENCY e SOCIALNETWORK é a mesma para cada nível de MOVEDWHEN. Essa correlação positiva como um efeito principal é significativa, mas as diferenças entre os diferentes níveis de MOVEDWHEN – as intercepções – não alcançam níveis padrões de significância. Portanto, uma vez que não removemos a interação aqui (novamente, apenas por razões de exposição), poderíamos chegar a uma conclusão: “No geral, há uma correlação altamente significativa (L.R. $\chi^2=1214,8$; gl = 5; $p < 0,001$) [veja página 298 sobre como calcular isso], porém não particularmente forte ($R^2 = 0,26$). Essa correlação deve-se ao fato de que o número de horas passada em redes sociais correlaciona-se de forma significativamente positiva com os números de disfluências produzidas (L.R. $\chi^2=19,35$; gl = 1; $p < 0,001$), enquanto que a idade que os falantes tinham quando se mudaram para os Estados Unidos não ($p > 0,25$), and neither is their interaction ($p > 0.79$).”

Recomendação para estudo/leitura

- Recomendações de estudo [Gries \(2009a, Seção 5.1\)](#) sobre análise (hierárquica) de frequências configuracional (e um script para computar a análises interativamente) e [Field, Miles e Field \(2012, Seções 18.7-18.12\)](#) sobre análise loglinear; veja também `loglin` e `loglm` (do pacote MASS) para análises loglineares
- [Agresti \(2002, Cap. 4, 8-9\)](#), [Faraway \(2006, Cap. 3-4\)](#), [Zuur, Ieno e Smith \(2007, Seção 6.1\)](#), [Zuur et al. \(2009, Cap. 8-9, 11\)](#), [Hilbe \(2009, Cap. 11\)](#), [Fox e Weisberg \(2011, Seção 5.5-5.6\)](#)

Figura 78 – A interação MOVEDWHEN: SOCIAL NETWORK no model.01

5.4 Medidas repetidas: um indicador

A seção final nesta parte sobre modelagem de regressão é dedicada a um tipo de cenário que se difere de todos os anteriores. Todos os modelos discutidos até agora compartilharam uma mesma pressuposição, a de que os pontos de dados (e seus resíduos) são independentes uns dos outros. Por exemplo, na seção sobre regressões lineares, o tempo médio de reação a cada palavra foi considerado independente do tempo médio de reação de qualquer uma das outras palavras. Este cenário, embora frequente, não é o único possível - como você viu na página 159 acima, os grupos/amostras podem ser dependentes, o que significa que os pontos de dados estão relacionados entre si. As formas mais comuns em que os pontos de dados estão relacionados entre si envolvem os seguintes cenários:

- em contextos experimentais, você obtém mais de uma resposta por participante da pesquisa (isto é, você faz medições repetidas em cada participante, o que significa que as características de um participante afetam mais de um ponto de dados;
- em contextos experimentais, você obtém mais de uma resposta por, digamos, item lexical que você testa com diferentes estímulos, o que significa que as características de um item lexical afetam mais de um ponto de dados;
- em dados de corpus, você obtém mais de um ponto de dados por falante, o que significa ..., você entendeu.

Se seus dados envolvem tais pontos de dados relacionados, mas você ignora isso na sua análise estatística, você corre vários riscos. Primeiro, você corre o risco do que é chamado “perda de poder”, o que significa que você pode aceitar a H_0 embora H_1 seja verdadeira na população (o que é chamado de erro tipo II, um erro tipo I é aceitar H_1 embora H_0 seja verdadeiro na população). Em segundo lugar, você corre o risco de obter resultados incorretos porque sua análise estatística não leva em consideração toda a estrutura dos dados conhecida, e retornará - no contexto de modelagem de regressão - coeficientes que não são tão precisos quanto deveriam ser.

Nesta seção, vou falar sobre os métodos que são usados em tais casos. No entanto, esta seção será muito breve porque, embora sejam muito importantes e poderosos, os métodos que são usados em tais casos também envolvem uma complexidade considerável e exigem muito mais espaço do que eu posso dedicar a eles aqui. (Veja abaixo algumas excelentes referências para estudos posteriores, em particular [Girden \(1992\)](#), que inspirou algumas das discussões aqui, e [Field, Miles e Field \(2012\)](#).) Além disso, embora a lógica geral de medidas repetidas se aplique a muitos tipos e configurações diferentes de variáveis independentes e dependentes, discutirei apenas casos que poderiam ser considerados ANOVAs de medidas repetidas, isto é, casos nos quais a variável dependente é de escala de proporção/intervalo (ou seja, não categórica) e as variáveis independentes são tratadas como categóricas.

5.4.1 Uma variável independente dentro de participantes/itens

A título de introdução, começarei minha discussão aqui com um breve exemplo de três maneiras diferentes de se analisar o tipo mais simples possível de dados de amostras dependentes. Na seção 4.3.2.2, tratamos de um caso como esse quando examinamos se as traduções de 16 textos eram mais longas do que os textos originais ou não. Esse cenário envolveu amostras dependentes porque era possível conectar cada original com sua tradução e nós, portanto, calculamos um teste *t* para amostras dependentes. Vamos limpar a memória, carregar o pacote *ez*, recarregar esses dados (agora de `<_inputfiles/05-5-1_textlengths.csv>`) e então visitar este cenário, aqui, por razões expositivas, como uma hipótese bicaudal (a de que os comprimentos médios dos originais e das traduções são diferentes). Além disso, antes de anexarmos a estrutura de dados, convertemos a coluna `TEXT`, que simplesmente numera os textos, em um fator: esta variável é realmente apenas categórica uma vez que os números não fazem nada além de identificar a que texto pertence um comprimento - os tamanhos dos números não importam.

Código em R

```
> Texts=read.delim(file.choose())  
> Texts$TEXT=factor(Texts$TEXT)  
> str(Texts); attach(Texts)
```

Já sabemos com base no que vimos anterioremente que as diferenças entre os tamanhos dos originais e das traduções são distribuídos de forma normal, então podemos imediatamente calcular um teste *t* para amostras dependentes (novamente, aqui um bicaudal) e obter os valores-*t* e valores de graus de liberdade familiares, bem como um valor-*p* agora apenas marginalmente significativo.

Código em R

```
> t.test(LENGTH~TEXTSOURCE, paired=TRUE)
```

Acima, vimos que um modelo linear com um preditor binário é essencialmente equivalente a um teste *t* para amostras independentes (lembre-se da página 266). Portanto, pode não ser uma grande surpresa que uma ANOVA de medidas repetidas com um preditor binário seja essencialmente equivalente a um teste *t* para amostras dependentes. As duas ANOVAs diferem-se, no entanto, em relação a como a variabilidade nos dados é dividida na análise. Um modelo linear de medidas independentes com um preditor binário ou categórico divide a variabilidade dos dados em variabilidade que é atribuída aos níveis da variável independente e variabilidade que é atribuída à variação aleatória (ruído aleatório, variabilidade residual, ou erro). O efeito da variável independente é então avaliado por meio da comparação entre as duas quantidades de variabilidade, e quanto mais variabilidade a variável independente explicar em comparação com a variabilidade residual, mais provável é que o efeito da variável independente seja significativo.

Em uma ANOVA de medidas repetidas, a variabilidade é dividida de forma diferente. Em primeiro lugar, há variabilidade entre sujeitos, aqui textos, tipicamente diferentes. Mas, então há também variabilidade

dentro de diferentes sujeitos (aqui, textos diferentes), e uma parte dessa variabilidade é devido à variável independente (aqui, TEXTSOURCE: ORIGINAL vs. TEXTSOURCE: TRANSLATION) e o restante é erro aleatório / variação residual. Uma vez que medidas repetidas em ANOVAs o efeito da variável independente se dá dentro dos sujeitos ou, aqui, textos, comparamos a quantidade de variabilidade dentro dos sujeitos/textos que é atribuída à variável independente não para a variabilidade total restante, mas para a quantidade restante de variabilidade dentro dos sujeitos/textos e, novamente, quanto mais a variabilidade interna dos sujeitos é explicada pela variável independente em comparação com a variabilidade residual dentro dos sujeitos, mais provável é que o resultado será significativo. E é por isso que os estudos de amostras dependentes / medidas repetidas podem ser mais precisos: os efeitos de variáveis independentes são comparados a uma quantidade menor de variabilidade residual (dentro do sujeito / texto).

Como fazemos isso no R? Utilizamos a função `aov` (do inglês *analysis of variance* - análise de variância) e dizemos a ela (i) que queremos um modelo no qual LENGTH (comprimento) seja modelado como uma função de TEXTSOURCE (LENGTH ~ TEXTSOURCE, sem surpresas aqui) e (ii) qual é a fonte relevante de erro / variabilidade residual (ERROR (. . .)) declarando que a variável independente TEXTSOURCE está aninhada, isto é, repetida dentro de, cada elemento de TEXT (TEXT/TEXTSOURCE):

Código em R

```
> model.01.aov=aov(LENGTH~TEXTSOURCE + Error(TEXT/TEXTSOURCE))
> summary(model.01.aov)
Error: TEXT
Df Sum Sq Mean Sq F value Pr(>F)
Residuals 15 210479 14032

Error: TEXT:TEXTSOURCE
Df Sum Sq Mean Sq F value Pr(>F)
TEXTSOURCE 1 51040 51040 3.717 0.073 .
Residuals 15 205991 13733
```

Como discutido acima, o resultado divide a variabilidade geral naquela entre sujeitos/textos (isto é, a parte superior rotulada como Error: TEXT) e naquela dentro dos sujeitos/textos, a qual, por sua vez, é causada ou pela variável independente TEXTSOURCE (quadrado médio: 51,040) ou por ruído aleatório/residual (quadrado médio: 13733). O valor- F é então a razão entre os dois quadrados médios em níveis da variável independente menos 1 e os sujeitos/textos menos 1 grau de liberdade. Como você pode ver, esse resultado é então idêntico ao teste t : o valor- F é t^2 , o gl residual do valor- F é o gl dos testes t , os valores- p são idênticos e, obviamente, é a mesma conclusão que você escreveria: Com uma hipótese bicaudal, as médias (`model.tables(model.01.aov, "means")`) não se diferem significativamente uma da outra.

