R语言在CGED-Q JSL中的运用

陈俊
香港科技大学社会科学部
2025年5月(V2.1.1)

Five

直方图、散点图和折线图的制作

01

基础直方图

5.1 绘制基础直方图
5.1.1 `ggplot2`包

- 制作直方图推荐的程序包是`ggplot2`包，这是R语言中制图功能比较强大的包，使用者可以在区域四的`Packages`里面找到`ggplot2`，也可以使用代码运行`ggpolt2`包：

```
install.packages("ggplot2")
library(ggplot2)
help("ggplot2")
```

5.1.2 `ggplot2`函数制图步骤

- 制图包安装完成后，就可进行制图

- `ggpolt2`包制图分为两个步骤
  - 第一步是输入`ggpolt()`函数针对数据集建立图层
  - 第二步使用`geom_bar()`函数调整直方图的样式和内容

5.1.3 `ggplot2`函数

`g <- ggplot(data,mapping = aes(x, y, fill, color,…),…)`
`g + geom_bar(mapping,data,stat,position,just,width,,…)`

- 首先是`ggpolt()`函数，其中的
  - `data`为数据集
  - `mapping = aes()`为建立图层
  - `aes()`函数中必须包含必要的X轴变量，和Y轴变量
  - `fill`为直方图图形的填充变量
  - `color`为直方图颜色的填充变量

5.1.3 `ggplot2`函数

```
`g <- ggplot(data,mapping = aes(x, y, fill, color,…),…)`
`g + geom_bar(mapping,data,stat,position,just,width,,…)`
```

- 其次是`geom_bar()`函数，其中`mapping`（图层）、`data`（数据集）都已经在前一条代码中确定，所以此处可以直接令其等于默认值`NULL`

- `stat`参数有三种类型
  - 其一是`count`,表示计算频数
  - 其二是`identity`,表示按照y的值进行计数
  - 其三是`bin`,表示对连续变量进行统计转换

- `Position`参数也有三种类型
  - 其一是`stack`,堆积
  - 其二是`fill`, 且每个条形的最大值为1
  - 其三是`dodge`,分条展示

- `Just`表示条形距离中心的距离
- `Width`表示条形的宽度

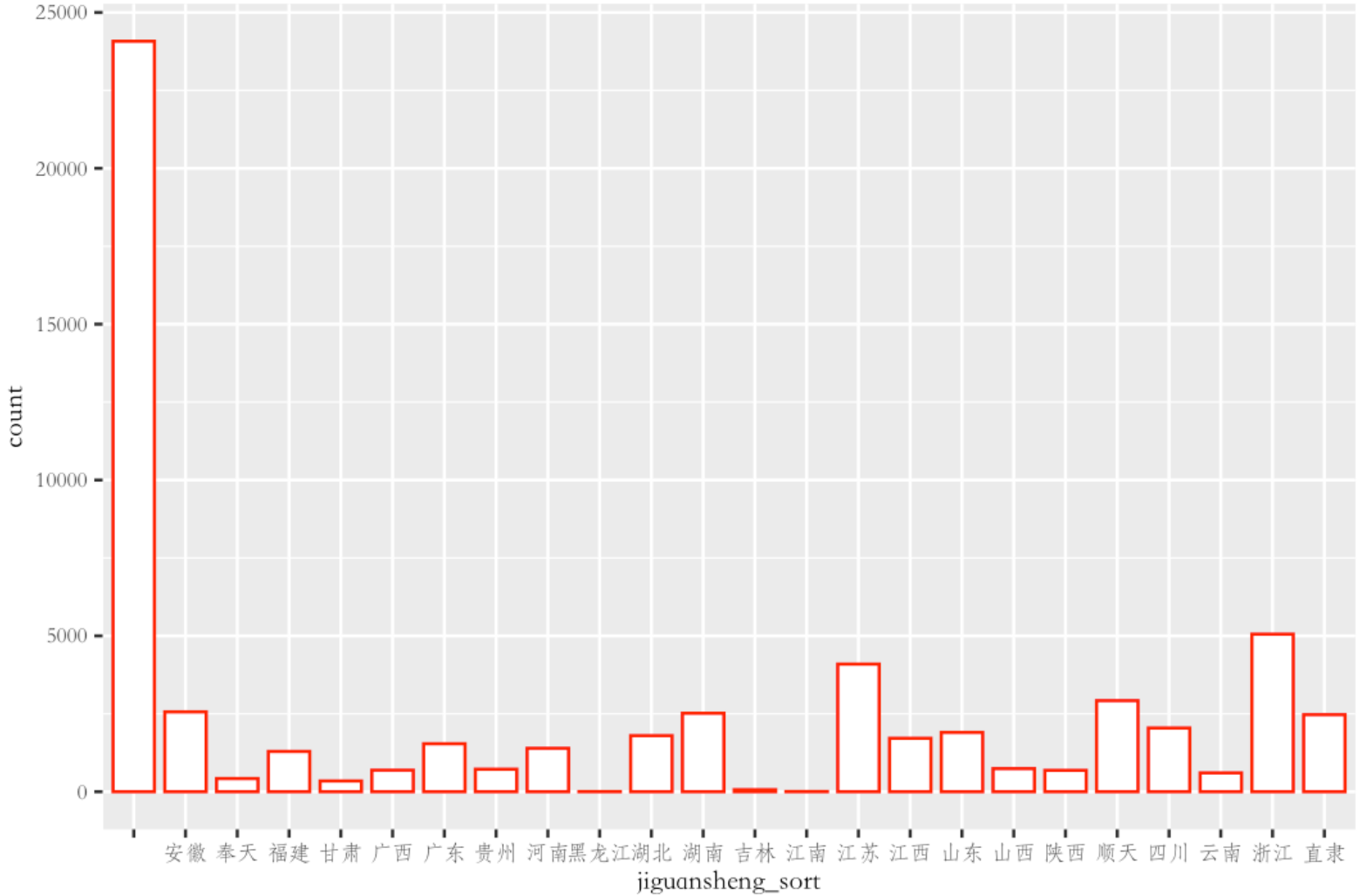
5.1.4 简单直方图的制作

- 设置这几个参数后，便可以尝试制作简单的直方图，示例代码如下：

```
P_BarChart <- ggplot(JSL1900_1912_only1910, aes(jiguansheng_sort)) +
  geom_bar(color="red", fill = "white", just= 0.5, width = 0.8) + theme(text =
element_text(family = "STKaiti", size = 9))

P_BarChart # 显示图片
```

