Analysis and Results: Stigmatizing the Radical Right

AUTHOR
Ethan vanderWilden (with Laia Balcells, Sergi Martinez, Vicente Valentim

PUBLISHED
February 10, 2025

Description

Main quantitative analysis file to replicate all numbers, figures, and tables.

1. Load Packages, Data, and Helper Functions

```
# Clear the local environment
rm(list=ls())

# Set working directory
setwd("~/Downloads/stigmaBMVV")

# List of packages
pkg = c("tidyverse", "patchwork", "haven", "modelsummary", "Hmisc",
        "collapse", "attritevis", "expss", "factoextra", "kableExtra", "ggridges")

# Check and install the "attritevis" package from GitHub if necessary
if (!("attritevis" %in% rownames(installed.packages()))){
  library(devtools)
  install_github("lbassan/attritevis")
}

# Install missing packages
if (length(setdiff(pkg, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(pkg, rownames(installed.packages())))
}

# Load packages
suppressWarnings(suppressMessages(lapply(pkg, library, character.only = TRUE)))
```

```
[[1]]
 [1] "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"     "readr"
 [7] "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"     "graphics"
[13] "grDevices" "utils"     "datasets"  "methods"   "base"

[[2]]
 [1] "patchwork" "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"
 [7] "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"
[13] "graphics"  "grDevices" "utils"     "datasets"  "methods"   "base"

[[3]]
 [1] "haven"     "patchwork" "lubridate" "forcats"   "stringr"   "dplyr"
 [7] "purrr"     "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse"
[13] "stats"     "graphics"  "grDevices" "utils"     "datasets"  "methods"
```

```
[19] "base"

[[4]]
 [1] "modelsummary" "haven"        "patchwork"    "lubridate"    "forcats"
 [6] "stringr"      "dplyr"        "purrr"        "readr"        "tidyr"
[11] "tibble"       "ggplot2"      "tidyverse"    "stats"        "graphics"
[16] "grDevices"    "utils"        "datasets"     "methods"      "base"

[[5]]
 [1] "Hmisc"        "modelsummary" "haven"        "patchwork"    "lubridate"
 [6] "forcats"      "stringr"      "dplyr"        "purrr"        "readr"
[11] "tidyr"        "tibble"       "ggplot2"      "tidyverse"    "stats"
[16] "graphics"     "grDevices"    "utils"        "datasets"     "methods"
[21] "base"

[[6]]
 [1] "collapse"     "Hmisc"        "modelsummary" "haven"        "patchwork"
 [6] "lubridate"    "forcats"      "stringr"      "dplyr"        "purrr"
[11] "readr"        "tidyr"        "tibble"       "ggplot2"      "tidyverse"
[16] "stats"        "graphics"     "grDevices"    "utils"        "datasets"
[21] "methods"      "base"

[[7]]
 [1] "attritevis"   "collapse"     "Hmisc"        "modelsummary" "haven"
 [6] "patchwork"    "lubridate"    "forcats"      "stringr"      "dplyr"
[11] "purrr"        "readr"        "tidyr"        "tibble"       "ggplot2"
[16] "tidyverse"    "stats"        "graphics"     "grDevices"    "utils"
[21] "datasets"     "methods"      "base"

[[8]]
 [1] "expss"        "maditr"       "attritevis"   "collapse"     "Hmisc"
 [6] "modelsummary" "haven"        "patchwork"    "lubridate"    "forcats"
[11] "stringr"      "dplyr"        "purrr"        "readr"        "tidyr"
[16] "tibble"       "ggplot2"      "tidyverse"    "stats"        "graphics"
[21] "grDevices"    "utils"        "datasets"     "methods"      "base"

[[9]]
 [1] "factoextra"   "expss"        "maditr"       "attritevis"   "collapse"
 [6] "Hmisc"        "modelsummary" "haven"        "patchwork"    "lubridate"
[11] "forcats"      "stringr"      "dplyr"        "purrr"        "readr"
[16] "tidyr"        "tibble"       "ggplot2"      "tidyverse"    "stats"
[21] "graphics"     "grDevices"    "utils"        "datasets"     "methods"
[26] "base"

[[10]]
 [1] "kableExtra"   "factoextra"   "expss"        "maditr"       "attritevis"
 [6] "collapse"     "Hmisc"        "modelsummary" "haven"        "patchwork"
[11] "lubridate"    "forcats"      "stringr"      "dplyr"        "purrr"
[16] "readr"        "tidyr"        "tibble"       "ggplot2"      "tidyverse"
[21] "stats"        "graphics"     "grDevices"    "utils"        "datasets"
[26] "methods"      "base"

[[11]]
 [1] "ggridges"     "kableExtra"   "factoextra"   "expss"        "maditr"
```

```
 [6] "attritevis"   "collapse"     "Hmisc"        "modelsummary" "haven"
[11] "patchwork"    "lubridate"    "forcats"      "stringr"      "dplyr"
[16] "purrr"        "readr"        "tidyr"        "tibble"       "ggplot2"
[21] "tidyverse"    "stats"        "graphics"     "grDevices"    "utils"
[26] "datasets"     "methods"      "base"
```

```
# Load data with incompletes
df_incompletes <- read_sav("df_incompletesBMVV.sav")

# Load data with incompletes removed
data <- read_sav("dataBMVV.sav")

# Alternatively, can run the following line to remove incompletes:
# data <- sbt(df_incompletes, quality.check == 7 & duration > 90 & duration < 1800 & F

# Load 'helper functions' file
source("functionsBMVV.R")
```

2. Data Manipulation

2.1 Flip "Acceptability of Radical Right Questions"

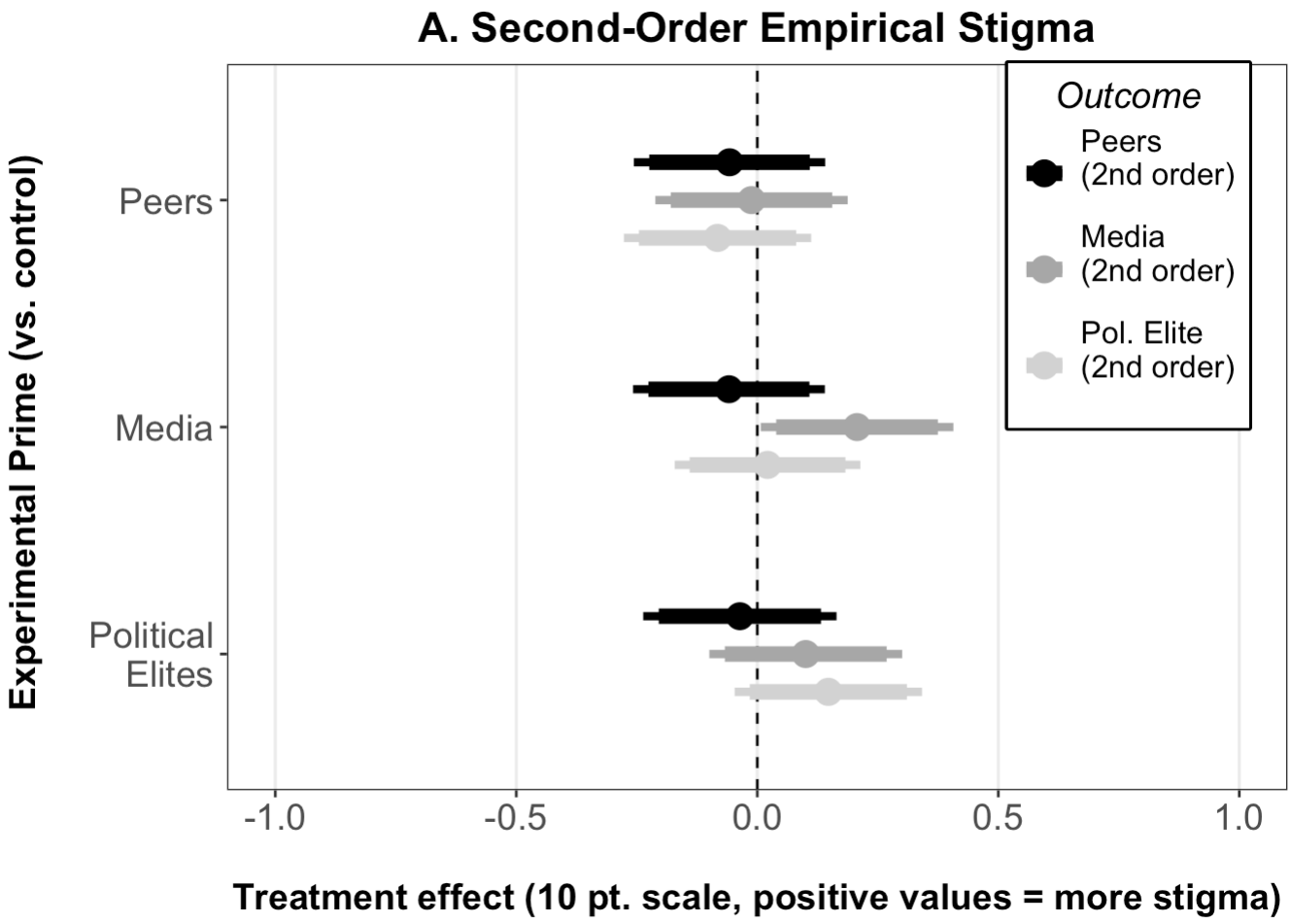
```
# Recode variables to flip such that high stigma = high value
data$first.accept <- 10 - data$first.accept
data$media.accept <- 10 - data$media.accept
data$peer.accept <- 10 - data$peer.accept
data$elite.accept <- 10 - data$elite.accept

# Adjust variable labels
data = apply_labels(data,
                    first.accept = paste(as.character(var_lab(data[,"first.accept"]),
                    media.accept = paste(as.character(var_lab(data[,"media.accept"]),
                    peer.accept = paste(as.character(var_lab(data[,"peer.accept"]), ",
                    elite.accept = paste(as.character(var_lab(data[,"elite.accept"]),

## 2.2 Add Manually Created Variables / Indices
data$second.accept <- rowMeans(data[, c(24:26)], na.rm = TRUE)
data$first.sanction.index <- rowMeans(data[,c(27:32)], na.rm = TRUE)
data$second.sanction.index <- rowMeans(data[,c(33:38)], na.rm = TRUE)

data$gen.stigma <- (data$first.accept + data$second.accept + data$first.sanction.index

data = apply_labels(data,
                    second.accept = "high values indicate higher perceived second orde
                    first.sanction.index = "high values indicate more willingness to s
                    second.sanction.index = "high values indicate higher expectation t
                    gen.stigma = "index of 4 stigma outcomes (average), high values in
```