Uma maneira alternativa muito interessante para conduzir uma ANOVA de medidas repetidas envolve a função muito útil `ezANOVA` da biblioteca `ez`. O primeiro argumento (`data`) é a estrutura de dados que contém os dados, o segundo (`dv`) especifica a variável dependente, o terceiro (`wid`) especifica o identificador dos sujeitos/textos e o quarto (`within`) define a variável independente aninhada dentro do identificador. Você obtém uma tabela ANOVA com o mesmo valor- F , seus dois graus de liberdade, o mesmo valor- p e uma medida do tamanho do efeito na coluna rotulada como `ges`. (Além disso, explore o código com `ezPlot` e veja `?ezStats`.)

Código em R

```
> ezANOVA(data=Texts, dv=.(LENGTH), wid=.(TEXT), within=.(TEXTSOURCE))
```


5.4.2 Duas variáveis independentes aninhadas em sujeitos/itens

Como podemos estender a abordagem acima para dados mais complexos, como casos em que duas variáveis estão aninhadas dentro de um sujeito (participante da pesquisa) ou de um item? Considere um caso hipotético em que se pede a cinco sujeitos que forneçam o máximo possível de sinônimos para oito estímulos (diferentes para cada sujeito, de modo que não há repetição de itens), os quais resultam da mistura de palavras com conotações positivas ou negativas (uma variável chamada *MEANING - SIGNIFICADO*) de quatro diferentes classes gramaticais (uma variável chamada *POS*, do inglês *parts of speech*). Vamos supor que desejamos saber se o número de sinônimos que os sujeitos forneceram em 30 segundos variou em função dessas variáveis independentes (e, para simplificar, estamos tratando essas frequências como dados de escala de intervalo/proporção). Vamos carregar os dados de `<_inputfiles/05-5-2_synonyms.csv>`:

Código em R

```
> Syns=read.delim(file.choose())  
> str(Syns); attach(Syns)
```

Neste caso, não há nenhum teste monofatorial disponível para comparação, então imediatamente fazemos a ANOVA de medidas repetidas. A lógica não é na verdade diferente da anterior: queremos estudar os efeitos de ambas as variáveis independentes e de sua interação, mas ambas essas variáveis estão aninhadas dentro de *SUBJECT*. Assim, ou usamos `aov...`

Código em R

```
> model.01.aov=aov(SYNONYMS~MEANING*POS + Error(SUBJECT/(MEANING*POS)))  
> summary(model.01.aov)
```

... ou `ezANOVA`:

Código em R

```
>ezANOVA(data=Syns, dv=.(SYNONYMS), wid=.(SUBJECT),within=.(MEANING, POS))
```

Como antes, ambos retornam os mesmos resultados: O único efeito que atinge níveis de significância padrão é *POS* (classe gramatical), e o resultado de `model.tables` mostra que substantivos e verbos resultaram em grandes números de sinônimos, enquanto que adjetivos e advérbios só produziram números médios e baixos, respectivamente. O resultado de `ezANOVA` também retorna um teste para esfericidade, uma suposição muito importante de ANOVAs de medidas repetidas (ver as recomendações para estudo posterior). Neste caso, todos os valores-*p* são $>0,05$, portanto a esfericidade não é violada e podemos confiar nos resultados de nossos testes *F*.

5.4.3 Duas variáveis independentes, uma entre, uma dentro de sujeitos/itens

Vejamos um último exemplo, no qual eu mostro como lidar com casos onde você tem duas variáveis independentes, mas apenas uma delas é aninhada dentro de sujeitos - a outra varia entre sujeitos. Imagine que você tenha 10 participantes da pesquisa falantes não-nativos, cada um dos quais participou de quatro testes de proficiência: um exame oral, um teste de gramática em sala de aula, um texto escrito em sala de aula e um texto escrito em casa. Esta é a variável aninhada dentro dos sujeitos. No entanto, você também suspeita que o sexo dos falantes exerce alguma influência e, adivinhem, essa variável não está aninhada dentro dos sujeitos ... mas sim entre eles. Vamos carregar os dados de `<_inputfiles/05-5-3_mistakes.csv>`.

Código em R

```
> Mistakes=read.delim(file.choose())
> str(Mistakes); attach(Mistakes)
```

Deve estar claro o que fazer: para aov, você especifica a fórmula com todas as variáveis independentes e diz que apenas TASK (tarefa, aqui o tipo de teste de proficiência) é aninhado em SUBJECTS (participantes da pesquisa).

Código em R

```
> model.01.aov=aov(MISTAKES~SEX*TASK + Error(SUBJECT/TASK))
> summary(model.01.aov)
```

Para ezANOVA, você usa o argumento between para dizer à função que a variável independente SEX não varia dentro dos sujeitos, mas entre eles:

Código em R

```
> ezANOVA(data=Mistakes, dv=.(MISTAKES), wid=.(SUBJECT),within=.(TASK),
between=.(SEX))
```

Infelizmente, enquanto obtemos resultados significativos tanto para TASK quanto para a sua interação com SEX - explore novamente as médias com `model.tables` - desta vez os testes de esfericidade são motivo de alarme. ezANOVA sugere duas correções para violações de esfericidade, as quais ainda retornam valores significativos para TASK e TASK:SEX, mas isso está além do escopo deste livro, veja as recomendações para estudo posterior abaixo e na próxima seção.

5.4.4 Efeitos mixos / Modelos de vários níveis

O que vimos acima já indicou que as ANOVAs de medidas repetidas nem sempre são tão simples de usar como o que vimos anteriormente pode ter feito você esperar. Primeiramente, as ANOVAs de medidas repetidas como discutidas acima envolvem somente variáveis independentes categóricas, mas muitas vezes você pode ter variáveis de escala de intervalo/proporção e pode não querer fatorizá-las (dada a perda de informações e de poder que isso pode acarretar, veja [Baayen \(2010\)](#)). Em segundo lugar, muitas variáveis que você pode querer incluir não são *efeitos fixos* (ou seja, variáveis cujos níveis no estudo cobrem todos os níveis possíveis na população), mas são *efeitos aleatórios* (isto é, variáveis cujos níveis no estudo não abrangem todos os níveis possíveis na população, tais como SUBJECT, ITEM, ..., ver [Gelman e Hill \(2007, p. 245\)](#)). Em terceiro lugar, a ANOVA de medidas repetidas requer um design equilibrado e pode, portanto, ser problemática com dados que estão faltando em experimentos e em dados de observação não balanceados. Finalmente, violações de esfericidade não são sempre fáceis de abordar e sem controvérsias; (Ver [Baguley \(2012, Seção 18.2.2\)](#) para mais discussão).

Uma estratégia para manipular dados com pontos de dados dependentes/relacionados e efeitos aleatórios que é atualmente muito utilizada na linguística é o uso de modelos de efeitos mistos, ou modelos de vários níveis. De forma bem simplificada, estes são modelos de regressão que podem lidar com efeitos fixos e aleatórios, bem como com medidas repetidas, dados não balanceados e dados hierárquicos/aninhados. Eles fazem isso modelando simultaneamente diferentes fontes de variabilidade, por exemplo, ao invés de simplesmente ajustarem uma linha de regressão sobre muitos sujeitos através de uma nuvem de pontos em um sistema de coordenadas, eles permitem ao analista modelar a variável dependente com uma linha de regressão diferente para cada sujeito ou item, onde as diferentes linhas de regressão podem ter, digamos, diferentes interceptos por sujeito específico ou item específico (chamadas de *interceptos aleatórios*, porque são modeladas como uma variável aleatória normalmente distribuída)

e/ou inclinações/desvios *chamadas de inclinações aleatórias*). Por razões de espaço e outras coisas que serão discutidas abaixo, não vou discutir esses modelos altamente complexos aqui em detalhes, mas quero dar dois breves exemplos. Para o primeiro deles, retornarei novamente ao teste *t* para amostras dependentes:

Código em R

```
> rm(list=ls(all=TRUE)); library(effects); library(nlme)¶
> Texts=read.delim(file.choose())¶
> Texts$TEXT = factor(Texts$TEXT)¶
> str(Texts); attach(Texts)¶
```

O pacote *nlme* (assim como o pacote mais recente *lme4*) permite o ajuste de uma grande variedade de modelos de efeitos mistos. Esta é uma forma de aplicar esses modelos para os dados do teste *t*. A função para efeitos mistos lineares é *lme*, e aqui ela toma dois argumentos. O primeiro, *fixed*, define a estrutura de efeitos fixos do modelo, e nossa única variável independente de efeito fixo é *TEXTSOURCE*. O segundo argumento, *random*, descreve a estrutura de efeitos aleatórios do modelo, e a notação significa que queremos que o intercepto(1) seja capaz de variar por *TEXT* (|*TEXT*), o que é apenas outra maneira de capturar a variabilidade específica do texto como fizemos na ANOVA de medidas repetidas.