- 这里用到的数据集`JSL1900_1912_only1910`是上一章指定条件制表时创建的数据集，现在我们用其来做图。图层里面选取的是“籍贯”这一变量作为横轴，纵轴默认使用"籍贯"的频数。`theme()`函数可以更改图形的样式。边框颜色使用红色，填充设置为白色，条状图形居中，条状宽度0.8cm


25000
20000
15000
10000
5000
0
count
安徽
奉天
福建
甘肃
广西
广东
贵州
河南
黑龙江
湖北
湖南
吉林
江南
江苏
江西
山东
山西
陕西
顺天
四川
云南
浙江
直隶
jiguansheng_sort

图像说明

- 可以看出，这只是一个初步的图形，存在许多问题：
  - 首先，横轴字符较为拥挤，看不清楚
  - 其次，横轴和纵轴的标题没有更改，且整个图形没有标题
  - 最后，条形最高的一项只有引号，即没有内容（缙绅录当中，在籍贯所在省任职的官员，其籍贯一项留空不录，这是缙绅录的编纂体例导致的），这一项所含信息较多，如果单纯地将其处理为缺失值，会影响分析结果

- 简单直方图存在许多问题，需要进一步调整制图函数的参数，优化图形效果

02

分条直方图

5.2 分条直方图的制作
5.2.1 数据准备

- 进阶直方图是在简单直方图的基础上控制更多函数的参数所制作出的更美观、更准确的图形

- 进阶直方图分为多种类型:
  - 分条直方图
  - 堆积直方图

- 这一部分主要讲分条直方图，在制作分条直方图前，我们需要先建立一个数据集再制图

```
# 指定条件提取只在兵部衙门任职的官员，以该条件为基础生成数据集
JSL1900_1912_bingbu <- subset(JSL1900_1912_clean,机构一 == "兵部衙門")
```

5.2.2 `theme()`函数

- 修饰函数`theme()`，使用方法就是在在原始代码的后面加入加上`+`号，将整个`theme()`函数添加到代码中。

- 这一函数能处理图例颜色、图例样式、网格间距、网格颜色等等修饰图形的细节

5.2.2 `theme()`函数

- `theme()`函数有非常多参数。值得注意的是，`theme()`函数中的参数必须与`element_text()`函数合用，`element()`系列函数有4个：
  - `element_text()`
  - `element_line()`
  - `element_rect()`
  - `element_blank()`

- 使用者必须根据参数的不同而调整`element()`函数的使用，也可利用help功能查看具体用法

- 使用者可根据自己的喜好设置图形的细节

- 在之后的图形制作讲解中，为了使代码看起来不冗长，我们不再标注`theme()`函数，如果有意学习`theme()`函数的使用者，可以在控制台键入`?theme`，或者登录`gglpot2`官网，学习`theme()`的语法

