code.R

philipmoniz

2025-10-27

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(sandwich)
library(texreg)
```

```
## Version:  1.39.4
## Date:     2024-07-23
## Author:   Philip Leifeld (University of Manchester)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
##
## Attaching package: 'texreg'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
```

```
##
##     filter
```

```
library(rsample)
library(BayesFactor)
```

```
## Loading required package: coda
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## ************
## Welcome to BayesFactor 0.9.12-4.7. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```

```
setwd("~/Dropbox/Name Recognition Replication/")
#### Experiment 1: IL, PA, MD 2012 ####
dCD <- read_csv("2012_congress.csv")
```

```
## Rows: 8350 Columns: 51
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): pid2, pid3, state, state_cd
## dbl (47): Congress, gov_approval, Chal_4A, Chal_4B, Attention_4C, Attention_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dCD <- mutate(dCD, vote_inc = case_when(
      House_vote_13a == 1 ~ 1,
      House_vote_13b == 1 ~ 1,
      House_vote_13c == 1 ~ 1,
      TRUE ~ 0),

      vote_inc_no_labels = if_else(House_vote_13a == 1, 1, 0),

      vote_inc_party = if_else(House_vote_13b == 1, 1, 0),

      vote_inc_party_incumbency = if_else(House_vote_13c == 1, 1, 0),

      vote_inc_no_incumbency = case_when(
            !is.na(vote_inc_no_labels) ~ vote_inc_no_labels,
            !is.na(vote_inc_party) ~ vote_inc_party,
            TRUE ~ NA_real_
      ),

      vote_chal_no_labels = 1 - vote_inc_no_labels,

      vote_chal_party = 1 - vote_inc_party,
```

```
      vote_chal_party_incumbency = 1 - vote_inc_party_incumbency,

      vote_chal_no_incumbency = 1 - vote_inc_no_incumbency,

      treat = case_when(
            !is.na(Chal_4A) ~ 1,
            !is.na(Chal_4B) ~ 2,
            !is.na(Attention_4C) ~ 3,
            !is.na(Attention_4D) ~ 4,
            !is.na(Attention_4E) ~ 5,
            !is.na(Attention_4F) ~ 6,
            TRUE ~ 0),

      treat_party = case_when(
            treat == 2 | treat == 4 ~ 1,
            treat == 0 ~ 0,
            TRUE ~ NA_real_),

      treat_chal = case_when(
            treat == 1 | treat == 3 ~ 1,
            treat == 0 ~ 0,
            TRUE ~ NA_real_),

      treat_incumb = case_when(
            treat == 5 ~ 1,
            treat == 0 ~ 0,
            TRUE ~ NA_real_),

      treat_chal_pt = 1 - treat_chal,

      chal_primed = if_else(treat != "0", 1, 0),

      pid2 = case_match(
            Party_ID,
            1 ~ "Democratic",
            2 ~ "Republican",
            TRUE ~ NA_character_),

      ind = if_else(Party_ID > 2, 1, 0),

      pid3 = case_match(
            Party_ID,
            1 ~ "Democratic",
            2 ~ "Republican",
            3 ~ "Independent",
            TRUE ~ NA_character_),

      outpartisan = case_when(
            pid2 == "Democratic" & inc_dem == 1 ~ 0,
            pid2 == "Democratic" & inc_dem == 0 ~ 1,
            pid2 == "Republican" & inc_dem == 0 ~ 0,
            pid2 == "Republican" & inc_dem == 1 ~ 1),
```

```
      outpartisan_subset = case_when(
            pid2 == "Democratic" & inc_dem == 1 ~ 0,
            pid2 == "Democratic" & inc_dem == 0 ~ 1,
            pid2 == "Republican" & inc_dem == 0 ~ 0,
            pid2 == "Republican" & inc_dem == 1 ~ 1,
            TRUE ~ 0),

      bachelors = ifelse(Education >= 5, 1, 0),

      state = case_when(
            supplement == 2 & cd < 7 ~ "MD",
            supplement == 2 & cd > 7 ~ "PA",
            TRUE ~ "IL"),

      state_cd = paste0(state, cd))

# name-only datasets
dmere <- dCD |> filter(treat %in% c(0,1,3)) |>
      mutate(treat = case_when(
            treat == 0 ~ 0,
            treat == 1 ~ 1,
            TRUE ~ NA_real_))

dmereIL1 <- filter(dmere, state_cd == "IL1")
dmereIL3 <- filter(dmere, state_cd == "IL3")
dmereIL10 <- filter(dmere, state_cd == "IL10")
dmereMD3 <- filter(dmere, state_cd == "MD3")
dmereMD6 <- filter(dmere, state_cd == "MD6")
dmerePA10 <- filter(dmere, state_cd == "PA10")
dmerePA12 <- filter(dmere, state_cd == "PA12")

# name + party-only datasets
dnp <- dCD |> filter(treat %in% c(0,2,4)) |>
      mutate(treat = case_when(
            treat == 0 ~ 0,
            treat > 0 ~ 1,
            TRUE ~ NA_real_))

dnpIL1 <- filter(dnp, state_cd == "IL1")
dnpIL3 <- filter(dnp, state_cd == "IL3")
dnpIL10 <- filter(dnp, state_cd == "IL10")
dnpMD3 <- filter(dnp, state_cd == "MD3")
dnpMD6 <- filter(dnp, state_cd == "MD6")
dnpPA10 <- filter(dnp, state_cd == "PA10")
dnpPA12 <- filter(dnp, state_cd == "PA12")

# name + incumbency-only datasets
dinc <- dCD |> filter(treat %in% c(0,5)) |>
      mutate(treat = case_when(
            treat == 0 ~ 0,
            treat > 0 ~ 1,
            TRUE ~ NA_real_))
```

```r
dincIL1 <- filter(dnp, state_cd == "IL1")
dincIL3 <- filter(dnp, state_cd == "IL3")
dincIL10 <- filter(dnp, state_cd == "IL10")
dincMD3 <- filter(dnp, state_cd == "MD3")
dincMD6 <- filter(dnp, state_cd == "MD6")
dincPA10 <- filter(dnp, state_cd == "PA10")
dincPA12 <- filter(dnp, state_cd == "PA12")

# for merged data
# # name + party-only datasets
dCD4merge <- dCD |> filter(treat %in% c(0,2,4)) |>
      mutate(treat = case_when(
            treat == 0 ~ 0,
            treat > 0 ~ 1,
            TRUE ~ NA_real_)) |>
      select(treat, vote_chal = vote_chal_no_incumbency,
             state_cd, bachelors, outpartisan) |>
      mutate(study = state_cd) |>
      select(!state_cd)


# treat codes
# 1 = chal name
# 2 = chal name + party
# 3 = chal name + quality
# 4 = chal name + party + quality
# 5 = chal name + inc name
# 6 = chal name + inc name + party + quality

#### analysis ####

##### Name Only Treatment #####
######  Table 1 #######

## proportions hearing of challenger ##
# 3 = not heard of challenger
round(proportions(table(dmereIL1$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.113 0.187 0.700
```

```r
round(proportions(table(dmereIL3$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.075 0.169 0.756
```

```r
round(proportions(table(dmereIL10$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.192 0.163 0.644
```

```r
round(proportions(table(dmereMD3$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.034 0.138 0.828
```

```
round(proportions(table(dmereMD6$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.514 0.257 0.229
```

```
round(proportions(table(dmerePA10$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.034 0.103 0.862
```

```
round(proportions(table(dmerePA12$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.226 0.194 0.581
```

```
# % of all respondents
round(proportions(table(dmere$Chal_4A)), 3)
```

```
##
##     1     2     3
## 0.136 0.175 0.689
```

```
row_wise_prop_test(table(dmereIL1$vote_inc_no_labels, dmereIL1$treat_chal), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       438    71    78     0.364     0.321     0.714 0.398     1  -0.0510