3. Main Results

Figures 2 [a and b]

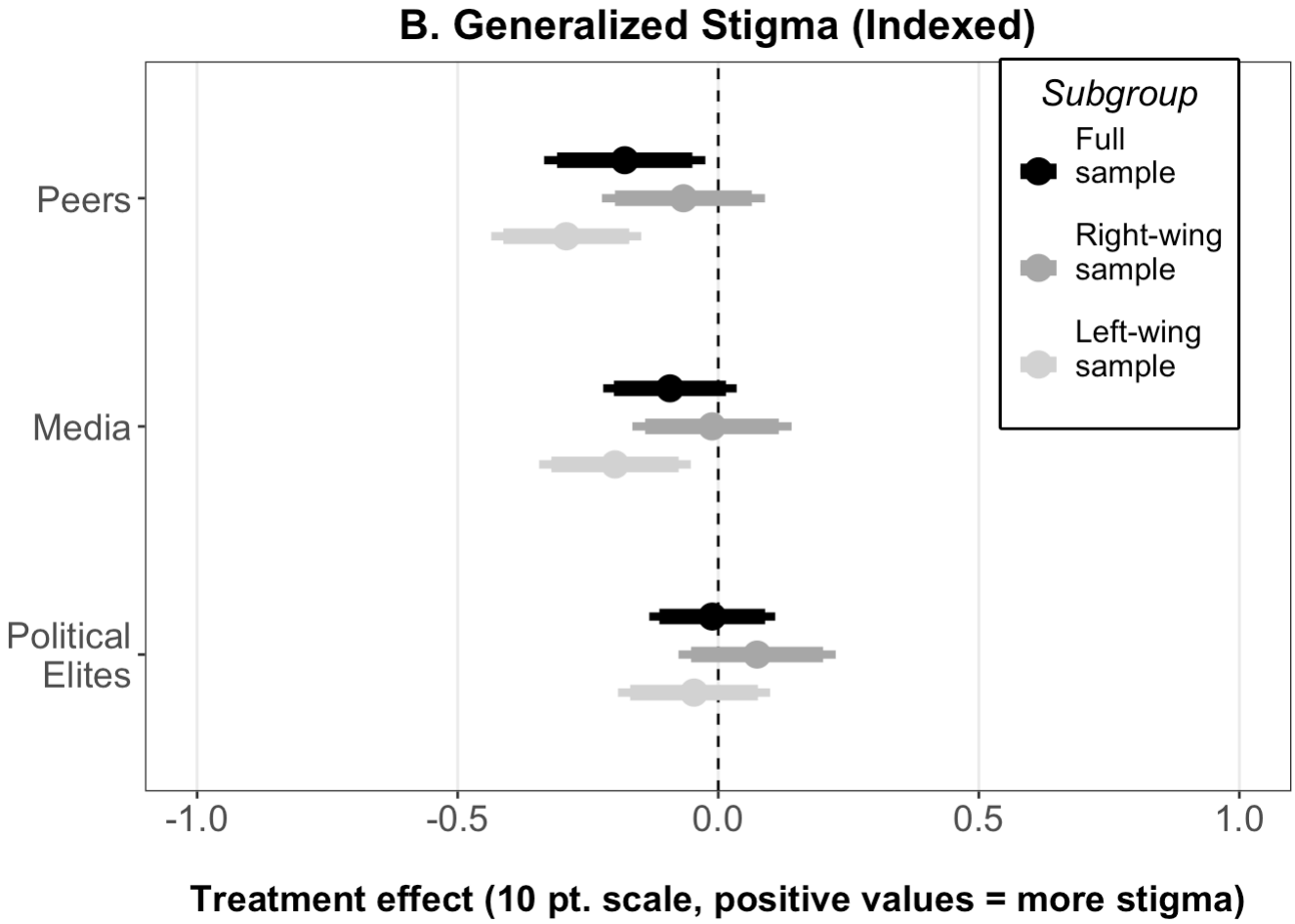
```
get_plot(validation_coefs(data, controls = FALSE), mytitle = "A. Second-Order Empirica
```

```
Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
ℹ Please use the `linewidth` argument instead.

Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
3.5.0.
ℹ Please use the `legend.position.inside` argument of `theme()` instead.

Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```




```
get_plot(reg_coefs(data, controls = FALSE), mytitle = "B. Generalized Stigma (Indexed)
  theme(axis.title.y = element_blank())
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```




```
## Estimates included in the written section
summary(lm(elite.accept ~ factor(treat), data = sbt(data, data$treat %in% c(0, 3)))) #
```

```
Call:
lm(formula = elite.accept ~ factor(treat), data = sbt(data, data$treat %in%
    c(0, 3)))

Residuals:
LABEL: En una escala de 0 a 10, ¿Cómo de aceptado diría que se considera a Vox
entre... - los políticos españoles?
VALUES:
-6.5007, -1.5007, 0.4993, 1.6467, 3.6467

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.35332    0.06984  90.972   <2e-16 ***
factor(treat)3  0.14741    0.09909   1.488    0.137
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.598 on 2748 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.0008047, Adjusted R-squared:  0.000441
F-statistic: 2.213 on 1 and 2748 DF,  p-value: 0.137
```

```
round(mean(sbt(data, data$treat == 0 & data$ideology < 5)$gen.stigma, na.rm = TRUE), 2
```

```
[1] 5.28
```