Código em R

```
> model.01.lme=lme(fixed = LENGTH~TEXTSOURCE, random= ~ 1|TEXT)¶
> summary(model.01.lme)¶
```

O resultado que obtemos contém muitas informações, mas vamos nos concentrar apenas nos efeitos aleatórios e nos efeitos fixos. Os primeiros (na seção “Efeitos aleatórios”) contém uma estimativa da variabilidade dos 16 interceptações aleatórias para os 16 textos, a dizer, um desvio padrão de 12,23172. Os segundos, (na seção “Efeitos fixos: ...”) contém o tipo familiar de tabela de coeficientes, erros padrão, valores-*t* e valores-*p*. O valor-*t* (1,92787), seu *gl* (15) e o valor-*p* (0,073) devem parecer muito familiares, uma vez que correspondem aos resultados acima para os mesmos dados. E você pode até criar o tipo familiar de gráfico de efeitos para este resultado porque a função *effect* aceita modelos *lme* como entrada:

Código em R

```
> plot(effect("TEXTSOURCE", model.01.lme))¶
```

Algumas outras aplicações de ANOVAs de medidas repetidas podem ser exploradas de forma semelhante. Por exemplo, os dados acima sobre os erros podem ser estudados com esta chamada de função:

Código em R

```
> summary(model.01.lme=lme(fixed = MISTAKES ~ SEX*TASK, random= ~ 1|SUBJECT))¶
```

Você pode permitir que os interceptos variem entre os sujeitos da mesma maneira que acima (você também pode permitir que as inclinações variem, mas eu não vou discutir isso aqui), você pode plotar os principais efeitos ou as interações de tais modelos com *effect*, e você pode até aplicar *Anova(model)* a modelos *lme* para obter valores-*p* para os preditores de efeito fixo.

Parece simples, não é? Por que não há uma seção inteira sobre isso, explicando e exemplificando tudo em detalhes, como para os outros modelos neste capítulo? Bem, infelizmente, as coisas estão muito

longe de serem tão simples. De fato, a modelagem de efeitos mistos é uma das mais fascinantes, porém também uma das mais complexas técnicas estatísticas que eu já vi. Neste momento, ela realmente parece ser vista como a melhor opção, e de fato o potencial desta abordagem é imensa e de longo alcance. Dito isto, devo admitir que às vezes eu acho que algumas propagandas sobre este método são um pouco prematuras simplesmente porque muitas coisas ainda não são claras. Pergunte a dois ou três especialistas sobre como fazer X com modelos de vários níveis, e você muitas vezes obterá respostas muito diferentes. Escolha duas ou três referências sobre a modelagem de efeitos mistos e verá que não só há muito pouco acordo sobre algumas questões aparentemente centrais, mas também que alguns tipos de problemas nem sequer são mencionados amplamente. Por exemplo,

- parece que não estamos nem perto de um consenso sobre o que seria um *processo de seleção de modelo* ou mesmo apenas um *modelo máximo*. Algumas fontes recomendam um processo de seleção de modelo onde começamos sem efeitos fixos, mas primeiro exploramos efeitos aleatórios. Outras fontes recomendam começar com um modelo máximo completo de efeitos fixos. Algumas fontes recomendam ainda começar com uma estrutura simples de efeitos aleatórios (apenas interceptos), enquanto outras recomendam começar com uma estrutura máxima de efeito aleatório com intercepções aleatórias e inclinações para tudo (e então simulações sugerem que esses modelos não convergem mesmo se receberem a estrutura do modelo correta)...;
- ainda não está claro como *preditores devem ser selecionados* para retenção em ou exclusão de um modelo: alguns usam valores-*p* (com base nos valores-*t* ou valores-*F*, mas então é debatido como escolher o os graus de liberdade residuais), alguns usam a amostragem MCMC (que não é facilmente disponível para alguns tipos de variáveis dependentes), alguns usam critérios de informação (como AIC ou BIC ou mesmo DIC) para todo o processo, alguns usam testes de razão de verossimilhança, que requerem atenção para saber se os modelos foram ajustados com ML ou REML, ...;
- Muitas referências não discutem como lidar com as *intercorrelações de intercepções e declives aleatórios*;
- muitas referências não dizem praticamente nada sobre como decidir sobre uma *estrutura de covariância dos dados*; Eu acho que eu vi apenas uma referência discutindo isso de forma um tanto acessível;
- você viu que *centrar variáveis* pode ser útil na modelação de regressão, mas como fazer isso da melhor forma em modelos de efeitos mistos também não é bem discutido - quando centralizamos em torno de uma média global ou em torno de uma média do grupo? E a lista continua, efeitos de bordas, como calcular o R^2 s, ...

Nada do que foi colocado acima nega que a modelagem de efeitos mistos é muito poderosa e tem o potencial de nos ajudar muito na análise de nossos dados ... uma vez que a área tenha desenvolvido um pouco mais de senso comum sobre como ela deve ser aplicada aos vários tipos de dados existentes. O fato de que agora alguns periódicos já exigem modelagens de efeitos mistos para conjuntos de dados específicos parece um pouco exagerado, considerando-se quantas perguntas ainda estão em aberto. No entanto, quando começarem a surgir alguns padrões sobre as muitas perguntas abertas e quando forem desenvolvidas algumas bibliotecas e funções que tornam a abordagem de algumas dessas perguntas mais fácil (a [pvals.fnc] de Baayen é um caso desse), então a disciplina se beneficiará de modelos de efeitos misturados de inúmeras maneiras. Até que isso aconteça, aqui estão algumas muito boas, eu acho, referências (com diferentes graus de tecnicidade) que podem lhe dar um auxílio inicial além desta pequena introdução feita aqui...

Recomendação para estudo/leitura

- para ANOVAs com medidas repetidas: Girden (1992), Johnson (2008, Seções 4.3-4.4), e especialmente Field, Miles e Field (2012, Cap. 13-14))
- para modelos de efeitos mistos/de vários níveis: Twisk (2006), Gelman e Hill (2007, Cap. 11-15)), Zuur, Ieno e Smith (2007, Cap. 8), Baayen (2008, Cap. 7), Baayen, Davidson e Bates (2008), Johnson (2008, Seções 7.3, 7.4), Zuur et al. (2009, Cap. 5), Field, Miles e Field (2012, Cap. 19), Baguley (2012, Cap. 18); também veja Baayen (2011)

5.5 Análise hierárquica de *cluster*

Nós até agora só nos preocupamos com métodos nos quais as variáveis independentes e dependentes estavam claramente separadas e onde já tínhamos pelo menos uma expectativa e uma hipótese antes da coleta de dados. Tais métodos são às vezes chamados de *estatísticas de teste de hipóteses*, e usamos estatísticas e valores-*p* para decidirmos se rejeitamos ou não uma H_0 . O método denominado análise hierárquica de cluster com o qual lidamos nesta seção é um método, ou, mais precisamente, uma família de métodos, chamado de *exploratório*, ou *gerador de hipóteses*. Esse método é normalmente usado para dividir um conjunto de elementos em aglomerados, ou grupos, de forma que os membros de um grupo são muito semelhantes entre si e, ao mesmo tempo, muito diferentes dos membros de outros grupos. Uma razão óbvia para se usar as análises de cluster para este fim é que este método pode lidar com grandes quantidades de dados e ser, ao mesmo tempo, mais objetivo do que seres humanos observando grandes tabelas.

Para você obter uma primeira compreensão do que as análises de cluster fazem, vamos olhar para um exemplo fictício de uma análise de cluster baseada em julgamentos de similaridade entre fonemas consonantais da língua inglesa. Vamos supor que você desejou determinar como falantes nativos do inglês distinguem os seguintes fonemas consonantais: /b/, /d/, /f/, /g/, /l/, /m/, /n/, /p/, /s/, /t/, e /v/. Você pediu que 20 sujeitos classifikassem as similaridades de todos os $^{11 \times 10}_2 = 55$ pares de consoantes numa escala de 0 ('completamente diferente') a 1 ('completamente idêntico'). Como resultado, você obteve 20 julgamentos de similaridade para cada par e poderia calcular um julgamento médio para cada par. Você poderia agora calcular uma análise de cluster com base nesses julgamentos de similaridade médios para determinar (i) quais consoantes e grupos de consoantes os sujeitos parecem distinguir e (ii) como esses grupos talvez possam ser explicados. A Figura 79 mostra o resultado que tal análise de cluster poderia produzir - como você o interpretaria?

Pausa para pensar

O "resultado" sugere que os julgamentos dos sujeitos provavelmente foram fortemente influenciados pela forma de articulação das consoantes: em um nível muito geral, há dois grupos, um com /b/, /p/, /t/, /d/, e /g/, e outro com /l/, /n/, /m/, /v/, /f/, e /s/. É imediatamente óbvio que o primeiro conjunto contém todas e apenas as consoantes oclusivas (isto é, consoantes cuja produção envolve uma obstrução momentânea do fluxo de ar) que foram apresentadas aos sujeitos, enquanto que o segundo conjunto contém todas e apenas as consoantes nasais, líquidas e fricativas (isto é, consoantes cuja produção envolve apenas uma obstrução momentânea e parcial do fluxo de ar).