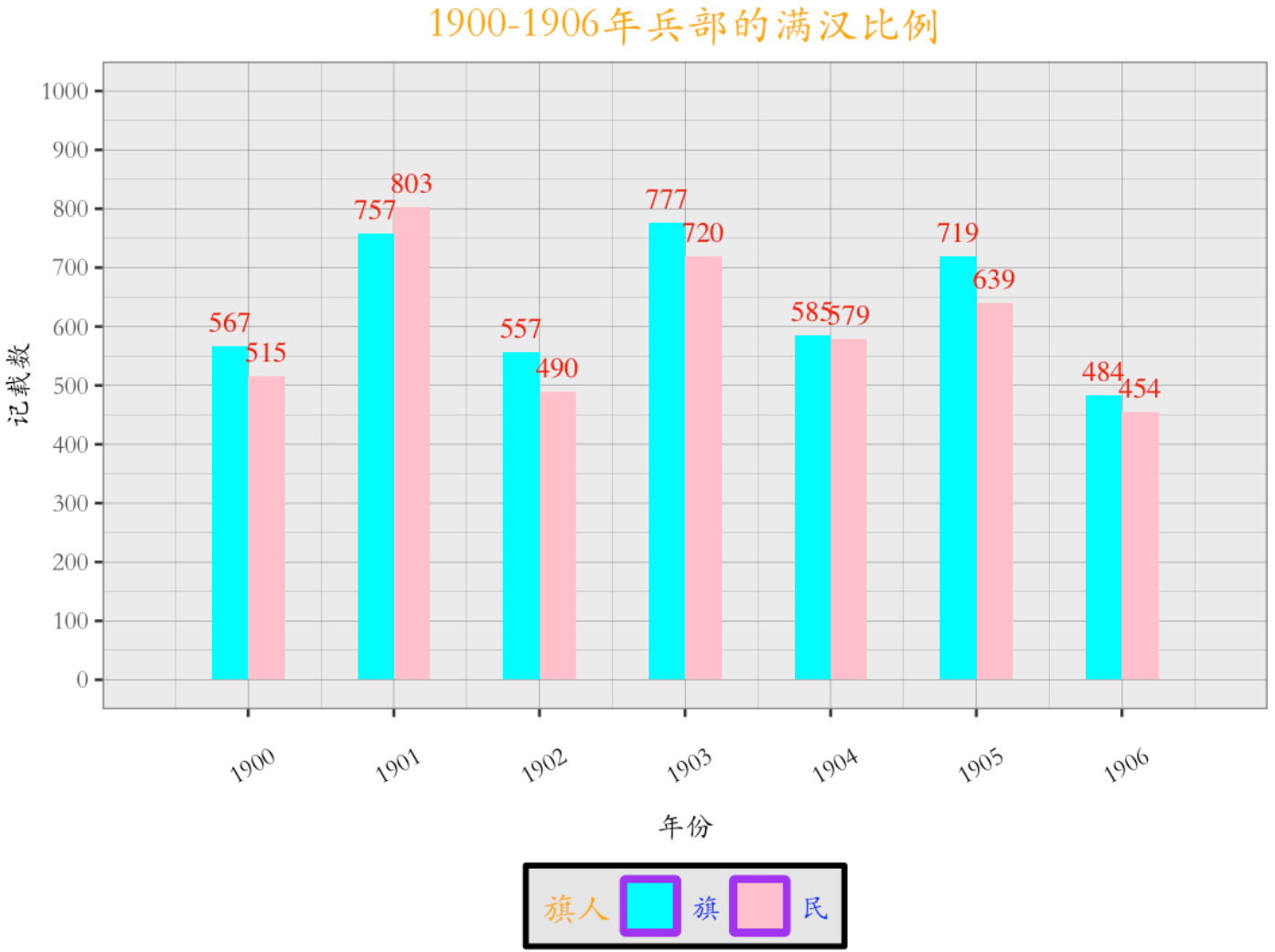
```
# 由于加入以下代码会导致代码整体偏长，如未特殊说明的情况下，往后的制图函数默认使用了该代码，但不在函数中显示。同时，不同图形的`theme()`的参数有时会有微调，请具体参考图形
theme(text = element_text(family = "STKaiti",size = 9), # 能处理图例颜色、图例样式、网格间距、网格颜色等等修饰图形的细节，这一参数可以设置的细节众多
   plot.title = element_text(size = 15,hjust = 0.5,color = "orange"),
      axis.text.x = element_text(size = 9,angle = 30,color = "black", margin = margin(t = .5, unit = "cm")),
      axis.text.y = element_text(size = 9),
      axis.title.x = element_text(size = 10,hjust = 0.5,vjust = 0),
      axis.title.y = element_text(size = 10),
      legend.position = "bottom",
      legend.title = element_text(size = 12,hjust = 0.5,color = "orange" ),
      legend.background = element_rect(color = "black",fill = "grey90", linewidth = 1),
      legend.text = element_text(size = 10,color = "blue"),
      legend.key = element_rect(linewidth = 2,color = "purple"),
      panel.background = element_rect(color = "grey50"),
      panel.grid = element_line(color="grey50",linewidth = 0.1)) # Mac系统制图后如无法显示中文字体，请在此添加这条代码"text = element_text(family = "STKaiti",size = 9),"，windows系统请忽略后同
```

制图代码

```
# 已省略`theme()`函数

P_BarChart_Dodge <- ggplot(JSL1900_1912_bingbu,aes(阳历年份numeric, fill = qiren)) +
  geom_bar(position = "dodge",just= 0.5,width = 0.5)+
  labs(x = "年份", y = "记载数",title = "1900-1906年兵部的满汉比例") + # 可设置横轴、纵轴、整个图形的标题
  guides(fill=guide_legend(title = "旗人")) + # 可设置图例的标题。观察图5.2的图例，图例的标题已将按照代码的要求被设置成"旗人"
  scale_fill_manual(values = c("cyan","pink")) + # 可设置条形的颜色，图中，民人和旗人颜色分别为青色和粉红色
  scale_x_continuous(expand = c(0.1,0.1),breaks = seq(1900,1906,1)) + # 可调整X轴与y轴的最大最小值、间隔等等
  scale_y_continuous(limits = c(0,1000),breaks = seq(0,1000,100)) +
  geom_text(aes(label = ..count..,family = "serif"), stat = "count",size = 3.5, # 可设置直方图数值标签的统计方式、大小、间隔、位置等
            vjust = -0.8,
            hjust = 0.5,
            color = "red2",
            position = position_dodge(0.5))

P_BarChart_Dodge # 显示图片
```


1900-1906年兵部的满汉比例
记载数
1000
900
800
700
600
500
400
300
200
100
0
567
515
757
803
557
490
777
720
585
579
719
639
484
454
1900
1901
1902
1903
1904
1905
1906
年份
旗人
旗
民