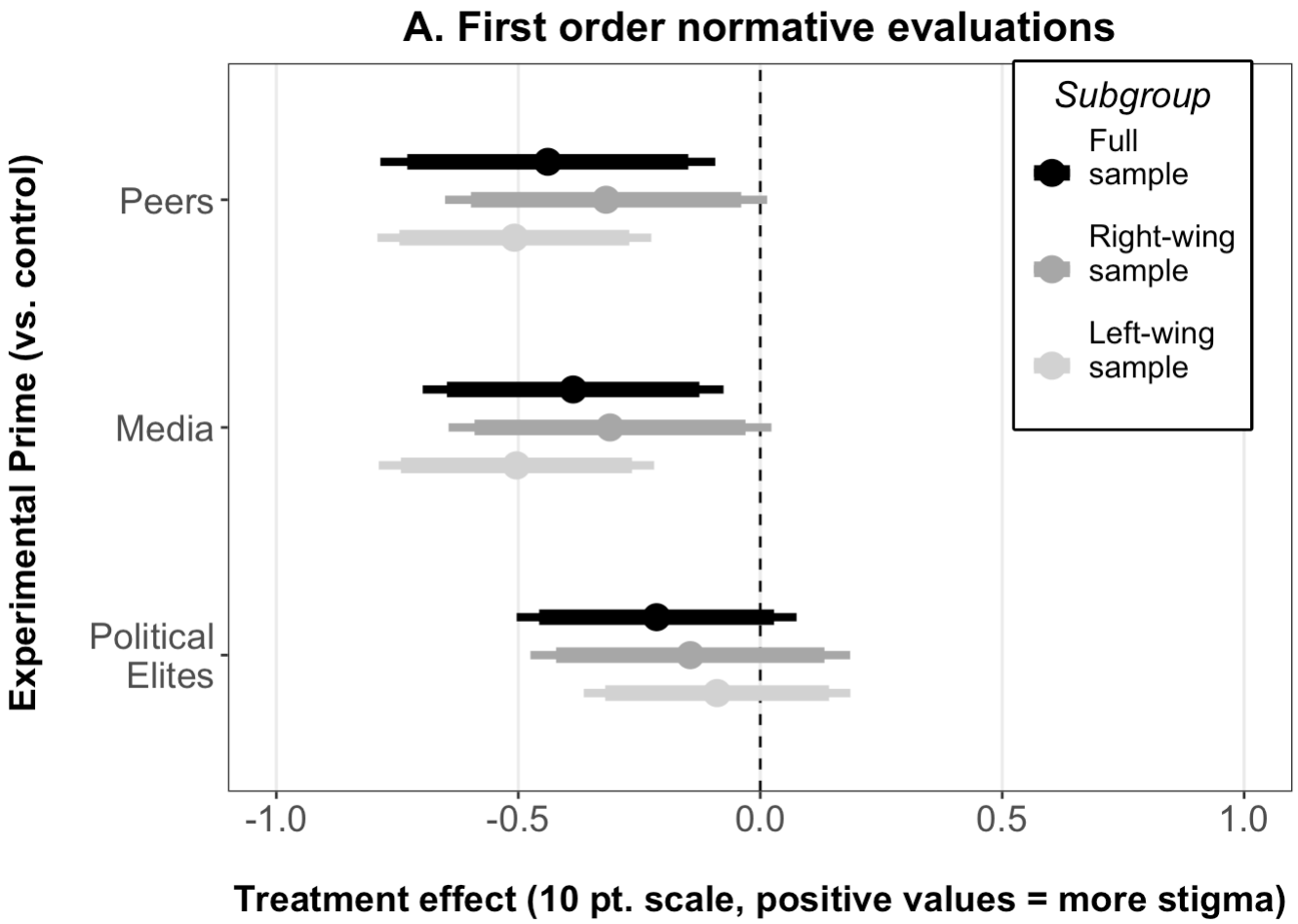
```
round(mean(sbt(data, data$treat == 0 & data$ideology >= 5)$gen.stigma, na.rm = TRUE),
```

```
[1] 3.65
```

Figure 3 [a - d]

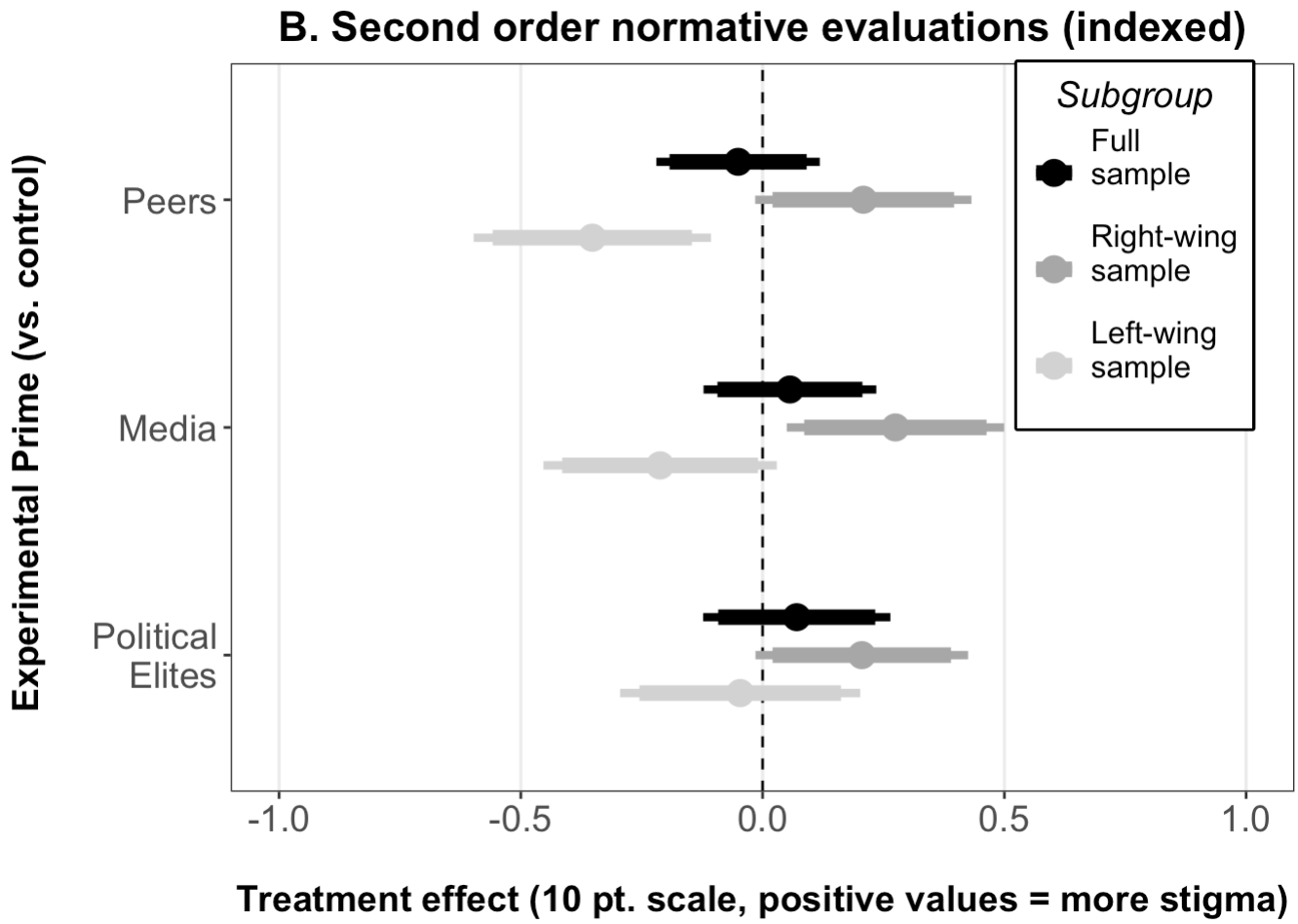
```
get_plot(reg_coefs(data, outcome = "first.accept", controls = FALSE), mytitle = "A. Fi
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



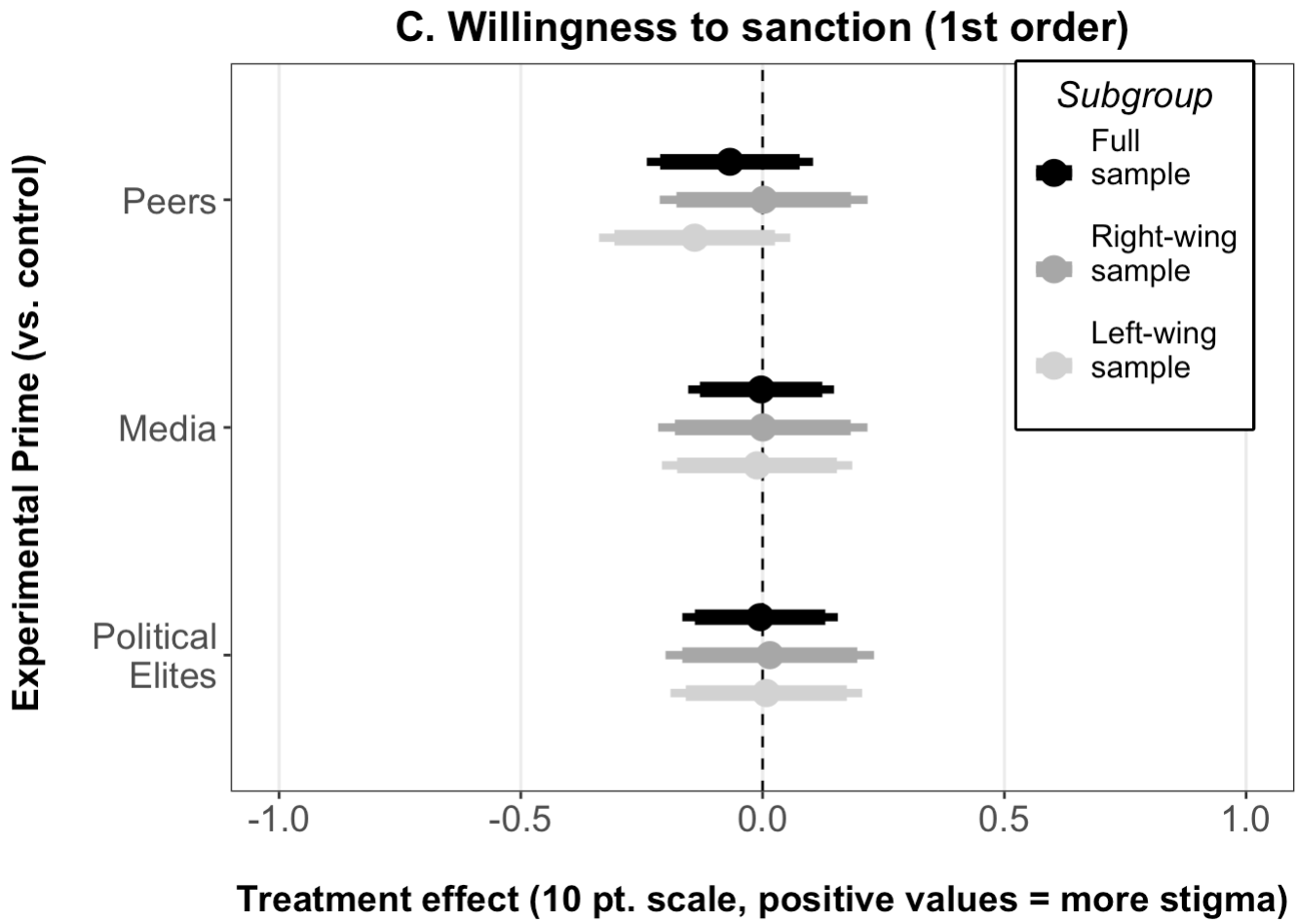