No entanto, há mais o que se concluir dos resultados. O primeiro dos dois grupos tem uma estrutura interna notável com dois "subgrupos". O primeiro subgrupo, por assim dizer, contém todos e apenas os fonemas bilabiais, enquanto que o segundo subgrupo agrupa ambos os sons alveolares e velares.

O segundo dos dois grandes grupos também tem uma estrutura interna com dois subgrupos. O primeiro deles contém todas e apenas os sons nasais e líquidos (ou seja, fonemas que às vezes são classificados como estando entre vogais e consoantes), e novamente os fonemas com o mesmo local

Figura 79 – Resultados fictícios de uma análise de cluster de consoantes do inglês

de articulação são agrupados primeiro (os dois sons alveolares). O mesmo é verdadeiro para o segundo subgrupo, que contém todas e apenas as fricativas, com as fricativas labiodentais combinadas primeiro.

Os comentários acima preocuparam-se apenas com quais elementos são membros de quais grupos. Outras tentativas de interpretação podem focar em quantos agrupamento na Figura 79 são diferentes uns dos outros de forma suficientemente forte para serem considerados agrupamentos independentes. Essa discussão é idealmente baseada em testes subsequentes que são muito complexos para serem discutidos aqui, mas como uma heurística rápida você pode olhar para os comprimentos das linhas verticais em tal diagrama de árvore, ou dendrograma. Linhas verticais longas indicam subgrupos mais autônomos. Por exemplo, o subgrupo /b/ /p/ é bem diferente das demais oclusivas, uma vez que a linha vertical que vai para cima levando à sua fusão com /t/ /d/ /g/ é bem longa.⁹

Infelizmente, as análises de *clusters* geralmente não produzem um resultado tão perfeitamente interpretável, mas esses dendrogramas são muitas vezes surpreendentemente interessantes e reveladores. As análises de *clusters* são frequentemente usadas em estudos da Semântica, da Linguística Cognitiva, da Psicolinguística e da Linguística Computacional (ver Miller (1971), Sandra e Rice (1995), Rice (1996) e Manning, Schütze e SCHUTZE (2000, Cap. 14) para alguns exemplos) e são frequentemente a forma ideal para detectar padrões em conjuntos de dados grandes e aparentemente cheios de ruídos/caóticos. Você deve perceber, no entanto, que mesmo se as análises de cluster permitirem uma identificação objetiva de grupos, o analista ainda deve tomar pelo menos três decisões potencialmente subjetivas. As primeiras duas influenciam a aparência exata do dendrograma; a terceira você já viu: deve-se decidir o que o dendrograma reflete. No que se segue, vou mostrar-lhe como você mesmo pode fazer tal análise com o R. As análises de agrupamentos hierárquicos geralmente envolvem as seguintes etapas:

⁹ Para um exemplo semelhante mas autêntico (baseado em dados sobre formantes de vogal), ver Kornai (1998).

Tabela 47 – Frequências esquemáticas de co-ocorrência de sete palavras do inglês no BNC

Collocate	<i>bronze</i>	<i>gold</i>	<i>silver</i>	<i>bar</i>	<i>cafe</i>	<i>menu</i>	<i>restaurant</i>
X	0	0	0	3	0	0	0
Y	0	11	0	0	0	0	1
Z	0	1	1	0	0	0	1
A	0	0	0	1	0	2	0
B	1	0	0	1	0	0	0
C	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...

Procedimentos

Tabulação dos dados

- Cálculo de uma matriz de similaridade/dissimilaridade com base numa métrica de similaridade/dissimilaridade definida pelo usuário
- Cálculo de uma estrutura de cluster com base numa regra de amalgamação definida pelo usuário
- Representação da estrutura do cluster em um dendrograma e interpretação do mesmo
- Exploração *post-hoc* (como larguras de silhueta médias)

O exemplo que vamos discutir é do domínio da Linguística Computacional/de Corpus. Em ambas as disciplinas, o grau de semelhança entre duas palavras é muitas vezes aproximado com base no número e frequência de colocados compartilhados. Uma definição muito vaga de “colocados e uma palavra w/” é “as palavras que ocorrem frequentemente no ambiente de w”, onde *ambiente*, por sua vez, é muitas vezes definido como ‘na mesma frase’ ou dentro de uma janela de 4 ou 5 palavras ao redor de w. Por exemplo: se você encontrar a palavra “carro” em um texto, muitas vezes palavras como *motorista*, *motor*, *gasolina* e/ou *acidente* estão relativamente próximas dela, enquanto que palavras como *farinha*, *paz*, *tratado*, *ditatorial* e *coleção de cactos* provavelmente não são frequentes perto de “carro”. Em outras palavras, quanto mais colocados duas palavras x e y compartilham, mais provavelmente há uma relação semântica entre elas (ver [Oakes \(1998, Cap. 3\)](#), [Manning, Schütze e SCHUTZE \(2000, Seção 14.1 e 15.2\)](#) bem como [Gries \(2009a\)](#) para saber como obter colocados).

No presente exemplo, consideramos as seguintes sete palavras do inglês: *bronze* (*bronze*), *gold* (*ouro*), *silver* (*prata*), *bar* (*bar* ou *barra*), *cafe* (*cafeteria* ou *lanchonete*), *menu* (*cardápio*) e *restaurant* (*restaurante*). Obviamente, eu não escolhi essas palavras de forma aleatória - eu as escolhi porque elas intuitivamente caem em dois grupos com a palavra “bar” (e, portanto, constituem um bom caso de teste).

Um cluster consiste em três co-hipônimos do metal, o outro consiste em três co-hipônimos de *estabelecimento gastronômico* assim como uma palavra do mesmo campo semântico. Suponhamos que você tenha extraído do British National Corpus (BNC) todas as ocorrências dessas palavras e seus colocados que sejam palavras lexicais (isto é, substantivos, verbos, adjetivos e advérbios). Para cada colocado que ocorreu com pelo menos uma das sete palavras, você determinou com que frequência ele ocorreu com cada uma das sete palavras. A Tabela 46 é uma representação esquemática das primeiras seis linhas de uma tabela com essas informações. O primeiro colocado, aqui referido como X, co-ocorreu somente com “bar” (três vezes); A segunda colocação, Y, co-ocorreu 11 vezes com *gold* e uma vez com *restaurante*, etc.

Queremos agora saber quais as palavras são mais semelhantes entre si do que a outras. Ou seja, como no exemplo acima, você quer agrupar elementos - acima, fonemas, aqui, palavras - com base em propriedades - acima, julgamentos médios de similaridade, aqui, frequências de co-ocorrência. Primeiro,

Tabela 48 – Combinações de características de dois elementos binários

	Elemento 2 exibe a característica x	Elemento 2 não exibe a característica x
Elemento 1 exibe a característica x	a	b
Elemento 1 não exibe a característica x	c	d

você precisa de um conjunto de dados como a Tabela 46, o que você pode obter a partir do arquivo <_inputfiles/05-6_collocates.RData>, o qual contém uma grande tabela de dados de co-ocorrência - sete colunas e aproximadamente 31.000 linhas.

Código em R

```
> load(file.choose()) # load the data frame
> ls() # check what was loaded
[1] "collocates"
> str(collocates)
'data.frame': 30936 obs. of 7 variables:
 $ bronze : num 0 0 0 0 1 0 0 0 0 0 ...
 $ gold   : num 0 11 1 0 0 0 0 1 0 0 ...
 $ silver : num 0 0 1 0 0 0 0 0 0 0 ...
 $ bar     : num 3 0 0 1 1 1 1 0 1 0 ...
 $ cafe    : num 0 0 0 0 0 0 0 0 0 1 ...
 $ menu    : num 0 0 0 2 0 0 0 0 0 0 ...
> attach(collocates)
$ restaurant: num 0 1 0 0 0 0 0 0 0 0 ...
```

Alternativamente, você poderia carregar esses dados com `read.table(...)` a partir do arquivo <_inputfiles/05-6_collocates.csv>. Se os seus dados contêm *dados ausentes*, você deve ignorar os dados. Não há dados faltando, mas é útil de conhecer a função (ver a recomendação no final do Capítulo 2):

Em seguida, você deve gerar uma matriz de similaridade/dissimilaridade para as sete palavras. Aqui, você tem que fazer a primeira decisão possivelmente subjetiva, decidir sobre uma medida de similaridade/dissimilaridade. Você precisa considerar dois aspectos: o nível de medição das variáveis em questão e a definição de semelhança a ser utilizada. No que se refere à primeira questão, apenas faremos distinção entre variáveis binárias/nominais e variáveis de escala de proporção. Discutirei medidas de similaridade/dissimilaridade para ambos os tipos de variáveis, mas depois focalizarei nas variáveis de escala de proporção.

No caso das variáveis nominais, dois elementos podem ser semelhantes ou diferentes um do outro de quatro formas diferentes, representadas na Tabela 47. Com base na Tabela 47, a similaridade entre dois elementos é tipicamente quantificada usando a fórmula (66), na qual w_1 e w_2 são definidos pelo analista:

$$(66) \quad \frac{a + w_1 d}{(a + w_1 d)(w_2(b + c))}$$

Três medidas de similaridade devem ser mencionadas aqui:

- o coeficiente de Jaccard: $w_1 = 0$ e $w_2 = 1$;
- o coeficiente de correspondência simples: $w_1 = 1$ e $w_2 = 1$;
- o coeficiente de Dice: $w_1 = 0$ e $w_2 = 0,5$.