03

堆积直方图

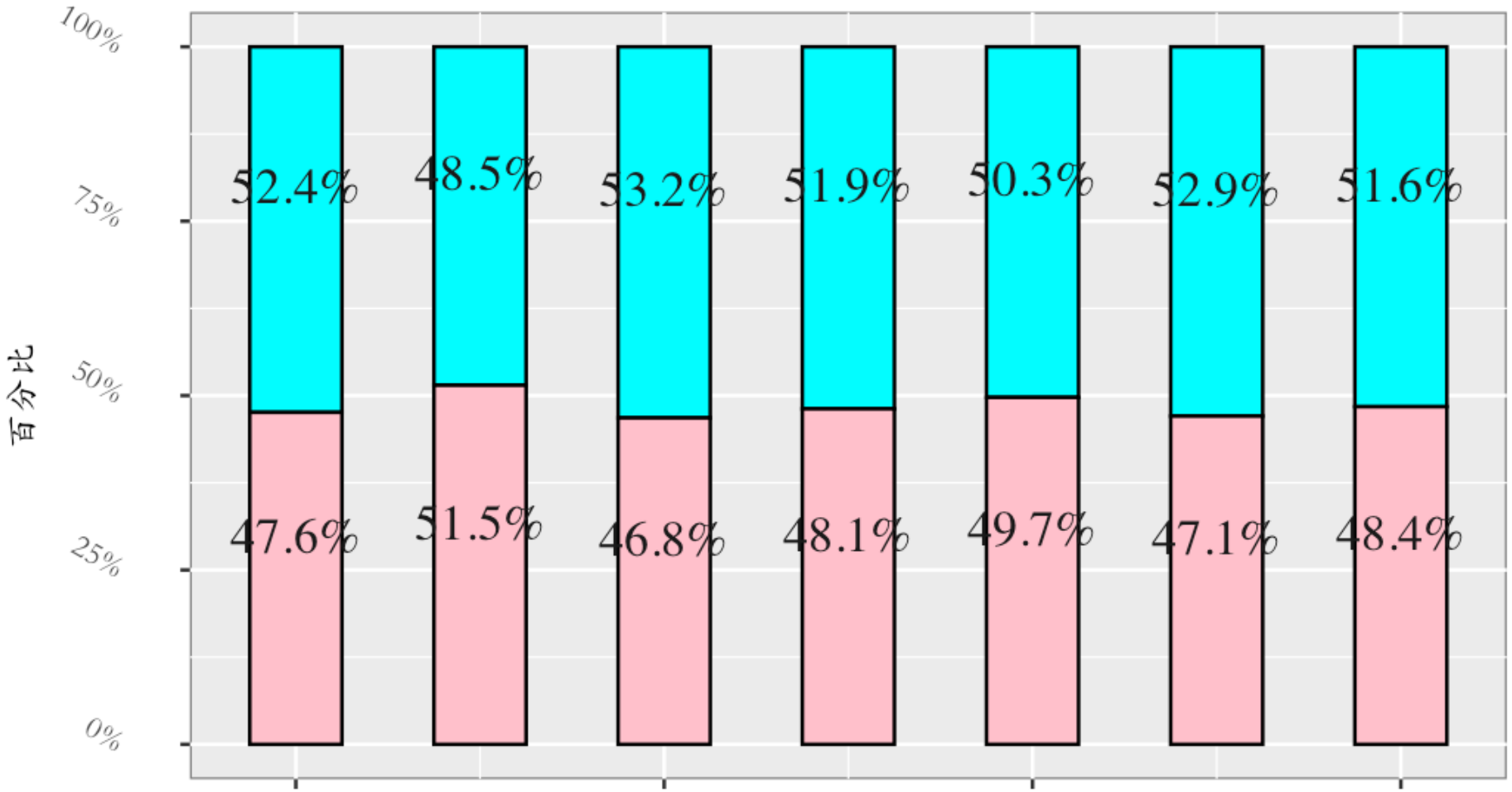
5.3 堆积直方图的制作

- 堆积直方图相较于分条直方图更适合研究比例问题
- 堆积直方图和分条直方图的代码大体相似，但在一些参数的设置上必须要细致。示例如下：

```
# 已省略`theme()`函数
P_BarChart_Fill <- ggplot(JSL1900_1912_bingbu,aes(阳历年份, fill = qiren)) +
  geom_bar(position = "fill",color="black",just= 0.5,width = 0.5)+
  labs(x = "年份", y = "百分比",title = "1900-1906年兵部的满汉比例")+
  guides(fill=guide_legend(title = "旗人"))+
  scale_fill_manual(values = c("cyan","pink"))+
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))+
  geom_text(aes(label=paste0(sprintf("%1.1f",
                        ..count.. / tapply(..count.., ..x.., sum)
                        [as.character(..x..)]*100),
                        “%”), # 引用自： https://stackoom.com/question/43QCB
            family = "serif"),
        stat = "count",
        size = 5,
        vjust = -0.8,
        hjust = 0.5,
        color = "gray10",
        position = position_fill(0.5))

P_BarChart_Fill # 显示图片
```

1900-1906年兵部的满汉比例



图像说明

- 对比分条直方图，堆积直方图主要有两个地方不同：
  - 第一个是`position`变量变成了`fill`
  - 第二个是`geom_text()`的`label`出现了嵌套函数，如果直接套用分条堆积图的代码，会导致每个条形比例之后不是100%，所以在这里我们运用了`paste0()`的嵌套函数