## 2 1       438   124   165     0.636     0.679     0.714 0.398     1  -0.137
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(dmereIL3$treat_chal, dmereIL3$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       301    56    96     0.514       0.5    0.0120 0.913     1   -0.111
## 2 1       301    53    96     0.486       0.5    0.0120 0.913     1   -0.138
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(dmereIL10$treat_chal, dmereIL10$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       333    70    95     0.467     0.519     0.710   0.4     1  -0.166
## 2 1       333    80    88     0.533     0.481     0.710   0.4     1  -0.0614
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
row_wise_prop_test(table(dmereMD3$treat_chal, dmereMD3$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       139    72    31     0.742     0.738  1.33e-30     1     1   -0.159
## 2 1       139    25    11     0.258     0.262  2.27e-30     1     1   -0.167
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
row_wise_prop_test(table(dmereMD6$treat_chal, dmereMD6$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       133    76    35     0.835     0.833  2.11e-31     1     1   -0.136
## 2 1       133    15     7     0.165     0.167  2.11e-31     1     1   -0.140
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
row_wise_prop_test(table(dmerePA10$treat_chal, dmerePA10$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       135    60    51     0.779     0.879      1.63 0.201     1  -0.240
## 2 1       135    17     7     0.221     0.121      1.63 0.201     1  -0.0400
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
row_wise_prop_test(table(dmerePA12$treat_chal, dmerePA12$vote_inc_no_labels), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       140    65    44     0.793     0.759    0.0737 0.786     1   -0.121
## 2 1       140    17    14     0.207     0.241    0.0737 0.786     1   -0.190
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
round(proportions(table(dmere$vote_inc_no_labels)), 3)
```

```
##
##     0     1
## 0.466 0.534
```

```r
###### Table 2 & A1 ######
## pooled linear model ##
m1 <- lm(vote_chal_no_labels ~ treat_chal + state_cd, dCD)
texreg::screenreg(m1, digits = 3)
```

```
##
## ==========================
##               Model 1
## --------------------------
## (Intercept)      0.339 ***
##                 (0.027)
```

```
## treat_chal       0.002
##                 (0.026)
## state_cdIL10     0.110 **
##                 (0.035)
## state_cdIL3      0.022
##                 (0.036)
## state_cdMD3      0.358 ***
##                 (0.048)
## state_cdMD6      0.345 ***
##                 (0.049)
## state_cdPA10     0.231 ***
##                 (0.048)
## state_cdPA12     0.246 ***
##                 (0.048)
## --------------------------
## R^2              0.068
## Adj. R^2         0.064
## Num. obs.     1619
## ==========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
### bayes factor ###

dbf <- dCD |> select(vote_chal_no_labels, treat_chal, state_cd) |>
      na.omit()
mfull <- lmBF(vote_chal_no_labels ~ treat_chal + state_cd, dbf)
```

```
## Warning: data coerced from tibble to data frame
```

```
mnull <- lmBF(vote_chal_no_labels ~ state_cd, dbf)
```

```
## Warning: data coerced from tibble to data frame
```

```
mfull / mnull
```

```
## Bayes factor analysis
## --------------
## [1] treat_chal + state_cd : 0.05744961 ±0.92%
##
## Against denominator:
##   vote_chal_no_labels ~ state_cd
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
##### appendix analysis #####

###### Table A3 ######
# linear model with DV that does not give party label
m <- lm(vote_chal_no_labels ~ treat_party + state_cd, dCD,
        family = binomial())
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'family' will be disregarded
```

```
texreg::screenreg(m, digits = 3)
```

```
##
## ==========================
```

```
##               Model 1
## -------------------------
## (Intercept)      0.355 ***
##                 (0.027)
## treat_party      0.016
##                 (0.025)
## state_cdIL10     0.078 *
##                 (0.036)
## state_cdIL3      0.026
##                 (0.035)
## state_cdMD3      0.356 ***
##                 (0.047)
## state_cdMD6      0.331 ***
##                 (0.045)
## state_cdPA10     0.183 ***
##                 (0.047)
## state_cdPA12     0.221 ***
##                 (0.047)
## -------------------------
## R^2              0.063
## Adj. R^2         0.059
## Num. obs.     1694
## =========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### Table A4 ######
# linear model with only DV that gives party label
m <- lm(vote_chal_party ~ treat_party + state_cd, dCD,
        family = binomial())
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'family' will be disregarded
```

```
texreg::screenreg(m, digits = 3)
```

```
##
## =========================
##               Model 1
## -------------------------
## (Intercept)      0.341 ***
##                 (0.028)
## treat_party     -0.003
##                 (0.025)
## state_cdIL10     0.128 ***
##                 (0.036)
## state_cdIL3      0.062
##                 (0.036)
## state_cdMD3      0.359 ***
##                 (0.047)
## state_cdMD6      0.319 ***
##                 (0.049)
## state_cdPA10     0.315 ***
##                 (0.047)
## state_cdPA12     0.365 ***
##                 (0.049)
```

```
## --------------------------
## R^2              0.080
## Adj. R^2         0.076
## Num. obs.     1610
## ==========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### Table A8 ######
# pooled logit #
m1 <- glm(vote_chal_no_labels ~ treat_chal + state_cd, dCD,
          family = binomial())
texreg::screenreg(m1, digits = 3)
```

```
##
## ============================
##                 Model 1
## ----------------------------
## (Intercept)        -0.667 ***
##                    (0.118)
## treat_chal          0.008
##                    (0.110)
## state_cdIL10        0.464 **
##                    (0.149)
## state_cdIL3         0.097
##                    (0.157)
## state_cdMD3         1.502 ***
##                    (0.213)
## state_cdMD6         1.439 ***
##                    (0.216)
## state_cdPA10        0.949 ***
##                    (0.205)
## state_cdPA12        1.012 ***
##                    (0.202)
## ----------------------------
## AIC              2140.727
## BIC              2183.844
## Log Likelihood  -1062.364
## Deviance         2124.727
## Num. obs.        1619
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
# balance test
summary(glm(treat_chal ~ Gender + Age + factor(Race) + Education, dCD,
            family = binomial()))
```

```