Quais são seus coeficientes de similaridade pareados para estes três vetores?

Código em R

```
> aa=c(1, 1, 1, 1, 0, 0, 1, 0, 0, 0)¶
> bb=c(1, 1, 0, 1, 0, 1, 0, 1, 0, 1)¶
> cc=c(1, 0, 1, 1, 1, 1, 1, 1, 1, 0)¶
```

Pausa para pensar

- coeficiente de Jaccard: para aa e bb: 0,375, para aa e cc 0,444, para bb e cc 0,4;
- coeficiente de correspondência simples: para aa e bb: 0,5, para aa e cc 0,5, para bb e cc 0,4;
- coeficiente de Dice: para aa e bb: 0,545, para aa e cc 0,615, para bb e cc 0,571 (veja o arquivo de códigos para uma função faz esses cálculos).

Mas quando e qual dos três usar? Uma regra de ouro é que quando a presença de uma característica é tão informativa quanto a sua ausência, então você deve usar o coeficiente de correspondência simples, caso contrário, escolha o coeficiente de Jaccard ou o coeficiente de Dice. A razão para isso é que, como você pode ver na fórmula (66) e nas definições das medidas acima, apenas o coeficiente de correspondência simples inclui totalmente os casos em que ambos os elementos exibem ou não a característica em questão.

Para as variáveis de proporção-escala, há (muitas) outras medidas, e eu não posso discutir todas aqui. Vou focar em (i) um conjunto de medidas de distância ou dissimilaridade (ou seja, medidas em que grandes valores representam grandes graus de dissimilaridade) e em (ii) um conjunto de medidas de similaridade (isto é, medidas onde grandes valores representam grandes graus de similaridade). Muitas medidas de distância são novamente baseadas em uma fórmula e, portanto, são diferentes em termos de configurações de parâmetros. Esta fórmula básica é a chamada métrica de Minkowski representada em (67).

$$(67) \quad \left(\sum_{i=1}^n |x_{qi} - x_{ri}|^y \right)^{1/y}$$

Quando y é definido como 2, você obtém a chamada distância euclidiana.¹⁰ Se você inserir $y = 2$ em (67) para calcular a distância euclidiana entre os vetores aa e bb, você obtém:

Código em R

```
> sqrt(sum((aa-bb)^2))¶
[1] 2.236068
```

Quando y é definido como 1, você obtém a chamada distância de Manhattan ou de City-Block entre os vetores acima. Para aa e bb, você obtém:

Código em R

```
> sum(abs(aa-bb))¶
[1] 5
```

As medidas de similaridade são medidas correlacionais. Uma delas você já conhece: o coeficiente de correlação produto-momento de Pearson r . Uma medida similar usada frequentemente na Linguística

¹⁰ A distância euclidiana de dois vetores de comprimento n é a distância espacial direta entre dois pontos dentro de um espaço n -dimensional. Isto pode soar complexo, mas para o caso mais simples de um sistema de coordenadas bidimensional esta é simplesmente a distância que você mediria com uma régua.

Computacional é o cosseno (ver Manning e Schütze 1999: 299-303). O cosseno e todas as outras medidas para proporção-escala estão disponíveis na função `Dist` da biblioteca `amap`¹¹. Essa função requer que (i) os dados estejam disponíveis na forma de uma matriz ou de uma estrutura de dados e que (ii) os elementos cujas semelhanças que você procura estejam nas linhas, e não nas colunas como de costume. Se esta última condição não for satisfeita, muitas vezes você pode simplesmente transpor uma estrutura de dados (com `t`):

Código em R

```
» library(amap)¶
  collocates.t=t(collocates)¶
```

Você pode então aplicar a função `Dist` à estrutura de dados transposta. Essa função toma os seguintes argumentos:

- `x`: a matriz ou estrutura de dados para a qual você quer as medidas;
- `method="euclidean"` para a distância euclidiana; `method="manhattan"` para a métrica de City-Block; `method="correlation"` para a correlação r de produto-momento (mas veja mais abaixo!); `method="pearson"` para o cosseno (mas veja mais abaixo!) (há mais algumas medidas disponíveis que eu não vou discutir aqui);
- `diag=FALSE` (o padrão) ou `diag=TRUE`, dependendo se a matriz de distância deve conter sua diagonal principal ou não;
- `upper=FALSE` (o padrão) ou `upper=TRUE`, dependendo se a matriz de distância deve conter somente a metade inferior esquerda ou ambas as metades.

Assim, se você deseja gerar uma matriz de distância baseada em distâncias euclidianas para nosso conjunto de dados de colocados, basta digitar:

Código em R

```
> Dist(collocates.t, method="euclidean", diag=TRUE, upper=TRUE)¶
```

Como você pode ver, você obtém uma matriz de distância (simétrica) na qual a distância de cada palavra para si mesma é, naturalmente, 0. Essa matriz agora lhe diz qual palavra é mais semelhante a qual palavra. Por exemplo, a palavra *silver* (prata) é mais parecida com *cafe* (cafeteria, lanchonete) porque a distância entre *silver* e *cafe* (2385,566) é a menor distância entre *silver* e qualquer outra palavra.

O seguinte comando calcula uma matriz de distância usando a métrica City-Block:

Código em R

```
> Dist(collocates.t, method="manhattan", diag=TRUE, upper=TRUE)¶
```

Para obter uma matriz de similaridade com correlações de produto-momento ou cossenos, você deve calcular a diferença entre 1 menos os valores na matriz. Para obter uma matriz de similaridade com coeficientes de correlação, insira, portanto, isto:

¹¹ A função `dist` da instalação padrão do R também permite calcular várias medidas de similaridade/dissimilaridade, mas menos do que `Dist` da biblioteca `amap`.

Código em R

```
> 1-Dist(collocates.t, method="correlation", diag=TRUE, upper=TRUE)¶
```

	bronze	gold	silver	bar	cafe	menu	restaurant
bronze	0.0000	0.1342	0.1706	0.0537	0.0570	0.0462	0.0531
gold	0.1342	0.0000	0.3103	0.0565	0.0542	0.0458	0.0522
silver	0.1706	0.3103	0.0000	0.0642	0.0599	0.0511	0.0578
bar	0.0537	0.0565	0.0642	0.0000	0.1474	0.1197	0.2254
cafe	0.0570	0.0542	0.0599	0.1474	0.0000	0.0811	0.1751
menu	0.0462	0.0458	0.0511	0.1197	0.0811	0.0000	0.1733
restaurant	0.0531	0.0522	0.0578	0.2254	0.1751	0.1733	0.0000

Você pode verificar os resultados comparando este resultado com aquele que você obtém de `cor(collocates)`. Para uma matriz de similaridade com cossenos, digite:

Código em R

```
> 1-Dist(collocates.t, method="pearson", diag=TRUE, upper=TRUE)¶
```

Existem também programas estatísticos que usam $1-r$ como medida de distância. Eles mudam a medida de similaridade r (valores próximos a zero significam baixa similaridade) para uma medida de distância (valores próximos a zero significam alta similaridade).

Se você comparar a matriz com distâncias euclidianas e a matriz com r , você pode notar algo que lhe parece estranho ...

Pausa para pensar

Na matriz de distância, valores pequenos indicam alta similaridade e o menor valor na coluna *bronze* está na linha para *cafe* (1734,509). Na matriz de similaridade, os valores grandes indicam alta similaridade e o maior valor na coluna *bronze* está na linha para *silver* (cerca de 0,1706). Como pode ser? Esta diferença mostra que mesmo uma abordagem algorítmica de cluster é influenciada por decisões subjetivas, embora esperançosamente motivadas. A escolha pela uma determinada métrica influencia os resultados porque existem diferentes formas em que os vetores podem ser semelhantes uns aos outros. Considere como um exemplo o seguinte conjunto de dados, que também é representado graficamente na Figura 80.