```
get_plot(reg_coefs(data, outcome = "second.accept", controls = FALSE), mytitle = "B. S
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



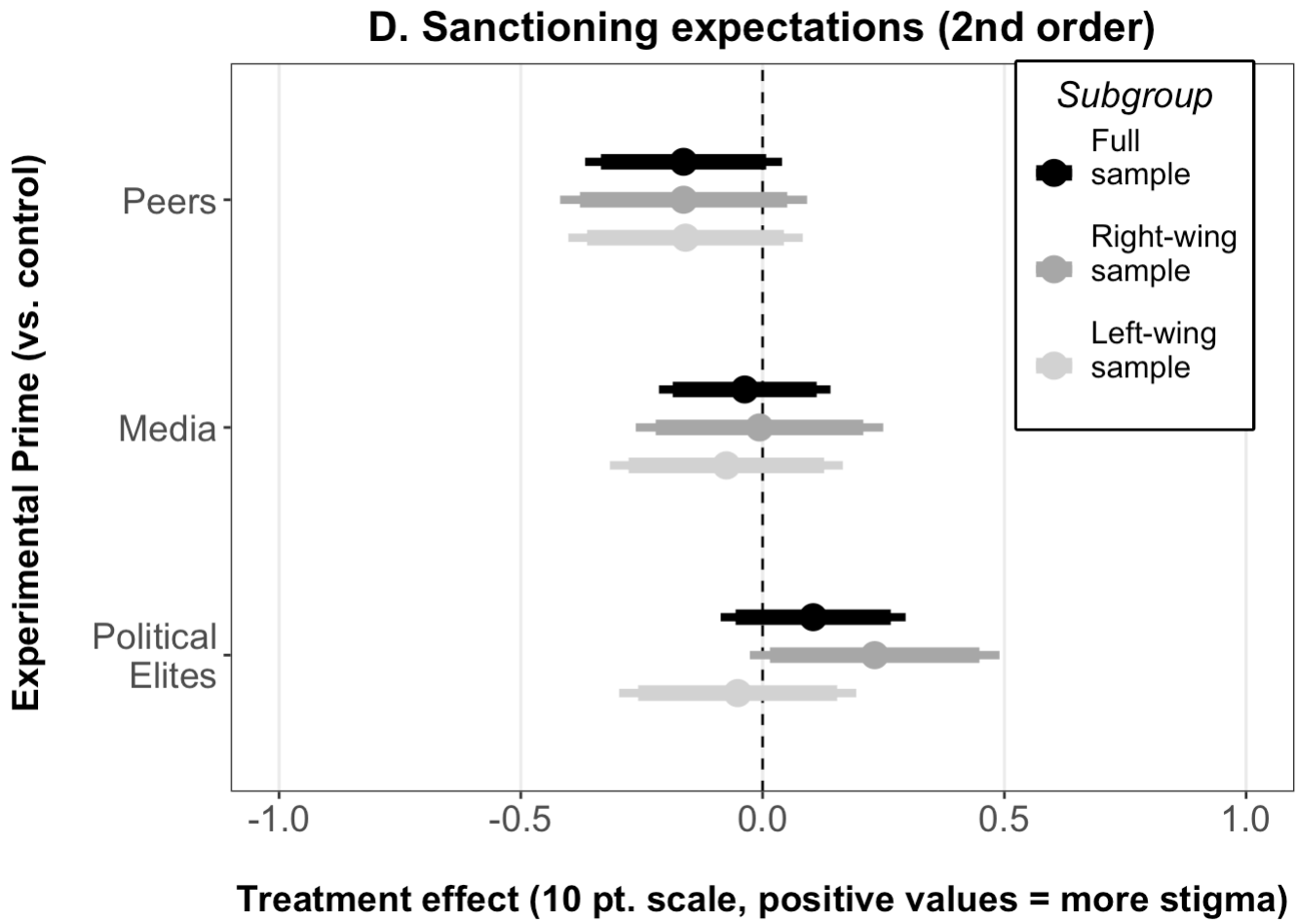

```
get_plot(reg_coefs(data, outcome = "first.sanction.index", controls = FALSE), mytitle
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



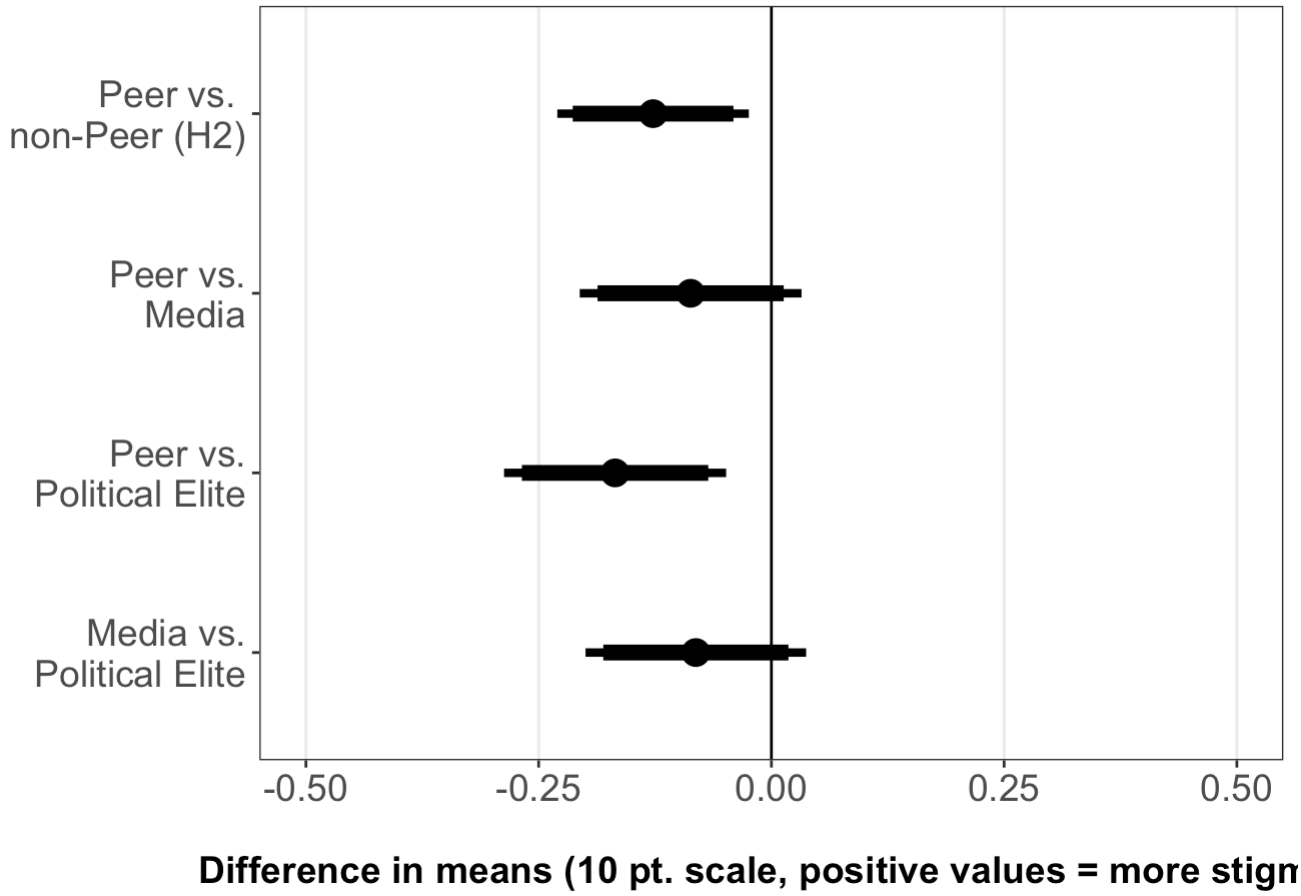