- 制作堆积直方图的代码较分条直方图较为复杂，使用者可根据需要直接套用代码。

- 以上为进阶直方图制作的具体流程，即在简单直方图函数上加上主题修饰函数`theme()`以及值标签制作函数`geom_text()`

- 另外可根据制图需求选择加上调整横、纵坐标间隔的`scale()`函数等。

04

利用离散型变量制作直方图

5.4 利用离散型数据制作直方图

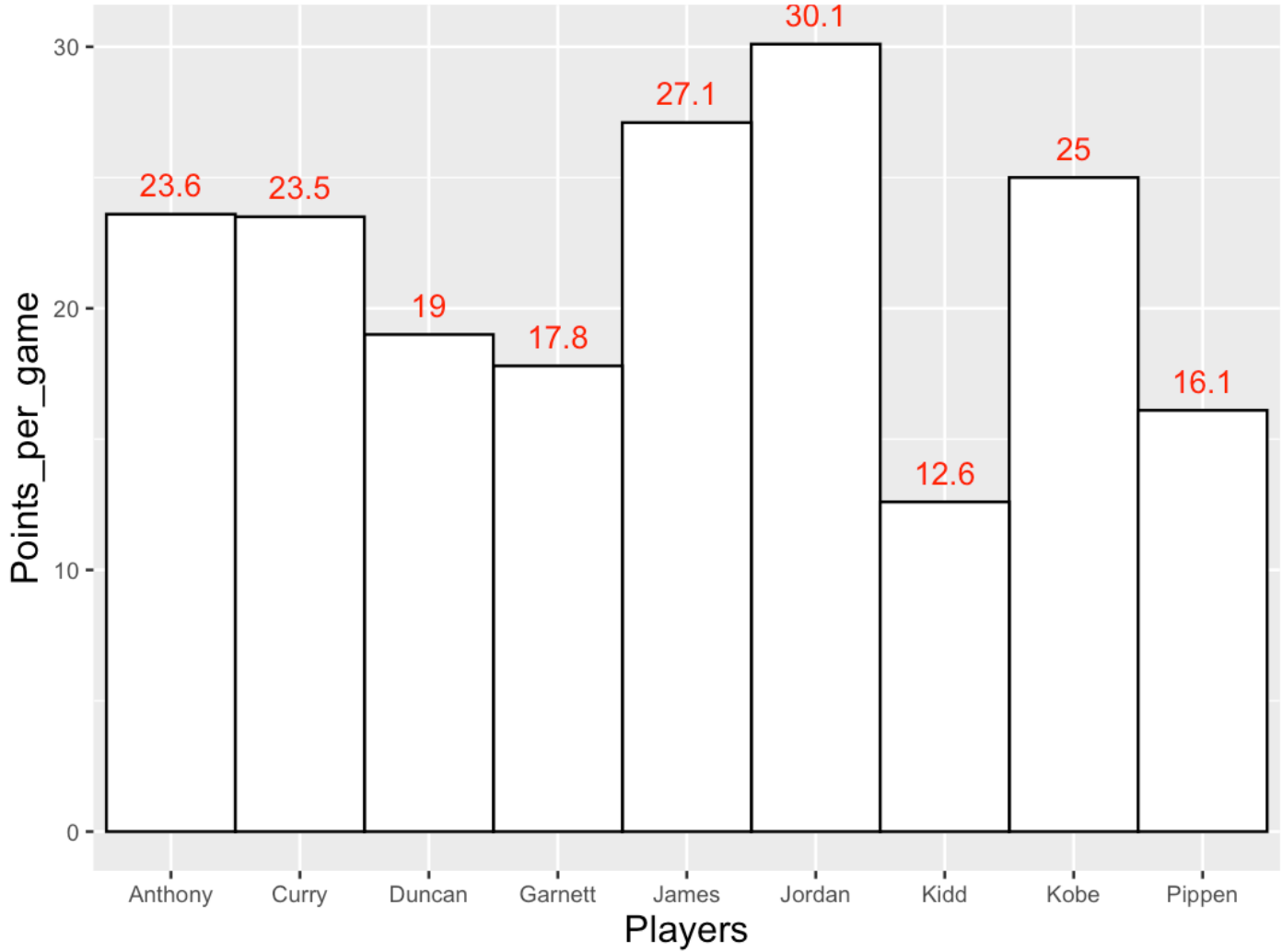
- 由于缙绅录数据库中缺少离散型变量，所以这里利用的数据集是第三章第三节提到的`NBA`数据集

- 其实离散型变量整体的代码与前面的相似，不过需要更改一些参数:
  - `geom_bar()`中需要添加一个y变量，
  - stat参数需要更改为`identity`，其默认为`count`，因为在这里我们添加了y变量，要用y的值来作为计数标准，所以选择`identity`
  - `geom_text()`中的`aes()`里的label参数要更改为与y轴一样的变量

- 代码示例如下:

```
P_NBA_BarChart <- ggplot(NBA,aes(Player,Points_per_game)) +
 geom_bar(color="black", fill="white", stat= "identity",just= 0.5,width = 1)+
 xlab("Players")+
 ylab("Points_per_game")+
 theme(axis.text.x = element_text(size = 8),
       axis.text.y = element_text(size = 8),
       axis.title.x = element_text(size = 13),
       axis.title.y = element_text(size = 13))+
 geom_text(aes(label=Points_per_game),
           vjust=-0.8,
           hjust=0.5,
           color="red")

P_NBA_BarChart # 显示图片
```


30
20
10
0
Points_per_game
23.6
23.5
19
17.8
27.1
30.1
12.6
25
16.1
Anthony
Curry
Duncan
Garnett
James
Jordan
Kidd
Kobe
Pippen
Players

注意事项

- 离散型数据制作直方图等图形相较于连续型数据较为方便，且离散型数据库的数量要远大于连续型数据库
- 因此，学习离散型数据库的制图方法也比较重要
- 使用者可根据自身需要结合代码和网络资源学习离散型数据制作图形的进阶方法

05

散点图

5.5 制作散点图
5.5.1 准备数据

- 散点图的制作流程和直方图类似，在建立图层的基础上套用`geom_point()`函数即可

- 在制作散点图之前，我们需要制作一个数据集来展现不同品级官员的满汉比例，以此来观察品级高低与官员满汉比例的关系。在进行制图之前，需要准备数据，创建一个数据集。这个数据集中包含
  - “尚书”（从一品）
  - “左侍郎”（二品）
  - “右侍郎”（二品）
  - “郎中”（五品）
  - “員外郎”（从五品）
  - “主事”（六品）
  - “七品京官”（七品）等7个官职的任职官员

5.5.1 准备数据

- 整体的思路是：
  - 首先创造一个空变量“官职品级”
  - 然后利用`str_detect()`判定函数，找出官职是“尚书”的记录
  - 再在“官职品级”变量中填入“尚书”（只在“尚书”官职判定为`TRUE`的行记录上），并利用相同思路，填入其他官职名称
  - 最后利用`factor`函数以“官职品级”为基础创造新变量“品级”并分层

```
JSL1900_1912_bingbu$官职品级 <- ""
尚书 <- str_detect(JSL1900_1912_bingbu$官职一,"尚書")
JSL1900_1912_bingbu$官职品级[尚书 == TRUE] <- "尚书"
左侍郎 <- str_detect(JSL1900_1912_bingbu$官职一,"左侍郎")
JSL1900_1912_bingbu$官职品级[左侍郎 == TRUE] <- "左侍郎"
右侍郎 <- str_detect(JSL1900_1912_bingbu$官职一,"右侍郎")
JSL1900_1912_bingbu$官职品级[右侍郎 == TRUE] <- "右侍郎"
主事 <- str_detect(JSL1900_1912_bingbu$官职一,"主事")
JSL1900_1912_bingbu$官职品级[主事 == TRUE] <- "主事"
員外郎 <- str_detect(JSL1900_1912_bingbu$官职一,"員外郎")
JSL1900_1912_bingbu$官职品级[員外郎 == TRUE] <- "員外郎"
郎中 <- str_detect(JSL1900_1912_bingbu$官职一,"郎中")
JSL1900_1912_bingbu$官职品级[郎中 == TRUE] <- "郎中"
七品京官 <- str_detect(JSL1900_1912_bingbu$官职一,"七品")
JSL1900_1912_bingbu$官职品级[七品京官 == TRUE] <- "七品京官"
JSL1900_1912_bingbu$品级 <- factor(JSL1900_1912_bingbu$官职品级,
                                   levels = c("尚书","左侍郎","右侍郎","郎中","員外郎","主事","七品京官"),
                                   labels = c(1.5,2,2,5,5.5,6,7))
```