Código em R

```
> y1=1:10; y2=11:20; y3=c(6, 6, 6, 5, 5, 5, 4, 4, 4, 3)¶
> y=t(data.frame(y1, y2, y3))¶
```

A questão é: quão semelhante y_1 é em relação a y_2 e a y_3 ? Há duas maneiras óbvias de considerar a similaridade. Por um lado, y_1 e y_2 são perfeitamente paralelos, mas estão longe um do outro (tanto quanto se pode dizer sobre um diagrama cujas dimensões não são definidas). Por outro lado, y_1 e y_3 não são de forma alguma paralelos um ao outro, mas estão próximos um do outro. As duas abordagens discutidas acima baseiam-se nessas diferentes perspectivas. As medidas de distância mencionadas (como a distância euclidiana) são baseadas na distância espacial entre vetores, que é pequena entre y_1 e y_3 , mas grande entre y_1 e y_2 . As medidas de similaridade discutidas (como o cosseno) baseiam-se na similaridade da curvatura dos vetores, que é pequena entre y_1 e y_3 , mas grande entre y_1 e y_2 . Você pode ver isso rapidamente a partir dos valores numéricos reais:

Figura 80 – Três vetores fictícios

Código em R

```
> Dist(y, method="euclidean", diag=TRUE, upper=TRUE)¶
      y1      y2      y3
y1  0.00000 31.62278 12.28821
y2 31.62278  0.00000 35.93049
y3 12.28821 35.93049  0.00000
> 1-Dist(y, method="pearson", diag=TRUE, upper=TRUE)¶
      y1      y2      y3
y1 0.0000000 0.9559123 0.7796728
y2 0.9559123 0.0000000 0.9284325
y3 0.7796728 0.9284325 0.0000000
```

De acordo com a distância euclidiana, y_1 é mais semelhante a y_3 do que a y_2 - $12,288 < 31,623$ - mas o inverso é verdadeiro para o cosseno: y_1 é mais semelhante a y_2 - $0,956 > 0,78$. As duas medidas baseiam-se em diferentes conceitos de similaridade. O analista deve decidir o que é mais relevante: distâncias espaciais baixas ou curvaturas semelhantes. Por agora, vamos supor que você deseja adotar uma abordagem baseada na curvatura e usar $1-r$ como medida. Em seus próprios estudos, é claro que

você também deve indicar qual medida de similaridade/distância você usou.¹²

Código em R

```
> dist.matrix=Dist(collocates.t, method="correlation",diag=TRUE, upper=TRUE)¶
> round(dist.matrix, 4)¶
```

	bronze	gold	silver	bar	cafe	menu	restaurant
bronze	0.0000	0.8658	0.8294	0.9463	0.9430	0.9538	0.9469
gold	0.8658	0.0000	0.6897	0.9435	0.9458	0.9542	0.9478
silver	0.8294	0.6897	0.0000	0.9358	0.9401	0.9489	0.9422
bar	0.9463	0.9435	0.9358	0.0000	0.8526	0.8803	0.7746
cafe	0.9430	0.9458	0.9401	0.8526	0.0000	0.9189	0.8249
menu	0.9538	0.9542	0.9489	0.8803	0.9189	0.0000	0.8267
restaurant	0.9469	0.9478	0.9422	0.7746	0.8249	0.8267	0.0000

O próximo passo é calcular uma estrutura de cluster dessa matriz de similaridade. Você faz isso com a função `hclust`, que pode tomar até três argumentos, dos quais vou discutir dois. O primeiro é uma matriz de similaridade/distância, o segundo escolhe uma regra de junção que define como os elementos nessa matriz se unem em *clusters*. Esta escolha é a segunda decisão potencialmente subjetiva e há novamente várias possibilidades.

A escolha `method="single"` usa o chamado *método de ligação única* (*single-linkage method*) ou *método do vizinho mais próximo* (*nearest-neighbor method*). Neste método, a similaridade dos elementos x e y - onde x e y podem ser elementos como consoantes individuais ou subconjuntos como $/b/$, $/p/$ na Figura 79 - é definida como a distância *mínima* entre qualquer elemento de x e qualquer elemento de y . No presente exame, isto significa que, na primeira fase de amalgamação, *gold* e *silver* seriam unidos uma vez que sua distância é a menor em toda a matriz ($1-r = 0,6897$). Depois, *bar* se junta a *restaurant* ($1-r = 0,7746$). Em seguida, e agora vem a parte interessante, *bar restaurant* se junta com *cafe* porque a menor distância restante é aquela que *restaurant* exibe em relação a *cafe*: $1-r = 0,8249$. E assim por diante. Esse método de fusão é bom para a identificação de *outliers* (dados extremos) nos dados, mas tende a produzir longas cadeias de *clusters* e não é, portanto, muitas vezes particularmente discriminatório.

A escolha `method="complete"` usa o chamado *método de ligação completa* (*complete-linkage method*) ou *método do vizinho mais distante* (*furthest-neighbor method*). Ao contrário do método de ligação única, aqui a semelhança entre x e y é definida como a distância *máxima* entre qualquer elemento de x e qualquer elemento de y . Primeiramente, *gold* e *silver* são unidos como antes, depois *bar* e *restaurant*. Na terceira etapa, *bar restaurant* se junta com *cafe*, mas a diferença em relação ao método de ligação única é que a distância entre os dois é agora 0,8526, não 0,8249, porque desta vez o algoritmo considera as distâncias máximas, dentre as quais a menor é escolhida para a fusão. Essa abordagem tende a formar grupos homogêneos menores e é um bom método se você suspeitar que há muitos grupos menores em seus dados.

Finalmente, a escolha `method="ward"` usa um método cuja lógica é semelhante à das ANOVAs porque reúne aqueles elementos cuja união aumenta menos a soma de erros de quadrados. Para cada amalgamação possível, o método calcula as somas das diferenças/desvios quadrados da média do cluster potencial, e então o agrupamento com a menor soma de desvios quadrados é escolhido. Este método é conhecido por gerar *clusters* menores que possuem frequentemente tamanhos semelhantes e tem-se mostrado bastante útil em muitas aplicações. Vamos usá-lo aqui também, e, novamente, você deve declarar explicitamente qual regra de fusão você usou em seus próprios estudos. Agora você pode calcular a estrutura do cluster e plotá-la.

¹² Estou simplificando muito aqui: as frequências não são nem normalizadas nem registradas/compensadas etc. (ver acima, Manning, Schütze e SCHUTZE (2000, Seção 15.2.2), ou Jurafsky e Martin (2008, Cap. 20)).

Código em R

```
> clust.ana=hclust(dist.matrix, method="ward")  
> plot(clust.ana)  
> rect.hclust(clust.ana, 2) # red boxes around clusters
```

Figura 81 – Dendrograma de sete palavras do inglês

Este é um resultado cuja interpretação é mais clara do que o usual. Como seria de esperar, as sete palavras caem exatamente nos dois principais *clusters* esperados: um com os “metais” e outro com as palavras relacionadas à gastronomia. O primeiro tem uma subestrutura em que *bronze* é um pouco menos semelhante aos outros dois metais, e o último tem pouca subestrutura mas agrupa os três co-hipôminos antes de *menu* ser adicionado. Com a seguinte linha, você pode fazer o R mostrar a qual grupo cada elemento pertence quando você assume dois *clusters*.

Código em R

```
> cutree(clust.ana, 2)¶
bronze      gold  silver      bar      cafe      menu restaurant
1           1       1       2       2       2           2
```

Embora eu não possa discutir o método em detalhes, quero dar-lhe pelo menos uma breve noção de como estruturas de cluster mais difíceis podem ser exploradas. Como você vai se lembrar, a Figura 79 foi um caso muito menos claro em termos de quantos *clusters* devem ser distinguidos: qualquer número entre 2 e 5 parece defensável. A função `cluster.stats` da biblioteca `fpc` oferece uma variedade de estatísticas de validação, o que pode ajudar a diminuir o número de *clusters* melhor distinguidos. Uma delas envolve a noção de largura média de de silhueta, que quantifica quão semelhantes os elementos são aos *clusters* onde estão em comparação a quão semelhantes os elementos são a outros *clusters*. É então possível calcular larguras médias de silhueta para todas as soluções de cluster possíveis e escolher a que tem as maiores larguras médias de silhueta. Se aplicarmos essa lógica à Figura 79, obtemos a Figura 82. Ela mostra por que a decisão para um número qualquer de *clusters* é tão difícil - muitas soluções são quase igualmente boas - mas por que quatro *clusters* devem ser distinguidos: com quatro *clusters*, a largura média da silhueta é mais alta: 0,14.

Figura 82 – Largura média da silhueta para todas as soluções de *cluster* da Figura 79

Agora você deve fazer os exercícios para o Capítulo 5 ...

Recomendação para estudo/leitura

- a função `daisy` (da biblioteca `cluster`) para calcular matrizes de distância para conjuntos de dados com variáveis de diferentes níveis de medição
- a função `kmeans` para fazer análises de cluster nas quais você fornecer o número de *clusters* de antemão
- a função `pvclust` (da biblioteca `pvclust`) para obter valores-*p* para *clusters* com base em métodos de reamostragem; ver também `pvrect` e `pvpick` (da mesma biblioteca)
- a função `varclus` (da biblioteca `Hmisc`) para fazer clustering variável
- a função `nj` (da biblioteca `ape`) para realizar análise de agrupamentos de vizinhos (*neighbor clustering*) e análise filogenética de cluster
- [Crawley \(2007, Cap. 23\)](#), [Baayen \(2008, Cap. 5\)](#), [Johnson \(2008, Cap. 6\)](#)

Epílogo

Agora que você já quase completou o livro todo, deixe-me dar a você um pouco mais de ideias para reflexão e informações adicionais. Ironicamente, algumas delas provavelmente irão “abalar” um pouco o que você já aprendeu até agora, mas eu espero que elas também possam estimular alguma curiosidade por outras coisas que se pode descobrir e explorar.