```
get_plot(reg_coefs(data, outcome = "second.sanction.index", controls = FALSE), mytitle
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```




```
## Figure 4
get_paired(data, controls = FALSE) # Render at 750 x 500
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
`position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



4. Appendix

A.1 Demographic Representative Assessment

```
# Copy dataset for adjustments
repdata <- data

# Get percentages in the data for each variable
# Create demographic indicators
repdata$M1 <- data$gender
repdata$AG1 <- ifelse(repdata$age %in% c(18:25), 1, 0)
repdata$AG2 <- ifelse(repdata$age %in% c(26:35), 1, 0)
repdata$AG3 <- ifelse(repdata$age %in% c(36:45), 1, 0)
repdata$AG4 <- ifelse(repdata$age %in% c(46:55), 1, 0)
repdata$AG5 <- ifelse(repdata$age %in% c(56:65), 1, 0)
repdata$AG6 <- ifelse(repdata$age > 65, 1, 0)
repdata$R1 <- ifelse(repdata$region.char == "Andalucía", 1, 0)
repdata$R2 <- ifelse(repdata$region.char == "Aragón", 1, 0)
repdata$R3 <- ifelse(repdata$region.char == "Asturias", 1, 0)
repdata$R4 <- ifelse(repdata$region.char == "las Islas Baleares", 1, 0)
repdata$R5 <- ifelse(repdata$region.char == "las Islas Canarias", 1, 0)
repdata$R6 <- ifelse(repdata$region.char == "Cantabria", 1, 0)
repdata$R7 <- ifelse(repdata$region.char == "Castilla y León", 1, 0)
repdata$R8 <- ifelse(repdata$region.char == "Castilla-La Mancha", 1, 0)
repdata$R9 <- ifelse(repdata$region.char == "Catalunya", 1, 0)
```

```
repdata$R10 <- ifelse(repdata$region.char == "Valencia", 1, 0)
repdata$R11 <- ifelse(repdata$region.char == "Extremadura", 1, 0)
repdata$R12 <- ifelse(repdata$region.char == "Galicia", 1, 0)
repdata$R13 <- ifelse(repdata$region.char == "Madrid", 1, 0)
repdata$R14 <- ifelse(repdata$region.char == "Murcia", 1, 0)
repdata$R15 <- ifelse(repdata$region.char == "Navarra", 1, 0)
repdata$R16 <- ifelse(repdata$region.char == "País Vasco", 1, 0)
repdata$R17 <- ifelse(repdata$region.char == "La Rioja", 1, 0)

# Create dataframe to store results
rep.tab <- data.frame(
  row.names = c("Male", "18-25", "26-35", "36-45", "46-55", "56-65", "66+",
                "Andalucía", "Aragón", "Asturias", "las Islas Baleares", "las Islas Ca
                "Cantabria", "Castilla y León", "Castilla-La Mancha", "Catalunya", "Va
                "Extremadura", "Galicia", "Madrid", "Murcia", "Navarra", "País Vasco",

  sample.prop = rep(NA, 24),

  # Census data from 2023 INE estimates
  census.prop = c(0.490, 0.097, 0.131, 0.169, 0.187, 0.158, 0.256,
                  0.179, 0.028, 0.021, 0.025, 0.046, 0.012, 0.050, 0.043,
                  0.165, 0.109, 0.022, 0.056, 0.143, 0.032, 0.014, 0.046, 0.007),
  difference = rep(NA, 24),
  p.value = rep(NA, 24)
)

# Calculate demographic indicators
for (i in 1:24){
  rep.tab$sample.prop[i] <- round(mean(repdata[, i + 45], na.rm = TRUE), 3)
  rep.tab$difference[i] <- round(abs(rep.tab$sample.prop[i] - rep.tab$census.prop[i]),
  rep.tab$p.value[i] <- round(t.test(repdata[, i + 45], mu = rep.tab$census.prop[i], a
}

# Print latex friendly table - TABLE A1
kbl(rep.tab, format = 'html', booktabs = TRUE)
```

	sample.prop	census.prop	difference	p.value
Male	0.497	0.490	0.007	0.279
18-25	0.095	0.097	0.002	0.581
26-35	0.129	0.131	0.002	0.662
36-45	0.167	0.169	0.002	0.694
46-55	0.186	0.187	0.001	0.880
56-65	0.163	0.158	0.005	0.345
66+	0.260	0.256	0.004	0.474
Andalucía	0.176	0.179	0.003	0.521
Aragón	0.029	0.028	0.001	0.792
Asturias	0.020	0.021	0.001	0.699
las Islas Baleares	0.025	0.025	0.000	0.874

	sample.prop	census.prop	difference	p.value
las Islas Canarias	0.048	0.046	0.002	0.578
Cantabria	0.012	0.012	0.000	0.837
Castilla y León	0.050	0.050	0.000	0.917
Castilla-La Mancha	0.043	0.043	0.000	0.875
Catalunya	0.168	0.165	0.003	0.609
Valencia	0.109	0.109	0.000	0.920
Extremadura	0.022	0.022	0.000	0.958
Galicia	0.055	0.056	0.001	0.839
Madrid	0.144	0.143	0.001	0.885
Murcia	0.031	0.032	0.001	0.827
Navarra	0.013	0.014	0.001	0.694
País Vasco	0.047	0.046	0.001	0.621
La Rioja	0.007	0.007	0.000	0.912

A.2 Balance

```
# Run t-tests and ANOVA for demographics

# Age
tInfo <- t.test(age ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(age ~ factor(treat), data = data))[1,5])
Age <- c(range(na.omit(data$age)),
         as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
         as.numeric(tInfo$p.value), anova_p)

# Gender
tInfo <- t.test(gender ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(gender ~ factor(treat), data = data))[1,5])
Gender <- c(range(na.omit(data$gender)),
            as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
            as.numeric(tInfo$p.value), anova_p)

# Education
tInfo <- t.test(education ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(education ~ factor(treat), data = data))[1,5])
Education <- c(range(na.omit(data$education)),
               as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
               as.numeric(tInfo$p.value), anova_p)

# Income
tInfo <- t.test(income ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(income ~ factor(treat), data = data))[1,5])
Income <- c(range(na.omit(data$income)),
            as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
```

```
            as.numeric(tInfo$p.value), anova_p)

# Political Knowledge
tInfo <- t.test(pol.know ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(pol.know ~ factor(treat), data = data))[1,5])
Political_Knowledge <- c(range(na.omit(data$pol.know)),
                         as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
                         as.numeric(tInfo$p.value), anova_p)

# Spanish Nationalism
tInfo <- t.test(SP.nationalism ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(SP.nationalism ~ factor(treat), data = data))[1,5])
Spanish_Nationalism <- c(range(na.omit(data$SP.nationalism)),
                         as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
                         as.numeric(tInfo$p.value), anova_p)

# Regional Identity
tInfo <- t.test(CCAA.nationalism ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(CCAA.nationalism ~ factor(treat), data = data))[1,5])
Regional_Identity <- c(range(na.omit(data$CCAA.nationalism)),
                       as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
                       as.numeric(tInfo$p.value), anova_p)

# Ideology
tInfo <- t.test(ideology ~ anytreat, data = data)
anova_p <- as.numeric(anova(lm(ideology ~ factor(treat), data = data))[1,5])
Ideology <- c(range(na.omit(data$ideology)),
              as.numeric(tInfo$estimate[2]), as.numeric(tInfo$estimate[1]),
              as.numeric(tInfo$p.value), anova_p)

# Combine results and re-name columns
balance_test <- as.data.frame(rbind(Age, Gender, Education, Income, Political_Knowledg
                                    Spanish_Nationalism, Regional_Identity, Ideology))
names(balance_test) <- c("Low", "High", "Mean Treated", "Mean Control", "P-value", "AN