##
## Call:
## glm(formula = treat_chal ~ Gender + Age + factor(Race) + Education,
##     family = binomial(), data = dCD)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.2115034  0.2115743  -1.000    0.317
## Gender        -0.0712215  0.0631100  -1.129    0.259
```

```
## Age           -0.0007537  0.0019009  -0.397    0.692
## factor(Race)2  0.3469513  0.1680849   2.064    0.039 *
## factor(Race)3 -0.0681281  0.1553440  -0.439    0.661
## factor(Race)4 -0.2083887  0.2075632  -1.004    0.315
## Education      0.0331259  0.0203142   1.631    0.103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5869.1  on 4271  degrees of freedom
## Residual deviance: 5839.4  on 4265  degrees of freedom
##   (4078 observations deleted due to missingness)
## AIC: 5853.4
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm(treat_party ~ Gender + Age + factor(Race) + Education, dCD,
            family = binomial()))
```

```
##
## Call:
## glm(formula = treat_party ~ Gender + Age + factor(Race) + Education,
##     family = binomial(), data = dCD)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.395068   0.214130  -1.845  0.06504 .
## Gender        -0.027007   0.061919  -0.436  0.66272
## Age           -0.000745   0.001892  -0.394  0.69370
## factor(Race)2  0.553634   0.171883   3.221  0.00128 **
## factor(Race)3  0.119599   0.160105   0.747  0.45506
## factor(Race)4  0.044094   0.207466   0.213  0.83169
## Education      0.034586   0.019905   1.737  0.08230 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6103.1  on 4419  degrees of freedom
## Residual deviance: 6069.2  on 4413  degrees of freedom
##   (3930 observations deleted due to missingness)
## AIC: 6083.2
##
## Number of Fisher Scoring iterations: 3
```

```
##### Name + Party Label Treatment #####

###### Table 4 ######
## proportions hearing of challenger
round(proportions(table(dnpIL1$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.133 0.301 0.566
```

```
round(proportions(table(dnpIL3$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.204 0.201 0.595
```

```
round(proportions(table(dnpIL10$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.306 0.238 0.456
```

```
round(proportions(table(dnpMD3$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.231 0.077 0.692
```

```
round(proportions(table(dnpMD6$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.538 0.231 0.231
```

```
round(proportions(table(dnpPA10$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.135 0.108 0.757
```

```
round(proportions(table(dnpPA12$Chal_4B)), 3)
```

```
##
##     1     2     3
## 0.344 0.125 0.531
```

```
## proportions tests ##
prop_test(table(dnpIL1$treat_party, dnpIL1$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   840   373   467     0.370     0.336     0.896 0.344     1  -0.0337     0.101
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```
prop_test(table(dnpIL3$treat_party, dnpIL3$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   668   297   371     0.387     0.402    0.0894 0.765     1  -0.0920    0.0632
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```
prop_test(table(dnpIL10$treat_party, dnpIL10$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   637   315   322     0.444     0.463     0.147 0.701     1  -0.0987    0.0622
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```r
prop_test(table(dnpMD3$treat_party, dnpMD3$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   290   212    78     0.689     0.756     0.957 0.328     1   -0.190    0.0549
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```r
prop_test(table(dnpMD6$treat_party, dnpMD6$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   297   209    88     0.656     0.727      1.15 0.284     1   -0.193    0.0495
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```r
prop_test(table(dnpPA10$treat_party, dnpPA10$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   294   224    70     0.589     0.629     0.199 0.656     1   -0.179     0.100
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```r
prop_test(table(dnpPA12$treat_party, dnpPA12$vote_inc_no_incumbency), detailed = T)
```

```
## # A tibble: 1 x 13
##       n    n1    n2 estimate1 estimate2 statistic     p    df conf.low conf.high
## * <dbl> <dbl> <dbl>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
## 1   274   209    65     0.656     0.585     0.794 0.373     1  -0.0752     0.217
## # i 3 more variables: method <chr>, alternative <chr>, p.signif <chr>
```

```r
#### Experiment 2: GOP Primary 2012 ####

d <- read_csv("2012_GOP.csv")
```

```
## Rows: 1600 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (13): vote_bachmann, bachmann_primed, vote_cain, cain_primed, vote_hunts...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
##### analysis #####

###### Table 3 ######
# group = 1 are rows voting for primed candidate
# n2 is group receiving prime treatment
row_wise_prop_test(table(d$vote_bachmann, d$bachmann_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0      1600   567   616     0.735     0.743    0.0850 0.771     1  -0.0520
## 2 1      1600   204   213     0.265     0.257    0.0850 0.771     1  -0.0367
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
```

```
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(d$vote_cain, d$cain_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0      1600   632   609     0.775     0.777   0.00242 0.961     1  -0.0444
## 2 1      1600   184   175     0.225     0.223   0.00242 0.961     1  -0.0399
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(d$vote_huntsman, d$huntsman_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0      1600   581   607     0.737     0.748     0.169 0.681     1  -0.0543
## 2 1      1600   207   205     0.263     0.252     0.169 0.681     1  -0.0339
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(d$vote_perry, d$perry_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0      1600   527   503     0.650     0.638     0.213 0.645     1  -0.0359
## 2 1      1600   284   286     0.350     0.362     0.213 0.645     1  -0.0605
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(d$vote_roemer, d$roemer_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df   conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>      <dbl>
## 1 0      1600   729   651     0.879     0.844      3.84  0.05     1 -0.0000942
## 2 1      1600   100   120     0.121     0.156      3.84  0.05     1 -0.0701
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
row_wise_prop_test(table(d$vote_santorum, d$santorum_primed), detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0      1600   622   656     0.792     0.805     0.318 0.573     1  -0.0531
## 2 1      1600   163   159     0.208     0.195     0.318 0.573     1  -0.0280
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
### bayes factors ###
contingencyTableBF(table(d$bachmann_primed, d$vote_bachmann), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.05844242 ±0%
```

```
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