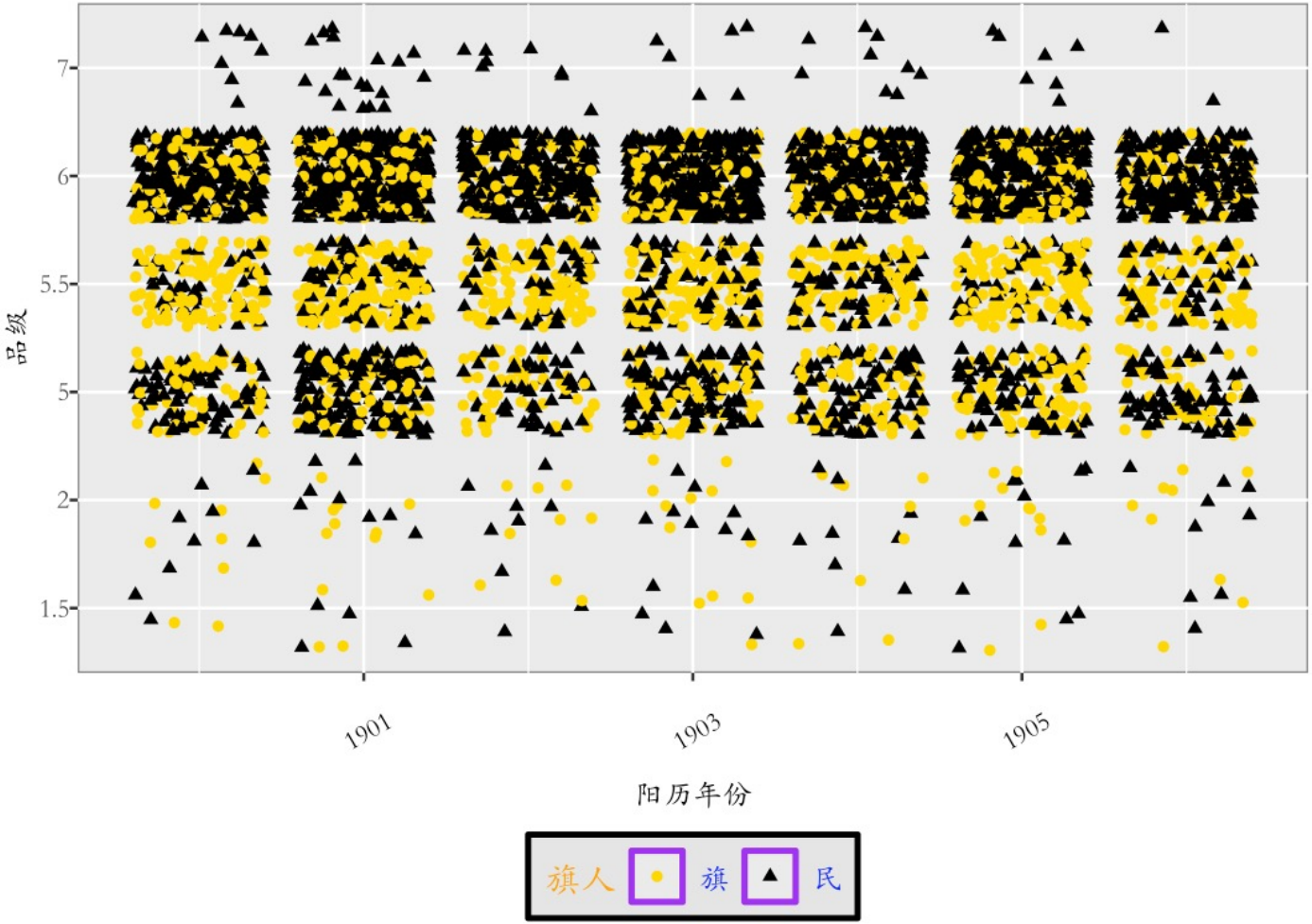
制图代码

```
# 已省略`theme()`函数

P_ScatterChart <- ggplot(data=subset(JSL1900_1912_bingbu, !is.na(品级)), # 利用`!is.na()`函数剔除缺失值
       aes(阳历年份,品级,shape = qiren,colour = qiren)) +
  geom_point(position = "jitter") + # 利用`jitter`添加抖动，防止点重合
  labs(title = "1900-1906年兵部官员的品级及数量")+
  guides(shape=guide_legend(title = "旗人"),colour=guide_legend(title = "旗人"))+
  scale_colour_manual(values = c("gold","black"))

P_ScatterChart # 显示图片
```

1900-1906年兵部官员的品级及数量



图像说明

- 从上图中可以看出：
  - 在中下层官员中，担任兵部员外郎的旗人数量比较多
  - 担任兵部郎中、兵部主事的民人数量较多
  - 没有旗人担任七品京官
  - 高层官员中，满汉比例几乎持平