# Round to 2 digits and print in latex friendly form
balance_test[,3:6] <- round(balance_test[,3:6], 2)

# Print TABLE A2
kbl(balance_test, format = "html", booktabs = TRUE)
```

	Low	High	Mean Treated	Mean Control	P-value	ANOVA p-value
Age	18	80	50.37	49.93	0.40	0.61
Gender	0	1	0.50	0.50	0.82	0.73
Education	1	8	5.05	4.96	0.03	0.02
Income	1	9	6.00	5.95	0.31	0.36
Political_Knowledge	0	1	0.60	0.57	0.17	0.49
Spanish_Nationalism	0	10	7.63	7.57	0.52	0.57
Regional_Identity	0	10	7.86	7.68	0.04	0.04
Ideology	0	10	4.79	4.78	0.96	0.56

A.3 Post-intervention Attrition

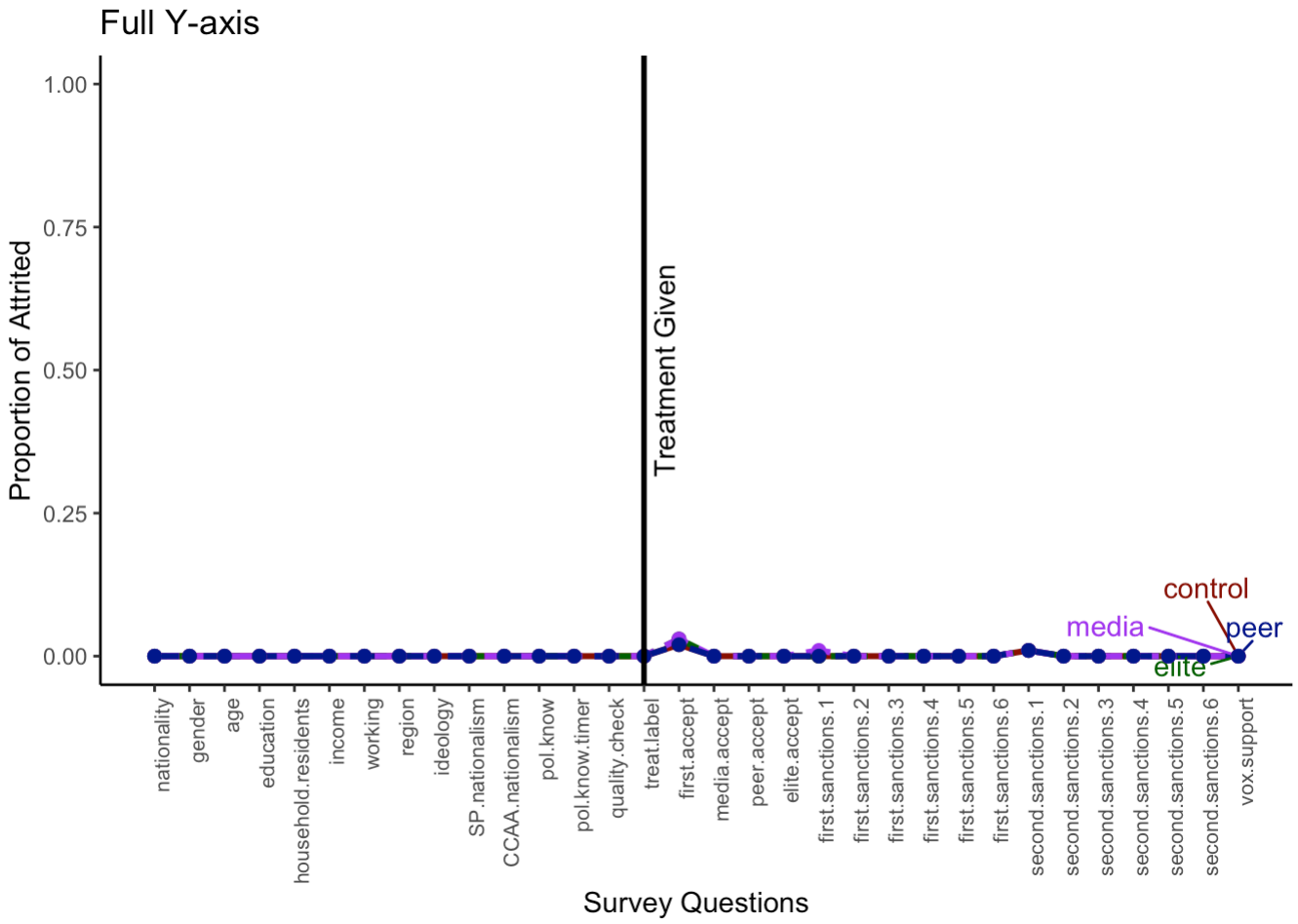
```
# Reported in appendix, attrition section
full <- sbt(df_incompletes, df_incompletes$quality.check == 7) # remove 2817
full <- sbt(full, full$duration < 1800) # remove 53
full <- sbt(full, full$duration > 90) # remove 34

# Save only the relevant, ordered data
names(full)
```

```
 [1] "duration"           "date"               "nationality"
 [4] "gender"             "age"                "education"
 [7] "household.residents" "income"            "working"
[10] "region"             "ideology"           "SP.nationalism"
[13] "CCAA.nationalism"   "pol.know"           "pol.know.timer"
[16] "quality.check"      "treat.label"        "peer.prime"
[19] "treat"              "anytreat"           "manip"
[22] "treat.time"         "first.accept"       "media.accept"
[25] "peer.accept"        "elite.accept"       "first.sanctions.1"
[28] "first.sanctions.2"  "first.sanctions.3"  "first.sanctions.4"
[31] "first.sanctions.5"  "first.sanctions.6"  "second.sanctions.1"
[34] "second.sanctions.2" "second.sanctions.3" "second.sanctions.4"
[37] "second.sanctions.5" "second.sanctions.6" "vox.support"
[40] "Finished"           "region.char"
```

```
dfAT <- as.data.frame(full[, c(3:17, 23:39)])

# FIGURE A1 - Render at both at 850 x 500
plot_attrition(dfAT, treatment_q = 'treat.label', freq = FALSE, total = FALSE, title =
               mycolors = c(control = "darkred", elite = "darkgreen", peer = "darkblue
```

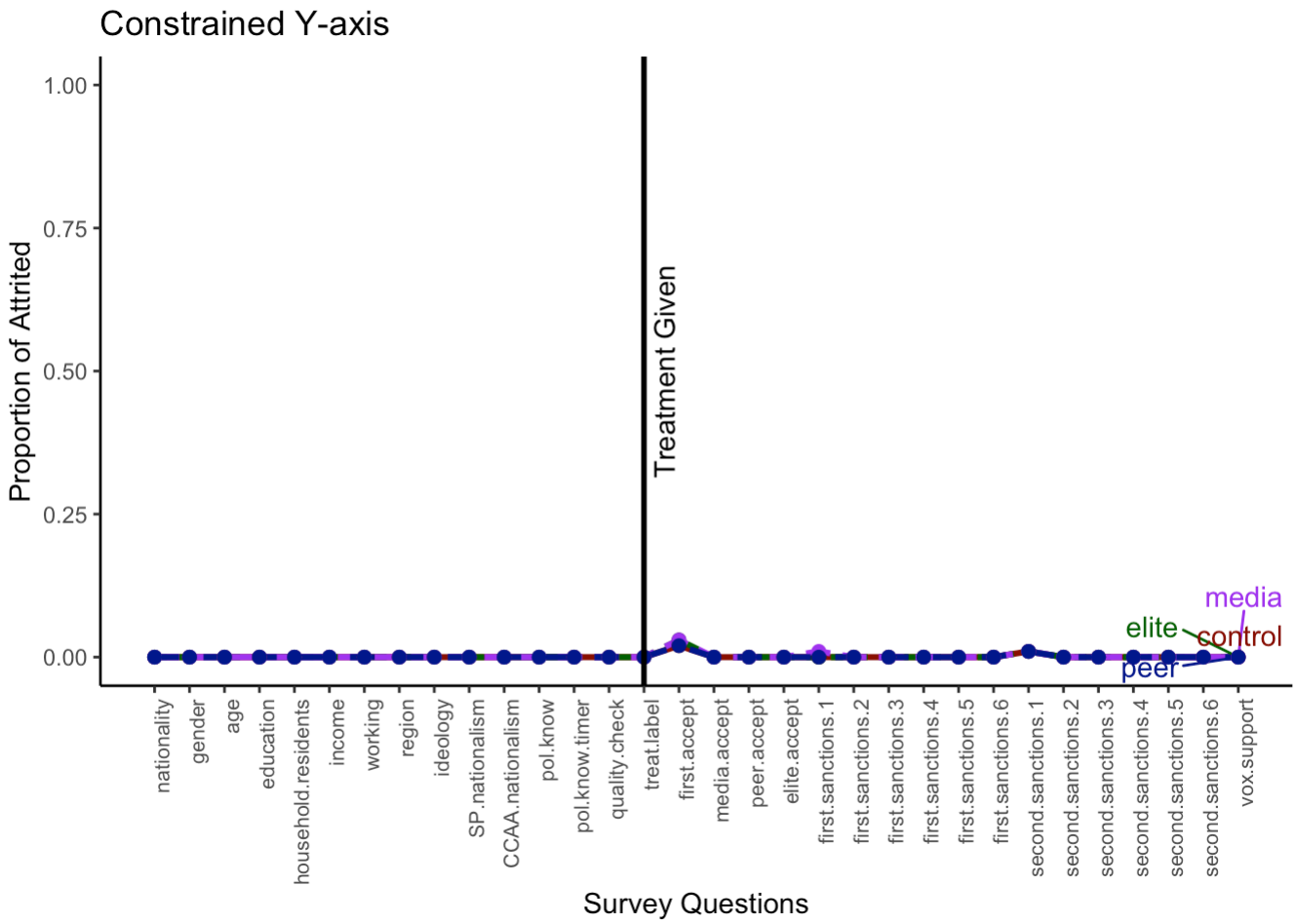
```
Loading required package: viridis