Permita-me primeiramente mencionar algumas áreas que você deve começar a explorar na medida em que você se torna mais familiarizado com modelos de regressão. Uma questão que eu só mencionei de passagem no arquivo de códigos é a de (cross)validação. Regressões frequentemente correm o risco do que é chamado de *overfitting*¹: elas se adequam muito bem para um determinado conjunto de dados, mas fazem generalizações ruins para outros, o que naturalmente compromete a possibilidade de generalização dos resultados para a população como um todo. Muitas vezes, os resultados podem ser validados dividindo-se a amostra existente em, frequentemente, 10 partes e, em seguida, fazendo-se 10 análises, sendo que para cada uma delas você obtém uma equação de regressão de 90% dos dados e a aplica para os 10% restantes. Tais métodos podem revelar muito sobre a estrutura interna de um conjunto de dados a há várias funções disponíveis no R para esses métodos. Um ponto relacionado é que, dado o poder cada vez maior dos computadores, abordagens de reamostragem e permutação tornam-se cada vez mais populares; exemplos incluem o *bootstrap*, o procedimento de *jackknife*, ou procedimentos de permutação exaustivos. Esses procedimentos são métodos não paramétricos que podem ser usados para estimar médias e variâncias, mas também correlações ou parâmetros de regressão sem grandes hipóteses de distribuição. Tais métodos não são a solução para todos os problemas estatísticos, mas mesmo assim podem ser ferramentas interessantes e poderosas (cf. as `library boot`, bem como de `bootstrap`).

Recomendação para estudo/leitura

Good (2005), Rizzo (2007, Cap. 7, 8)

Além disso, a análise de pontos de dados especiais em sua(s) amostra(s) é muito importante, dado o impacto que *outliers* (dados extremos) e pontos influentes podem ter sobre os dados. Similarmente, aprender mais sobre o que fazer com dados que estão faltando deve ter prioridade em sua lista de coisas a fazer. Por outro lado, pode ser útil, por exemplo, executar uma regressão em uma base de dados incompleta ou com dados incompletos para ver se há algo nos dados que lhe permite prever também quando, digamos, indivíduos de fato respondem a um estímulo. Por outro lado, pequenas proporções de dados que estão faltando podem ser imputadas, isso está previsto a partir de outros pontos de dados (ver Torgo (2010, Seção 2.5)).

¹ sobreajustamento

Em seguida, há uma variedade de técnicas adicionais que você pode querer explorar. Este livro focou em abordagens de teste de hipóteses, em regressões específicas, mas existem muitas ferramentas exploratórias interessantes que, por razões de espaço, eu não pude discutir: *análise de componentes principais* e *análise de correspondência* são dois exemplos bem conhecidos, *regras de associação* ou *classificadores ingênuos de Bayes* são outros.

Também é importante ressaltar que o R tem muito mais possibilidades de representação gráfica do que eu poderia mencionar aqui. Eu só usei o sistema gráfico tradicional, mas existem outras ferramentas mais poderosas, que estão disponíveis a partir das bibliotecas `lattice` e `ggplot2` (você deve explorar <http://www.yeroon.netggplot2>). O site <http://gallery.renthusiasts.com> fornece muitos exemplos muito interessantes e impressionantes para gráficos do R, e vários bons livros ilustram muitas das empolgantes possibilidades para exploração (ver [Unwin, Theus e Hofmann \(2007\)](#), [Cook et al. \(2007\)](#), [Sarkar \(2008\)](#), [Keen \(2010\)](#), e, claro, [Murrell \(2011\)](#)).

Finalmente, note que o paradigma do *teste de significância da hipótese nula* (TEHN) que está subjacente à maioria dos métodos discutidos aqui não é tão incontroverso como este livro (e mais outros) pode fazer você acreditar. Enquanto o cálculo de valores- p é certamente ainda a abordagem padrão, há pesquisadores que defendem uma perspectiva diferente. Alguns deles argumentam que os valores- p são problemáticos porque eles de fato não representam a probabilidade condicional na qual se está realmente interessado. Lembre-se, os valores- p acima respondem a pergunta “Qual é a probabilidade de se obter os dados observados quando a H_0 é verdadeira?”, mas o que realmente se quer saber é “Qual é a probabilidade da H_1 considerando os dados que tenho?”. Sugestões de melhoria incluem:

- deve-se concentrar não nos valores- p , mas nos tamanhos de efeito e/ou intervalos de confiança (e é por isso que eu os mencionei acima repetidamente);
- deve-se informar os chamados valores- p_{rep} , os quais, de acordo com [Killeen \(2005\)](#), fornecem a probabilidade de se replicar um efeito observado (mas os quais não são incontroversos);
- deve-se testar hipóteses nulas razoáveis ao invés de hipótese que nem sequer poderiam ser verdadeiras (sempre haverá algum efeito ou diferença).

Outra abordagem interessante é a chamada abordagem Bayesiana para a estatística, a qual nos permite incluir conhecimentos prévios subjetivos ou resultados anteriores nos nossos próprios dados. Vale a pena explorar todas essas coisas.

Recomendações para um estudo mais aprofundado

- [Cohen \(1994\)](#), [Loftus \(1996\)](#), [Denis \(2003\)](#) para discussão do NHST (Null Hypothesis Significance Testing - Teste de Significância de Hipótese Nula)
- [Killeen \(2005\)](#) sobre valores- p_{rep}
- [Iversen \(1984\)](#) sobre a estatística de Bayes

Espero que você possa usar as técnicas abordadas neste livro para muitas questões diferentes, e quando este pequeno epílogo também fizer você tentar aumentar o seu conhecimento e se familiarizar com ferramentas e métodos adicionais - por exemplo, há muitos recursos excelentes da web, <http://www.statmethods.netindex.html> e <http://www.r-bloggers.com> estão entre os meus favoritos - então este livro terá alcançado um dos seus principais objetivos.

Referências

- AGRESTI, A. **Categorical Data Analysis (2nd Edition)**. 2. ed. Hoboken, NJ: John Wiley and Sons, 2002.
- AGRESTI, A. **Analysis of Ordinal Categorical Data**. 2. ed. Hoboken, NJ: John Wiley and Sons, 2010. (Wiley Series in Probability and Statistics).
- ANSCOMBE, F. J. Graphs in statistical analysis. **The American Statistician**, v. 27, n. 1, p. 17–21, 1973.
- BAAYEN, R. H. **Analyzing Linguistic Data: A Practical Introduction to Statistics using R**. Cambridge: Cambridge University Press, 2008.
- BAAYEN, R. H. A real experiment is a factorial experiment? v. 5, p. 149–157, 2010.
- BAAYEN, R. H. Corpus linguistics and naive discriminative learning. **Revista Brasileira de Linguística Aplicada**, v. 11, n. 2, p. 295–328, 2011.
- BAAYEN, R. H.; DAVIDSON, D. J.; BATES, D. M. Mixed-effects modeling with crossed random effects for subjects and items. **Journal of Memory and Language**, v. 59, n. 4, p. 390–412, 2008.
- BAGULEY, T. **Serious Stats: A guide to advanced statistics for the behavioral sciences**. Hampshire: Palgrave Macmillan, 2012.
- BENCINI, G. M.; GOLDBERG, A. E. The contribution of argument structure constructions to sentence meaning. **Journal of Memory and Language**, v. 43, n. 4, p. 640–651, 2000.
- BEREZ, A. L.; GRIES, S. T. Correlates to middle marking in dena'ina iterative verbs. **International Journal of American Linguistics**, v. 76, n. 1, p. 145–165, 2010.
- BORTZ, J. **Statistik: für Human- und Sozialwissenschaftler**. Heidelberg: Springer Medizin Verlag, 2005.
- BORTZ, J.; DÖRING, N. Book. **Forschungsmethoden und Evaluation**. New York: Springer Berlin Heidelberg, 1995.
- BORTZ, J.; LIENERT, G.; BOEHNKE, K. **Verteilungsfreie Methoden in der Biostatistik**. Berlin, Heidelberg, New York: Springer, 1990.
- BRAUN, W.; MURDOCH, D. **A First Course in Statistical Programming with R**. Cambridge: Cambridge University Press, 2008.
- BRETZ, F.; HOTHORN, T.; WESTFALL, P. **Multiple Comparisons Using R**. Boca Raton, FL: Chapman and Hall/CRC, 2011.
- BREW, C.; MCKELVIE, D. Word-pair extraction for lexicography. In: PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON NEW METHODS IN LANGUAGE PROCESSING. Ankara: Bilkent University, 1976. p. 45–55.
- CHEN, P. Discourse and particle movement in English. **Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"**, v. 10, n. 1, p. 79–95, 1986.
- COHEN, J. The earth is round ($p < .05$). **American Psychologist**, v. 49, p. 997–1003, 1994.
- COOK, D. et al. **Interactive and Dynamic Graphics for Data Analysis: With R and GGobi**. New York: Springer, 2007.