```
contingencyTableBF(table(d$vote_cain, d$cain_primed), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.07536726 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

```
contingencyTableBF(table(d$vote_huntsman, d$huntsman_primed), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.07977138 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

```
contingencyTableBF(table(d$vote_perry, d$perry_primed), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.07452133 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

```
contingencyTableBF(table(d$vote_roemer, d$roemer_primed), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.7108992 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

```
contingencyTableBF(table(d$vote_santorum, d$santorum_primed), "indepMulti", fixedMargin = "rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 0.0948056 ±0%
##
## Against denominator:
##   Null, independence, a = 1
```

```
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
###### pooled meta-analytic estimate ######

mb <- lm(vote_bachmann ~ bachmann_primed, d)
mc <- lm(vote_cain ~ cain_primed, d)
mh <- lm(vote_huntsman ~ huntsman_primed, d)
mp <- lm(vote_perry ~ perry_primed, d)
mr <- lm(vote_roemer ~ roemer_primed, d)
ms <- lm(vote_santorum ~ santorum_primed, d)

# bootstrap 5000 samples
set.seed(02145)
samps <- rsample::bootstraps(d, 5000)

### estimate ATEs (betas) and regression standard errors (sigmas)
# bachmann betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_bachmann ~ bachmann_primed, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_bachmann ~ bachmann_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mb)
samps$bachmann_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                           mod_formula)
samps$bachmann_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                            mod_formula)