Loading required package: viridisLite

Loading required package: ggrepel

Loading required package: data.table


Attaching package: 'data.table'

The following objects are masked from 'package:expss':

    copy, like

The following objects are masked from 'package:maditr':

    copy, dcast, let, melt

The following object is masked from 'package:collapse':
```

```
    fdroplevels
The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:dplyr':

    between, first, last

The following object is masked from 'package:purrr':

    transpose
```

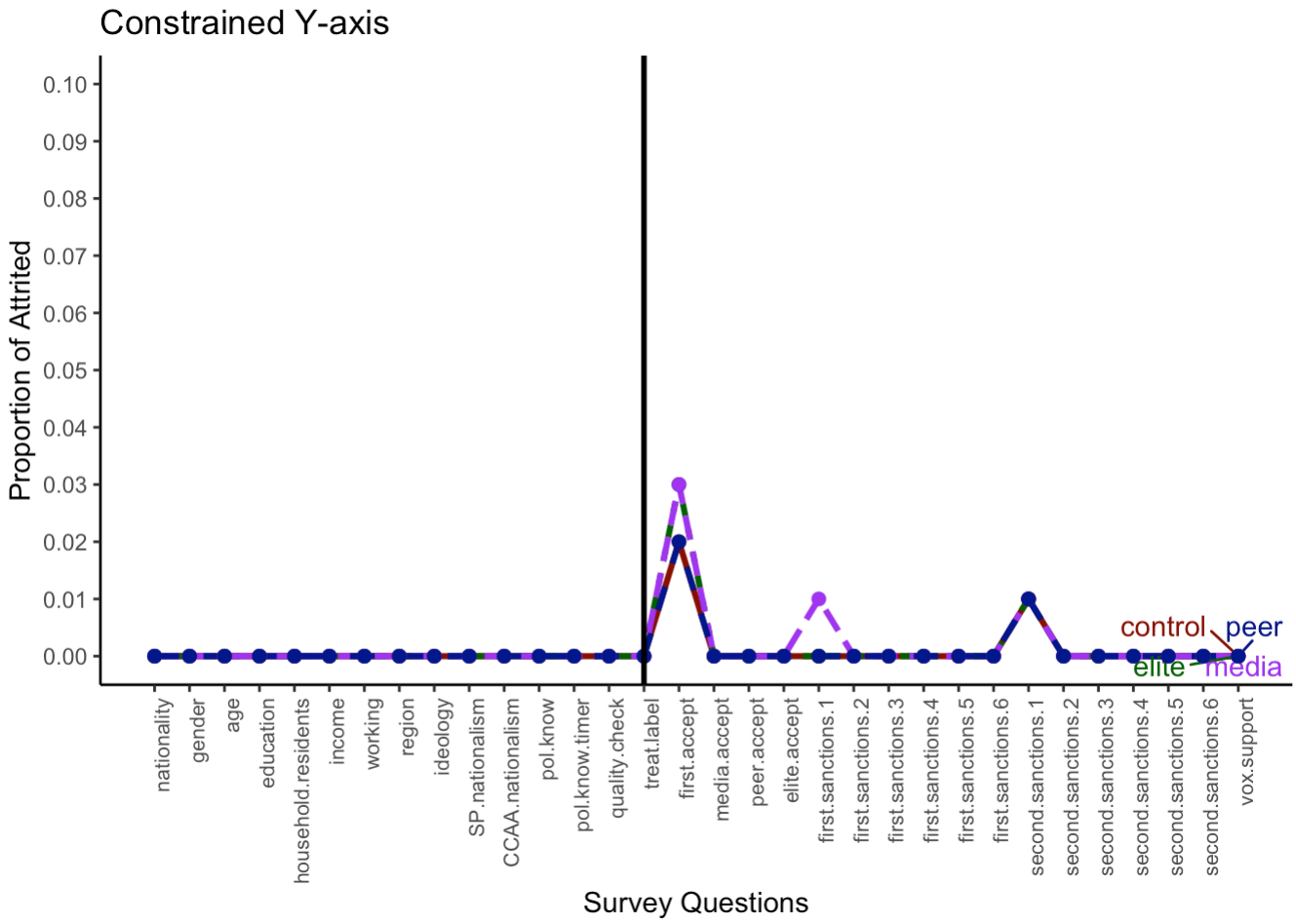



```
plot_attrition(dfAT, treatment_q = 'treat.label', freq = FALSE, total = FALSE, title =
               mycolors = c(control = "darkred", elite = "darkgreen", peer = "darkblue
  scale_y_continuous(limits = c(0, 0.1), breaks = seq(0, 0.1, 0.01))
```

```
Scale for y is already present.
Adding another scale for y, which will replace the existing scale.
```




```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_text()`).
```



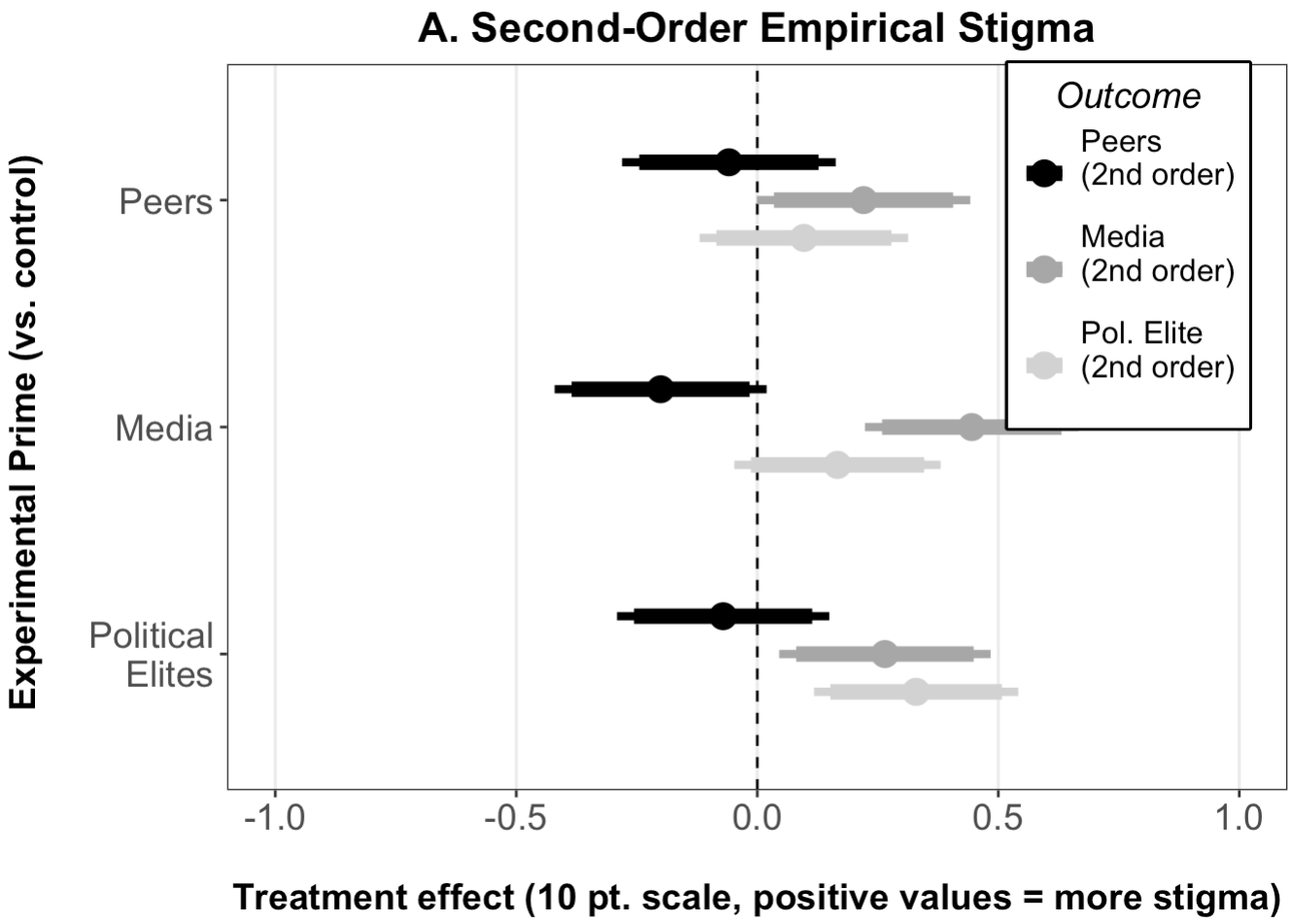
A.4 Factual Manipulation Check