- COWART, W. **Experimental syntax: applying objective methods to sentence judgements**. Thousand Oaks, CA: Sage Publications, 1997.
- CRAWLEY, M. J. **Statistical computing: An Introduction to data analysis using S-Plus**. Chichester: John Wiley, 2002. v. 167.
- CRAWLEY, M. J. **Statistics: An Introduction using R**. Chichester: John Wiley & Sons, 2005.
- CRAWLEY, M. J. **The R Book**. [S.l.]: Chichester: John Wiley, 2007.
- DALGAARD, P. **Introductory Statistics with R**. New York: Springer, 2002.
- DENIS, D. J. **Alternatives to null hypothesis significance testing alternatives to null hypothesis significance testing**. [s.n.], 2003. v. 4. 1-23 p. Disponível em: <http://theoryandscience.icaap.org/content/vol4.1/02_denis.html>.
- FARAWAY, J. **Linear Models with R**. Boca Raton: Chapman and Hall/CRC, 2005.
- FARAWAY, J. **Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models**. [S.l.]: Chapman and Hall/CRC, 2006. (Chapman & Hall/CRC Texts in Statistical Science).
- FIELD, A.; MILES, J.; FIELD, Z. **Discovering Statistics Using R**. Los Angeles and London: SAGE Publications, 2012.
- FOX, J.; WEISBERG, S. **An R Companion to Applied Regression**. Los Angeles and London: SAGE Publications, 2011.
- FRANKENBERG-GARCIA, A. Are translations longer than source texts?: A corpus-based study of explicitation. In: THIRD INTERNATIONAL CULT (CORPUS USE AND LEARNING TO TRANSLATE) CONFERENCE, 2004, Barcelona. [S.l.], 2004. p. 22-24.
- FRASER, B. Some remarks on the vpc in english. In: PROBLEMS IN SEMANTICS, HISTORY OF LINGUISTICS, LINGUISTICS AND ENGLISH. Washington, DC: Georgetown University Press, 1966. p. 45-61.
- GAUDIO, R. P. Sounding gay: Pitch properties in the speech of gay and straight men. **American Speech**, v. 69, n. 1, p. 30-57, 1994.
- GELMAN, A.; HILL, J. **Data Analysis Using Regression and Multilevel/Hierarchical Models**. Cambridge: Cambridge University Press, 2007.
- GENTLEMAN, R. **R Programming for Bioinformatics**. Boca Raton, FL: Chapman and Hall/CRC, 2009.
- GIRDEN, E. **ANOVA: repeated measures**. Los Angeles and London: Sage Publications, 1992. (Quantitative applications in the social sciences, N° 84).
- GOOD, P. **Introduction to Statistics Through Resampling Methods and R/S-PLUS**. Hoboken, NJ: Wiley, 2005.
- GOOD, P.; HARDIN, J. **Common Errors in Statistics (and How to Avoid Them)**. 4. ed. Hoboken, NJ: Wiley, 2012. (EBL-Schweitzer).
- GORARD, S. Revisiting a 90-year-old debate: the advantages of the mean deviation. In: THE BRITISH EDUCATIONAL RESEARCH ASSOCIATION ANNUAL CONFERENCE, University of Manchester. 2004. Disponível em: <<http://www.leeds.ac.uk/educol/documents/00003759.htm>>.
- GREENBAUM, S. Syntactic frequency and acceptability. **Lingua**, v. 40, n. 2, p. 99 – 113, 1976.
- GRIES, S. T. **Multifactorial Analysis in Corpus Linguistics: A Study of Particle Placement**. London, New York: Bloomsbury Publishing, 2003a. (Open Linguistics).
- GRIES, S. T. Towards a corpus-based identification of prototypical instances of constructions. In: ANNUAL REVIEW OF COGNITIVE LINGUISTICS. [S.l.], 2003b. p. 181-200.

- GRIES, S. T. Cognitive determinants of subtractive word formation: A corpus-based perspective. v. 17, p. 535–558, 2006.
- GRIES, S. T. **Quantitative Corpus Linguistics with R: A Practical Introduction**. London, New York: Taylor & Francis, 2009a.
- GRIES, S. T. **Statistics for Linguistics with R: A Practical Introduction**. Berlin, New York: De Gruyter, 2009b. (Mouton de Gruyter).
- GRIES, S. T. Frequency tables, effect sizes, and explorations. In: CORPUS METHODS FOR SEMANTICS: QUANTITATIVE STUDIES IN POLYSEMY AND SYNONYMY. Amsterdam & Philadelphia: John Benjamins, 2014. p. 365–389.
- GRIES, S. T.; WULFE, S. Do foreign language learners also have constructions? v. 3, p. 182–200, 2005.
- HARRELL, F. **Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis**. New York: Springer, 2001.
- HAWKINS, J. A. **A Performance Theory of Order and Constituency**. Cambridge: Cambridge University Press, 1994.
- HILBE, J. M. **Logistic Regression Models**. Boca Raton, FL: Chapman and Hall/CRC, 2009.
- IVERSEN, G. **Bayesian Statistical Inference**. Beverly Hills, CA: SAGE Publications, 1984.
- JACCARD, J.; JACCARD, J.; LEWIS-BECK, M. **Interaction Effects in Logistic Regression**. Thousand Oaks, CA: SAGE Publications, 2001. (A Sage university paper).
- JOHNSON, K. **Quantitative methods in linguistics**. Malden, MA: Blackwell Pub., 2008.
- JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2. ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2008.
- KEEN, K. **Graphics for Statistics and Data Analysis with R**. Boca Raton, FL: Taylor & Francis, 2010.
- KILLEEN, P. R. An alternative to null-hypothesis significance tests. **Psychological science**, v. 16, n. 5, p. 345–353, 2005.
- KORNAL, A. Analytic models in phonology. In: DURAND, J.; LAKS, B. (Ed.). **The Organization of Phonology: Constraints, Levels and Representations**. Oxford: Oxford University Press, 1998. p. 395–418.
- KUČERA, H.; FRANCIS, W. **Computational analysis of present-day American English**. RI: Brown University Press, 1967.
- LABOV, W. **What is a Linguistic Fact?** Indiana: John Benjamins Publishing Company, 1975. (Linguistic theory).
- LARSON-HALL, J. **A Guide to Doing Statistics in Second Language Research Using SPSS**. London and New York: Routledge, 2010.
- LIGGES, U. **Programmieren mit R**. Berlin, Heidelberg, New York: Springer, 2005.
- LOFTUS, G. R. Psychology will be a much better science when we change the way we analyze data. **Current Directions in Psychological Science**, v. 5, n. 6, p. 161–171, 1996.
- MANNING, C.; SCHÜTZE, H.; SCHUTZE, H. **Foundations of Statistical Natural Language Processing**. Cambridge, MA: MIT Press, 2000.
- MILLER, G. A. Empirical methods in the study of semantics. In: STEINBERG, D. D.; JAKOBOVITS, L. A. (Ed.). **Semantics: An Interdisciplinary Reader**. London, New York: Cambridge University Press, 1971. p. 569–585.
- MURRELL, P. **R Graphics, Second Edition**. 2. ed. Boca Raton, FL: Taylor & Francis, 2011.

- NAGATA, H. Long-term effect of repetition on judgments of grammaticality. **Perceptual and Motor Skills**, v. 65, n. 1, p. 295–299, 1987.
- NAGATA, H. Effect of repetition on grammaticality judgments under objective and subjective self-awareness conditions. **Journal of Psycholinguistic Research**, v. 18, n. 3, p. 255–269, 1989.
- OAKES, M. **Statistics for Corpus Linguistics**. Edinburgh: Edinburgh University Press, 1998.
- PAMPEL, F.; PUBLICATIONS, i. S. **Logistic Regression: A Primer**. Los Angeles and London: SAGE Publications, 2000.
- PETERS, J. Given vs. new information influencing constituent ordering in the vpc. In: LACUS FORUM XXVII: SPEAKING AND COMPREHENDING. Fullerton, CA: ACM Press, 2001. p. 133–140.
- RICE, S. Prepositional prototypes. In: PÜTZ, M.; DIRVEN, R. (Ed.). **The Construal of Space in Language and Thought**. Berlin, New York: Mouton de Gruyter, 1996. p. 35–65.
- RIZZO, M. **Statistical Computing with R**. Boca Raton, FL: Chapman and Hall/CRC, 2007.
- SANDRA, D.; RICE, S. Network analyses of prepositional meaning: Mirroring whose mind—the linguist's or the language user's? v. 6, n. 1, p. 89–130, 1995.
- SARKAR, D. **Lattice: Multivariate Data Visualization with R**. New York: Springer, 2008.
- SHEKIN, D. J. **Handbook of Parametric and Nonparametric Statistical Procedures**. Boca Raton, FL: Chapman and Hall/CRC, 2011.
- SHIRAI, Y.; ANDERSEN, R. W. The acquisition of tense-aspect morphology: A prototype account. **Language**, v. 71, n. 4, p. 743–762, 1995.
- SPECTOR, P. **Data Manipulation with R**. New York: Springer, 2008.
- SPENCER, N. J. Differences between linguists and nonlinguists in intuitions of grammaticality-acceptability. **Journal of Psycholinguistic Research**, v. 2, n. 2, p. 83–98, 1973.
- STEINBERG, D. Book. **An Introduction to Psycholinguistics**. London: Longman, 1993.
- STOLL, S.; GRIES, S. T. How to measure development in corpora? an association strength approach. **Journal of Child Language**, v. 36, n. 5, p. 1075–1090, 2009.
- TORGO, L. **Data Mining with R: Learning with Case Studies**. Boca Raton, FL: Chapman and Hall/CRC, 2010.
- TWISK, J. **Applied Multilevel Analysis: A Practical Guide for Medical Researchers**. Cambridge: Cambridge University Press, 2006.
- UNWIN, A.; THEUS, M.; HOFMANN, H. **Graphics of Large Datasets: Visualizing a Million**. New York: Springer, 2007.
- Van DONGEN, W. A. He put on his hat and he put his hat on. **Neophilologus**, v. 4, n. 1, p. 322–353, 1919.
- ZAR, J. **Biostatistical Analysis**. Upper Saddle River, NJ: Prentice Hall, 1999.
- ZUUR, A.; IENO, E.; SMITH, G. **Analyzing Ecological Data**. New York: Springer, 2007.
- ZUUR, A. et al. **Mixed Effects Models and Extensions in Ecology with R**. New York: Springer, 2009.