# cain betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_cain ~ cain_primed, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_cain ~ cain_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mc)
samps$cain_betas <- map(.x = samps$splits,
                        .f = lm_coefs,
                        mod_formula)
samps$cain_sigmas <- map(.x = samps$splits,
                         .f = lm_sigmas,
                         mod_formula)

# huntsman betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_huntsman ~ huntsman_primed, splits)
```

```
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_huntsman ~ huntsman_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mh)
samps$huntsman_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                           mod_formula)
samps$huntsman_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                            mod_formula)


# perry betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_perry ~ perry_primed, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_perry ~ perry_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mp)
samps$perry_betas <- map(.x = samps$splits,
                         .f = lm_coefs,
                        mod_formula)
samps$perry_sigmas <- map(.x = samps$splits,
                          .f = lm_sigmas,
                         mod_formula)


# roemer betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_roemer ~ roemer_primed, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_roemer ~ roemer_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mr)
samps$roemer_betas <- map(.x = samps$splits,
                          .f = lm_coefs,
                          mod_formula)
samps$roemer_sigmas <- map(.x = samps$splits,
                           .f = lm_sigmas,
                           mod_formula)

# santorum betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_santorum ~ santorum_primed, splits)
```

```
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_santorum ~ santorum_primed, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(ms)
samps$santorum_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                           mod_formula)
samps$santorum_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                             mod_formula)

### pooled meta-analytic regression estimate ###

(mb$coefficients[2] * (1/summary(mb)$sigma) +
            mc$coefficients[2] * (1/summary(mc)$sigma) +
            mh$coefficients[2] *  (1/summary(mh)$sigma) +
            mp$coefficients[2] * (1/summary(mp)$sigma) +
            mr$coefficients[2] * (1/summary(mr)$sigma) +
            ms$coefficients[2] * (1/summary(ms)$sigma)) / (
                  (1/summary(mb)$sigma) +
                        (1/summary(mc)$sigma) +
                        (1/summary(mh)$sigma) +
                        (1/summary(mp)$sigma) +
                        (1/summary(mr)$sigma) +
                        (1/summary(ms)$sigma)
            )
```

```
## bachmann_primed
##     0.003452427
```

```
### bootstrapped meta-analytic regression standard error ###

# making data frame of betas and sigmas
samp_df <- bind_cols(

      bachmann_betas = list_c(samps$bachmann_betas),
      cain_betas = list_c(samps$cain_betas),
      huntsman_betas = list_c(samps$huntsman_betas),
      perry_betas = list_c(samps$perry_betas),
      roemer_betas = list_c(samps$roemer_betas),
      santorum_betas = list_c(samps$santorum_betas),

      bachmann_sigmas = list_c(samps$bachmann_sigmas),
      cain_sigmas = list_c(samps$cain_sigmas),
      huntsman_sigmas = list_c(samps$huntsman_sigmas),
      perry_sigmas = list_c(samps$perry_sigmas),
      roemer_sigmas = list_c(samps$roemer_sigmas),
      santorum_sigmas = list_c(samps$santorum_sigmas),

)
```

```r
# new column is meta-analytic estimate

samp_df <- mutate(samp_df, tau =
                        (bachmann_betas * (1/bachmann_sigmas) +
                               cain_betas * (1/cain_sigmas) +
                               huntsman_betas * (1/huntsman_sigmas) +
                               perry_betas * (1/perry_sigmas) +
                               roemer_betas * (1/roemer_sigmas) +
                               santorum_betas * (1/santorum_sigmas)) / (
                                      (1/bachmann_sigmas) +
                                            (1/cain_sigmas) +
                                            (1/huntsman_sigmas) +
                                            (1/perry_sigmas) +
                                            (1/roemer_sigmas) +
                                            (1/santorum_sigmas)
                               ))

# bootstrapped standard error
sd(samp_df$tau)
```

```
## [1] 0.008721334
```

```r
# bootstrapped 95% CI
quantile(samp_df$tau, .025)
```

```
##        2.5%
## -0.01366305
```

```r
quantile(samp_df$tau, .975)
```

```
##      97.5%
## 0.02086583
```

```r
##### appendix analysis #####

###### controling for hispanic ######
mb <- lm(vote_bachmann ~ bachmann_primed + hispanic, d)
mc <- lm(vote_cain ~ cain_primed + hispanic, d)
mh <- lm(vote_huntsman ~ huntsman_primed + hispanic, d)
mp <- lm(vote_perry ~ perry_primed + hispanic, d)
mr <- lm(vote_roemer ~ roemer_primed + hispanic, d)
ms <- lm(vote_santorum ~ santorum_primed + hispanic, d)


# bootstrap 5000 samples
samps <- rsample::bootstraps(d, 5000)

### estimate ATEs (betas) and regression standard errors (sigmas)
# bachmann betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_bachmann ~ bachmann_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_bachmann ~ bachmann_primed + hispanic, splits)
      summary(mod)$sigma
```

```
}
mod_formula <- as.formula(mb)
samps$bachmann_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                            mod_formula)
samps$bachmann_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                             mod_formula)

# cain betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_cain ~ cain_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_cain ~ cain_primed + hispanic, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mc)
samps$cain_betas <- map(.x = samps$splits,
                        .f = lm_coefs,
                        mod_formula)
samps$cain_sigmas <- map(.x = samps$splits,
                         .f = lm_sigmas,
                         mod_formula)