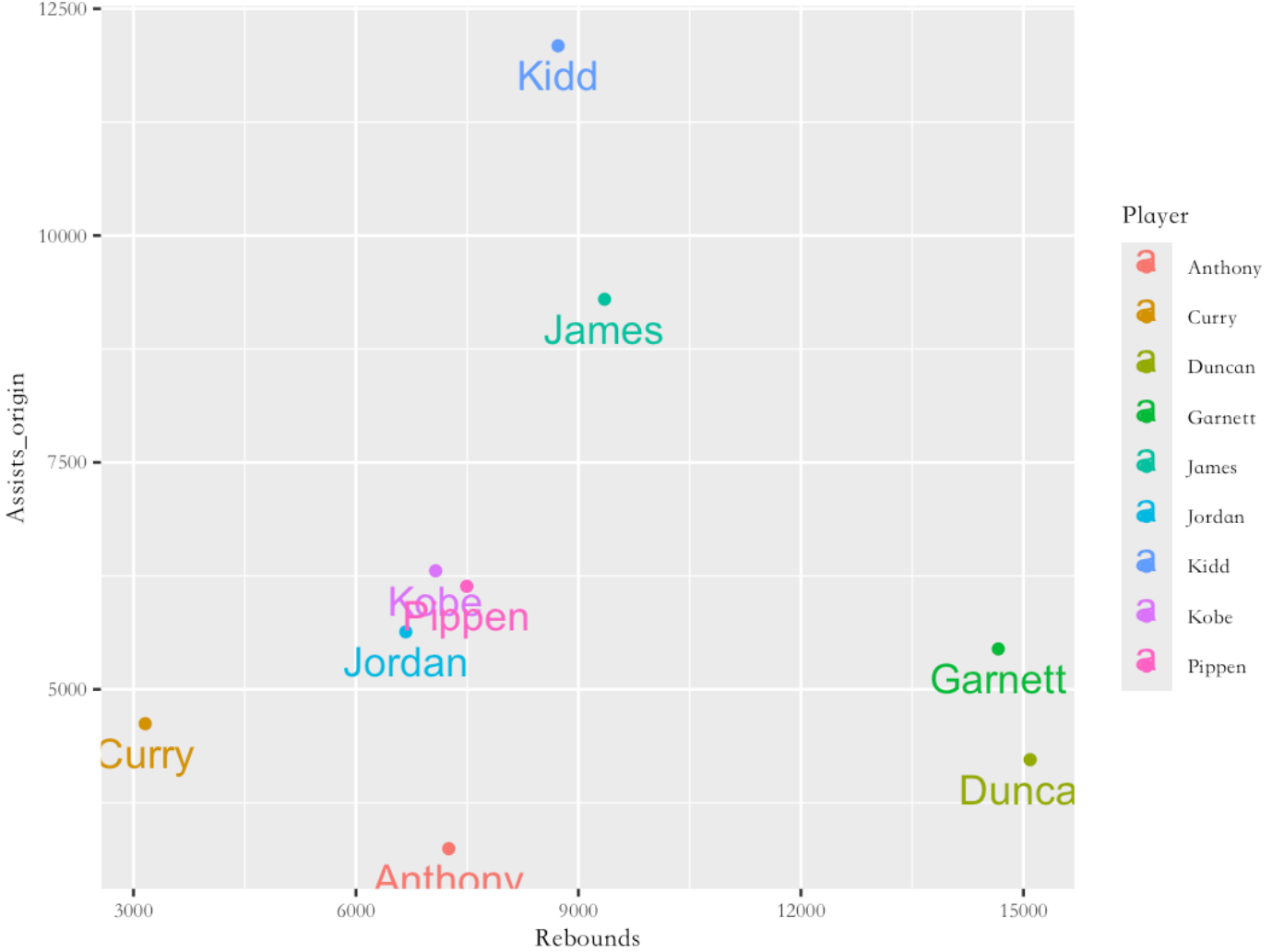
- 这是一张比较清晰的散点图，使用者可以套用代码研究其它部门官职品级和满汉比例的关系。

5.5.2 利用离散数据制作散点图

- 离散数据制作散点图较为简单，省去了准备数据的环节，只需要指定`x`、`y`值，我们还是用`NBA`数据集来举例子，示例代码如下：

```
P_NBA_ScatterChart <- ggplot( NBA, aes( Rebounds, Assists_origin, colour = Player )) +
  geom_point() +
  theme(text = element_text(family = "STKaiti",size = 9)) +
  geom_text( aes( label = Player ),size = 5, vjust = 1.5, hjust = 0.5)

P_NBA_ScatterChart # 显示图片
```


12500
10000
7500
5000
Assists_origin
Kidd
James
Kobe
Pippen
Jordan
Garnett
Curry
Duncan
Anthony
3000
6000
9000
12000
15000
Rebounds
Player
Anthony
Curry
Duncan
Garnett
James
Jordan
Kidd
Kobe
Pippen

图像说明

- 这个图形反映了NBA球员助攻篮板比的分布，越靠近右上区域的球员助攻篮板比的比例越大

- 使用者可根据以上代码在离散型数据集上尝试制作散点图

06

折线图

5.6 制作折线图
5.6.1 制图逻辑

- 折线图的制作逻辑是：
  - 先创造散点图，形成点状图层
  - 再制作折线图，将点状图层相连接，即形成折线图

- 折线图对数据的要求更高，x、y轴任意一轴需要是连续型变量，而另外一轴必须是离散型变量或连续型变量。JSL中缺少相应的离散变量，但可以根据数据计算出新的离散型变量，比如满汉比例。

5.6.2 准备数据

- 现在，我们尝试使用折线图来展现1900-1906年之间兵部衙门官员满汉比例的变化。
- 首先利用已知的1900-1906年兵部官员的满汉比例制作一个数据集，创造`year`、`percentage`、`shenfen`三列数据,并将这三条列数据整合为一个数据集。
- 请注意，`shenfen`与`percentage`一一是对应的，方便后面制图。