# huntsman betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_huntsman ~ huntsman_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_huntsman ~ huntsman_primed + hispanic, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mh)
samps$huntsman_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                            mod_formula)
samps$huntsman_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                             mod_formula)


# perry betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_perry ~ perry_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_perry ~ perry_primed + hispanic, splits)
      summary(mod)$sigma
}
```

```
mod_formula <- as.formula(mp)
samps$perry_betas <- map(.x = samps$splits,
                         .f = lm_coefs,
                        mod_formula)
samps$perry_sigmas <- map(.x = samps$splits,
                          .f = lm_sigmas,
                          mod_formula)


# roemer betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_roemer ~ roemer_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_roemer ~ roemer_primed + hispanic, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(mr)
samps$roemer_betas <- map(.x = samps$splits,
                          .f = lm_coefs,
                          mod_formula)
samps$roemer_sigmas <- map(.x = samps$splits,
                           .f = lm_sigmas,
                           mod_formula)

# santorum betas and sigmas
lm_coefs <- function(splits, ...){
      mod <- lm(vote_santorum ~ santorum_primed + hispanic, splits)
      coef(mod)[2]
}
lm_sigmas <- function(splits, ...){
      mod <- lm(vote_santorum ~ santorum_primed + hispanic, splits)
      summary(mod)$sigma
}
mod_formula <- as.formula(ms)
samps$santorum_betas <- map(.x = samps$splits,
                            .f = lm_coefs,
                            mod_formula)
samps$santorum_sigmas <- map(.x = samps$splits,
                             .f = lm_sigmas,
                             mod_formula)

### pooled meta-analytic regression estimate ###

(mb$coefficients[2] * (1/summary(mb)$sigma) +
            mc$coefficients[2] * (1/summary(mc)$sigma) +
            mh$coefficients[2] *  (1/summary(mh)$sigma) +
            mp$coefficients[2] * (1/summary(mp)$sigma) +
            mr$coefficients[2] * (1/summary(mr)$sigma) +
            ms$coefficients[2] * (1/summary(ms)$sigma)) / (
                  (1/summary(mb)$sigma) +
                        (1/summary(mc)$sigma) +
```

```
                        (1/summary(mh)$sigma) +
                        (1/summary(mp)$sigma) +
                        (1/summary(mr)$sigma) +
                        (1/summary(ms)$sigma)
            )
```

```
## bachmann_primed
##     0.003204764
```

```
### bootstrapped meta-analytic regression standard error ###

# making data frame of betas and sigmas
samp_df <- bind_cols(

      bachmann_betas = list_c(samps$bachmann_betas),
      cain_betas = list_c(samps$cain_betas),
      huntsman_betas = list_c(samps$huntsman_betas),
      perry_betas = list_c(samps$perry_betas),
      roemer_betas = list_c(samps$roemer_betas),
      santorum_betas = list_c(samps$santorum_betas),

      bachmann_sigmas = list_c(samps$bachmann_sigmas),
      cain_sigmas = list_c(samps$cain_sigmas),
      huntsman_sigmas = list_c(samps$huntsman_sigmas),
      perry_sigmas = list_c(samps$perry_sigmas),
      roemer_sigmas = list_c(samps$roemer_sigmas),
      santorum_sigmas = list_c(samps$santorum_sigmas),

)

# new column is meta-analytic estimate

samp_df <- mutate(samp_df, tau =
                        (bachmann_betas * (1/bachmann_sigmas) +
                               cain_betas * (1/cain_sigmas) +
                               huntsman_betas * (1/huntsman_sigmas) +
                               perry_betas * (1/perry_sigmas) +
                               roemer_betas * (1/roemer_sigmas) +
                               santorum_betas * (1/santorum_sigmas)) / (
                                     (1/bachmann_sigmas) +
                                           (1/cain_sigmas) +
                                           (1/huntsman_sigmas) +
                                           (1/perry_sigmas) +
                                           (1/roemer_sigmas) +
                                           (1/santorum_sigmas)
                               ))

# bootstrapped standard error
sd(samp_df$tau)
```

```
## [1] 0.008711408
```

```
# bootstrapped 95% CI
quantile(samp_df$tau, .025)
```

```
##        2.5%
## -0.01397233
```

```
quantile(samp_df$tau, .975)
```

```
##     97.5%
## 0.0199059
```

```
#### Experiment 3: MA 2018 ####

#setwd("Dropbox/Name recognition/")

dMA <- read_csv("MA_SOS_2018.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 2006 Columns: 60
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (26): lalvoterid, voters_active, residence_families_familyid, mailing_fa...
## dbl (31): walking_list_sequence, residence_addresses_zip, mailing_addresses_...
## lgl  (3): votertelephones_telcellflag, residence_addresses_extraaddress, mai...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dMA <- mutate(dMA, vote_dem = case_match(
      q5,
      1 ~ 1,
      2 ~ 0,
      3 ~ NA_real_),

      vote_chal = 1 - vote_dem,

      treat2 = case_match(
            version,
            1 ~ "Dem",
            2 ~ "GOP",
            3 ~ "Dem + Party",
            4 ~ "GOP + Party",
            5 ~ "Control"),

      treat = case_match(version,
                         4 ~ 1,
                         5 ~ 0,
                         TRUE ~ NA_real_),

      pid3 = case_match(
            PID,
            1 ~ "Democratic",
            2 ~ "Republican",
            3 ~ "Independent",
            TRUE ~ NA_character_),
```

```
      bachelors = case_match(q10,
                             c(5,6) ~ 1,
                             .default = 0),

      outpartisan = case_match(
            PID,
            1 ~ 0,
            2 ~ 1,
            TRUE ~ NA_real_),

      pid7ish = case_when(
            PID == 1 & q3 == 2 ~ 1,
            PID == 1 & q3 == 1 ~ 2,
            PID == 3 & q3 == 2 ~ 3, # ind + disapprove of Trump
            PID == 3 & q3 == 1 ~ 4, # ind + approve of Trump
            PID == 2 & q3 == 2 ~ 5,
            PID == 2 & q3 == 1 ~ 6,
            TRUE ~ NA_real_)
)

#### analysis ####

##### Name + Party Treatment #####

###### Table 4 cont. ######
## proportions tests ##
row_wise_prop_test(table(dMA$vote_chal, dMA$treat),
                   detailed = T, p.adjust.method = "none")
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       716   209   204     0.587     0.567     0.228 0.633     1  -0.0547
## 2 1       716   147   156     0.413     0.433     0.228 0.633     1  -0.0956
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```
### for merged dataset
dMA4merge <- dMA |> filter(version %in% c(4, 5)) |>
      select(treat, vote_chal, outpartisan, bachelors) |>
      mutate(study = "MA 2018")


#### Experiment 4: KY 2011 ####
dKY <- read_csv("2011_KY.csv")
```

```
## Rows: 601 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): vote_ppool, ppool_favorability, vote_intent, PID, educ, treatment_f...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
dKY <- mutate(dKY, vote_ppool = case_match(
      vote_intent,
      1 ~ 0,
      2 ~ 0,
      3 ~ 1,
      4 ~ 1,
      5 ~ NA_real_), # 1 and 2 are voting for incumbent

      vote_inc = ifelse(vote_intent < 3, 1 ,0),

      vote_chal = 1 - vote_inc,

      pid7 = na_if(PID, 8),

      pid3 = case_when(
            pid7 < 4 ~ "Democratic",
            pid7 == 4 ~ "Independent",
            pid7 > 4 ~ "Republican",
            pid7 == 8 ~ NA_character_),

      bachelors = case_match(educ,
                             c(6,7) ~ 1,
                             8 ~ NA_real_,
                             .default = 0),

      treat = ifelse(treatment_favorability == 1, 1, 0),

      outpartisan = case_when(
            pid7 < 4 ~ 0,
            pid7 == 4 ~ NA_real_,
            pid7 > 4 ~ 1,
            pid7 == 8 ~ NA_real_))

dKY4merge <- dKY |> select(treat, vote_chal, bachelors, outpartisan) |>
      mutate(study = "KY 2011")

##### analysis #####

###### Table 4 cont. ######
## proportions tests ##
# 64% never heard of p'pool
prop.table(table(dKY$ppool_favorability))
```

```
##
##         1         2         3         4         5         6         7
## 0.02640264 0.06270627 0.05940594 0.05610561 0.12871287 0.64356436 0.02310231
```

```r
proportions(table(dKY$vote_chal, dKY$treat), 2)
```

```
##
##             0         1
##   0 0.5100671 0.5049505
##   1 0.4899329 0.4950495
```

```r
row_wise_prop_test(table(dKY$vote_chal, dKY$treat),
                   p.adjust.method = "none", detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic     p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl> <dbl> <dbl>    <dbl>
## 1 0       601   152   153     0.510     0.505   0.00192 0.965     1  -0.0782
## 2 1       601   146   150     0.490     0.495   0.00192 0.965     1  -0.0884
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
#### merge datasets ####

dbig <- bind_rows(dCD4merge, dMA4merge, dKY4merge)

##### pooled proportions test #####

row_wise_prop_test(table(dbig$vote_chal, dbig$treat),
                   p.adjust.method = "none", detailed = T)
```

```
## # A tibble: 2 x 15
##   group     n    n1    n2 estimate1 estimate2 statistic       p    df conf.low
## * <chr> <int> <int> <int>     <dbl>     <dbl>     <dbl>   <dbl> <dbl>    <dbl>
## 1 0      4617  1255  1158     0.503     0.545      7.86 0.00505     1  -0.0711
## 2 1      4617  1238   966     0.497     0.455      7.86 0.00505     1   0.0125
## # i 5 more variables: conf.high <dbl>, method <chr>, alternative <chr>,
## #   p.adj <dbl>, p.adj.signif <chr>
```

```r
###### Table 5 & A2 ######
m5 <- lm(vote_chal ~ treat + study, dbig)
texreg::screenreg(m5, digits = 3)
```

```
##
## ==========================
##                Model 1
## --------------------------
## (Intercept)      0.346 ***
##                 (0.019)
## treat            0.009
##                 (0.015)
## studyIL10        0.103 ***
##                 (0.026)
## studyIL3         0.044
##                 (0.025)
## studyKY 2011     0.142 ***
##                 (0.026)
## studyMA 2018     0.072 **
##                 (0.025)
## studyMD3         0.358 ***
##                 (0.033)
## studyMD6         0.328 ***
##                 (0.033)
## studyPA10        0.250 ***
##                 (0.033)
## studyPA12        0.290 ***
```