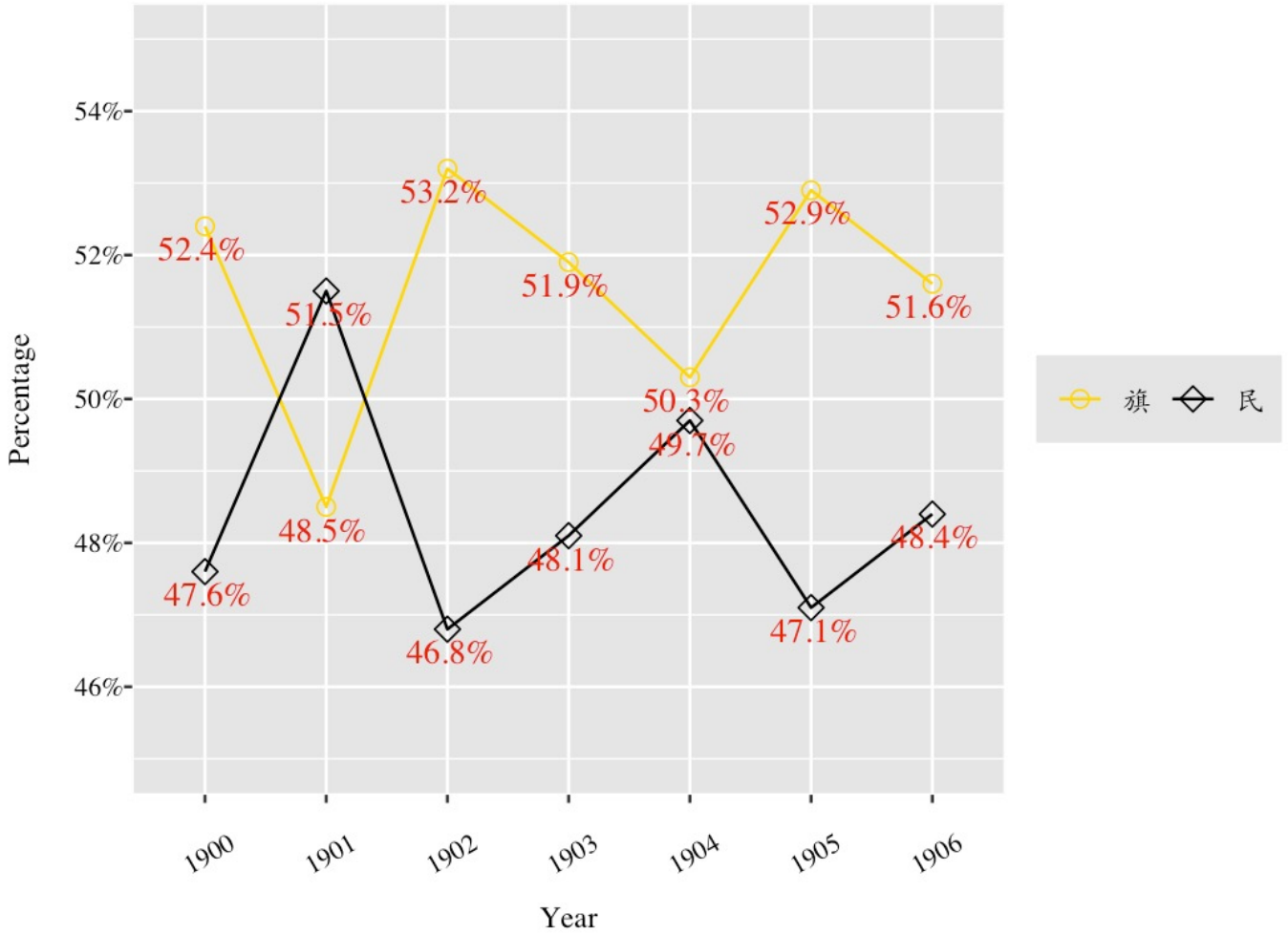
```
year <- c("1900","1901","1902","1903","1904","1905","1906")
percentage <- c(52.4,48.5,53.2,51.9,50.3,52.9,51.6,47.6,51.5,46.8,48.1,49.7,47.1,48.4)
shenfen <- c("旗","旗","旗","旗","旗","旗","旗","民","民","民","民","民","民","民")
bingbu1900_1906manhanbili <- data.frame(year,percentage,shenfen)
```

制图代码

```
# 数据集完成后，画出折线图
P_LineChart <- ggplot(bingbu1900_1906manhanbili,
        aes(x = year, y = percentage,group = shenfen,
            colour = shenfen, shape= shenfen)) +
# x、y轴有明确的指向，"group"表示按变量类型对散点进行分组，
# 如不分组，就会默认为全部点都在同一个组内，但这样就不能进行线段比较，
# 另外，按照"shape"区分颜色和点的类型，防止点重合
  geom_point(size=2.5,stroke=0.5) +
  geom_line() +
  scale_color_manual(values = c("gold","black")) +
  scale_shape_manual(values = c(1,5)) +
# `geom_line()`函数可以在散点图层制作完毕后，加入线段进行连接；
# `scale_shape_manual()`可更改点的样式
  labs(x = "Year",y= "Percentage",title = "1900-1906兵部衙门满汉比例折线图") +
scale_y_continuous(limits = c(45,55),breaks = c(46,48,50,52,54), labels =
c("46%","48%","50%","52%","54%"))+
  geom_text(aes(label=paste0(percentage,"%")),
# `geom_text()`函数中的"label=paste0(percentage,"%"" 用到了粘贴函数`paste0()`，
# 将百分比与%符号粘贴在一起
        family = “serif”,  stat = “identity”, size = 4, vjust = 1.5, hjust = 0.5, color = “red2”, position =
position_dodge(0.1))

P_LineChart # 显示图片
```

1900-1906兵部衙门满汉比例折线图



总结

- 以上即为三大基本图形的简单制作流程

- R中的ggpolt2包制作图形的方式如出一辙:
  - 首先建立图层
  - 再在图层上加上需要建立的图形函数`geom_bar()`、`geom_point()`、`geom_line()`
  - 最后对建立的图形进行修饰并加上值标签

- 如使用者有更多的制图需求，可以查看`ggplot2`包的帮助。

Thanks!