```
##                  (0.034)
## -------------------------
## R^2               0.051
## Adj. R^2          0.049
## Num. obs.      4617
## =========================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
### bayes factor ###

dbf <- dbig |> select(vote_chal, treat, study) |>
      na.omit()
mfull <- lmBF(vote_chal ~ treat + study, dbf)
```

```
## Warning: data coerced from tibble to data frame
```

```
mnull <- lmBF(vote_chal ~ study, dbf)
```

```
## Warning: data coerced from tibble to data frame
```

```
mfull / mnull
```

```
## Bayes factor analysis
## --------------
## [1] treat + study : 0.03818054 ±1.23%
##
## Against denominator:
##   vote_chal ~ study
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
###### Table A5 ######
# interaction with Out-Party ID
m5 <- glm(vote_chal ~ treat*outpartisan + study, dbig,
          family = binomial())
texreg::screenreg(m5, digits = 3)
```

```
##
## ================================
##                    Model 1
## --------------------------------
## (Intercept)           -1.876 ***
##                       (0.144)
## treat                  0.061
##                       (0.146)
## outpartisan            3.181 ***
##                       (0.142)
## studyIL10             -0.008
##                       (0.185)
## studyIL3              -0.412 *
##                       (0.185)
## studyKY 2011           0.376 *
##                       (0.166)
## studyMA 2018          -0.029
##                       (0.187)
## studyMD3               0.473 *
##                       (0.238)
```

```
## studyMD6               0.483 *
##                       (0.230)
## studyPA10             -0.342
##                       (0.216)
## studyPA12             -0.224
##                       (0.225)
## treat:outpartisan      0.257
##                       (0.200)
## --------------------------------
## AIC                 2758.288
## BIC                 2830.812
## Log Likelihood     -1367.144
## Deviance            2734.288
## Num. obs.           3114
## ================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### Table A6 ######
# interaction with education
m5 <- glm(vote_chal ~ treat*bachelors + study, dbig,
          family = binomial())
texreg::screenreg(m5, digits = 3)
```

```
##
## ==============================
##                   Model 1
## ------------------------------
## (Intercept)         -0.706 ***
##                     (0.089)
## treat                0.092
##                     (0.086)
## bachelors            0.156
##                     (0.084)
## studyIL10            0.407 ***
##                     (0.108)
## studyIL3             0.192
##                     (0.107)
## studyKY 2011         0.574 ***
##                     (0.110)
## studyMA 2018         0.299 **
##                     (0.105)
## studyMD3             1.489 ***
##                     (0.149)
## studyMD6             1.347 ***
##                     (0.145)
## studyPA10            1.034 ***
##                     (0.141)
## studyPA12            1.202 ***
##                     (0.147)
## treat:bachelors     -0.112
##                     (0.122)
## ------------------------------
## AIC                6151.232
## BIC                6228.445
## Log Likelihood    -3063.616
```

```
## Deviance          6127.232
## Num. obs.         4603
## ==============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### Table A7 ######
m5 <- glm(vote_chal ~ treat*study, dbig,
          family = binomial())
texreg::screenreg(m5, digits = 3)
```

```
##
## ================================
##                      Model 1
## --------------------------------
## (Intercept)            -0.532 ***
##                        (0.107)
## treat                  -0.148
##                        (0.145)
## studyIL10               0.309 *
##                        (0.156)
## studyIL3                0.073
##                        (0.160)
## studyKY 2011            0.492 **
##                        (0.158)
## studyMA 2018            0.180
##                        (0.152)
## studyMD3                1.326 ***
##                        (0.183)
## studyMD6                1.176 ***
##                        (0.181)
## studyPA10               0.893 ***
##                        (0.173)
## studyPA12               1.176 ***
##                        (0.181)
## treat:studyIL10         0.222
##                        (0.216)
## treat:studyIL3          0.208
##                        (0.216)
## treat:studyKY 2011      0.168
##                        (0.218)
## treat:studyMA 2018      0.232
##                        (0.210)
## treat:studyMD3          0.487
##                        (0.336)
## treat:studyMD6          0.486
##                        (0.316)
## treat:studyPA10         0.313
##                        (0.317)
## treat:studyPA12        -0.154
##                        (0.325)
## --------------------------------
## AIC                  6180.946
## BIC                  6296.821
## Log Likelihood      -3072.473
## Deviance             6144.946
```

```
## Num. obs.            4617
## ================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### Table A9 ######

m5 <- glm(vote_chal ~ treat + study, dbig,
          family = binomial())
texreg::screenreg(m5, digits = 3)
```

```
##
## ============================
##                 Model 1
## ----------------------------
## (Intercept)        -0.635 ***
##                    (0.080)
## treat               0.037
##                    (0.062)
## studyIL10           0.430 ***
##                    (0.108)
## studyIL3            0.188
##                    (0.107)
## studyKY 2011        0.586 ***
##                    (0.109)
## studyMA 2018        0.306 **
##                    (0.105)
## studyMD3            1.505 ***
##                    (0.149)
## studyMD6            1.363 ***
##                    (0.145)
## studyPA10           1.026 ***
##                    (0.141)
## studyPA12           1.195 ***
##                    (0.146)
## ----------------------------
## AIC              6170.493
## BIC              6234.868
## Log Likelihood  -3075.246
## Deviance         6150.493
## Num. obs.        4617
## ============================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```
###### power analysis ######

# STATA command for merged dataset and halved effect size
# power twoproportions .36 .4250, test(chi2) n1(2413) n2(2204)
```