```
# Check treatment identification
prop.table(table(data[data$treat != 0, ]$manip)) # About 69% of treated respondents co
```

```
        1         2         3 
0.6876964 0.2284264 0.0838772 
```

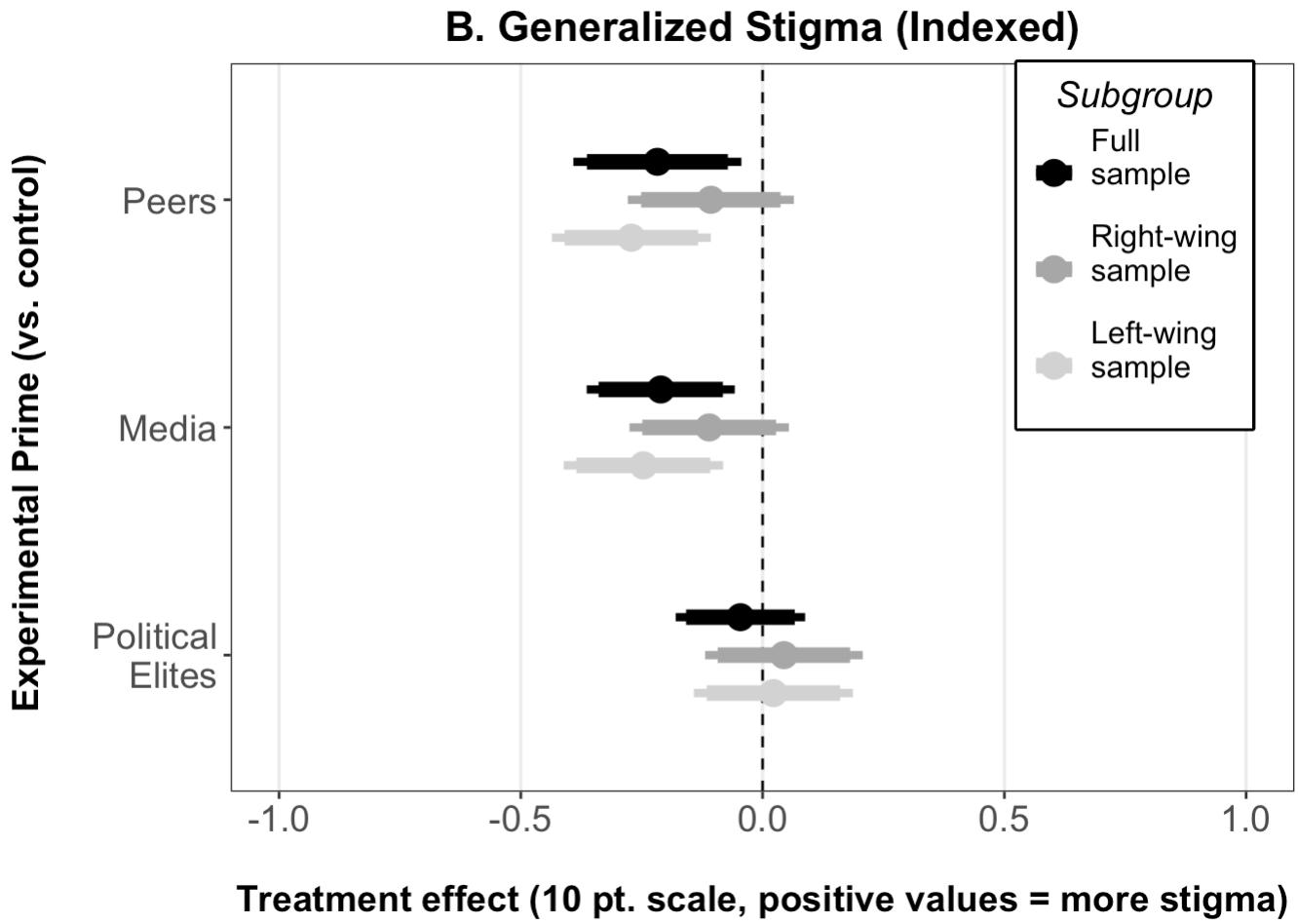
```
# Subset data for only 'successfully' treated
data$success <- ifelse(data$treat > 0, ifelse(data$manip == 1, 1, 0), 1)
ensured <- sbt(data, data$success == 1)

# FIGURE A2
get_plot(validation_coefs(ensured, controls = FALSE), mytitle = "A. Second-Order Empir
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



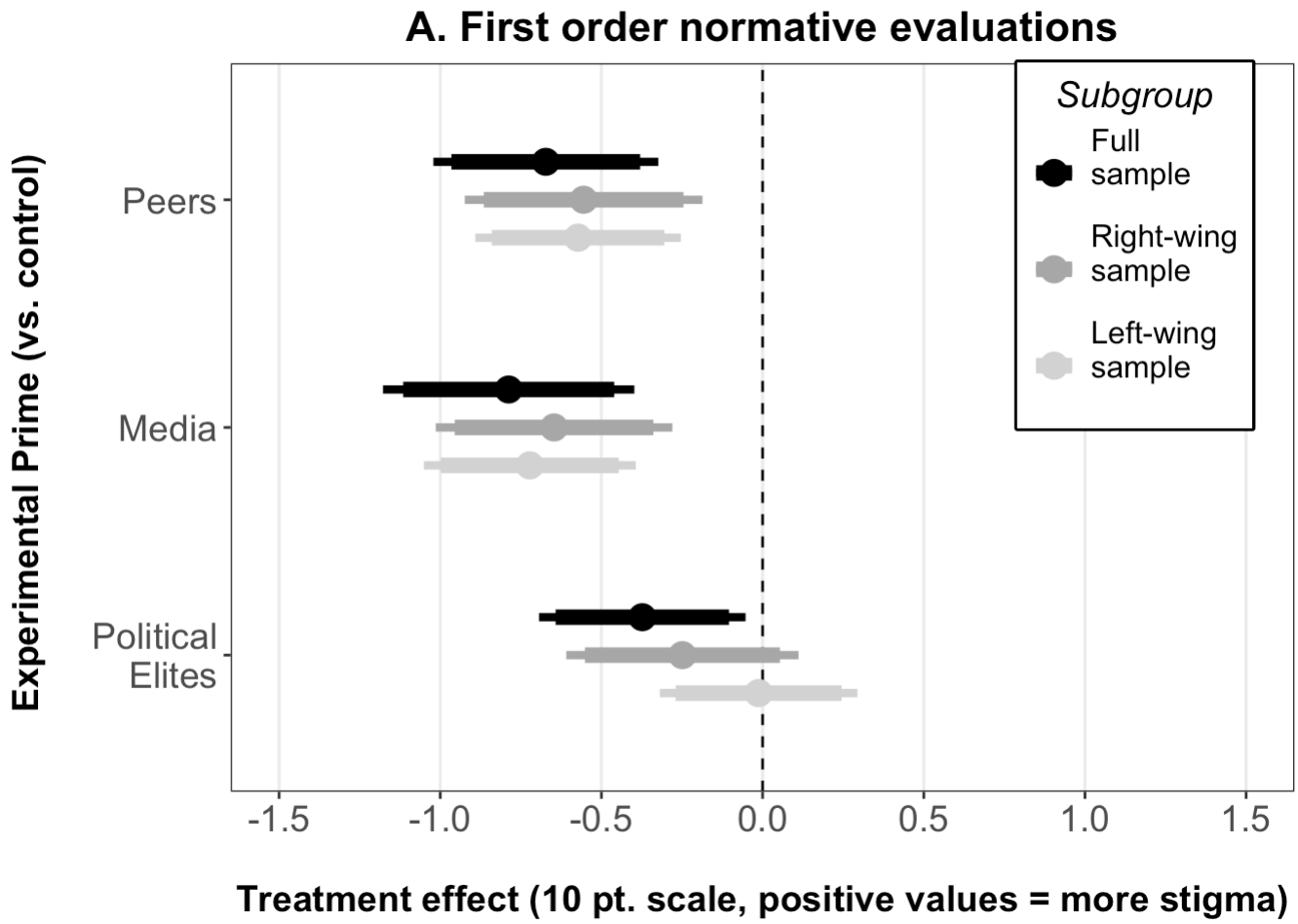

```
get_plot(reg_coefs(ensured, controls = FALSE), mytitle = "B. Generalized Stigma (Index
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



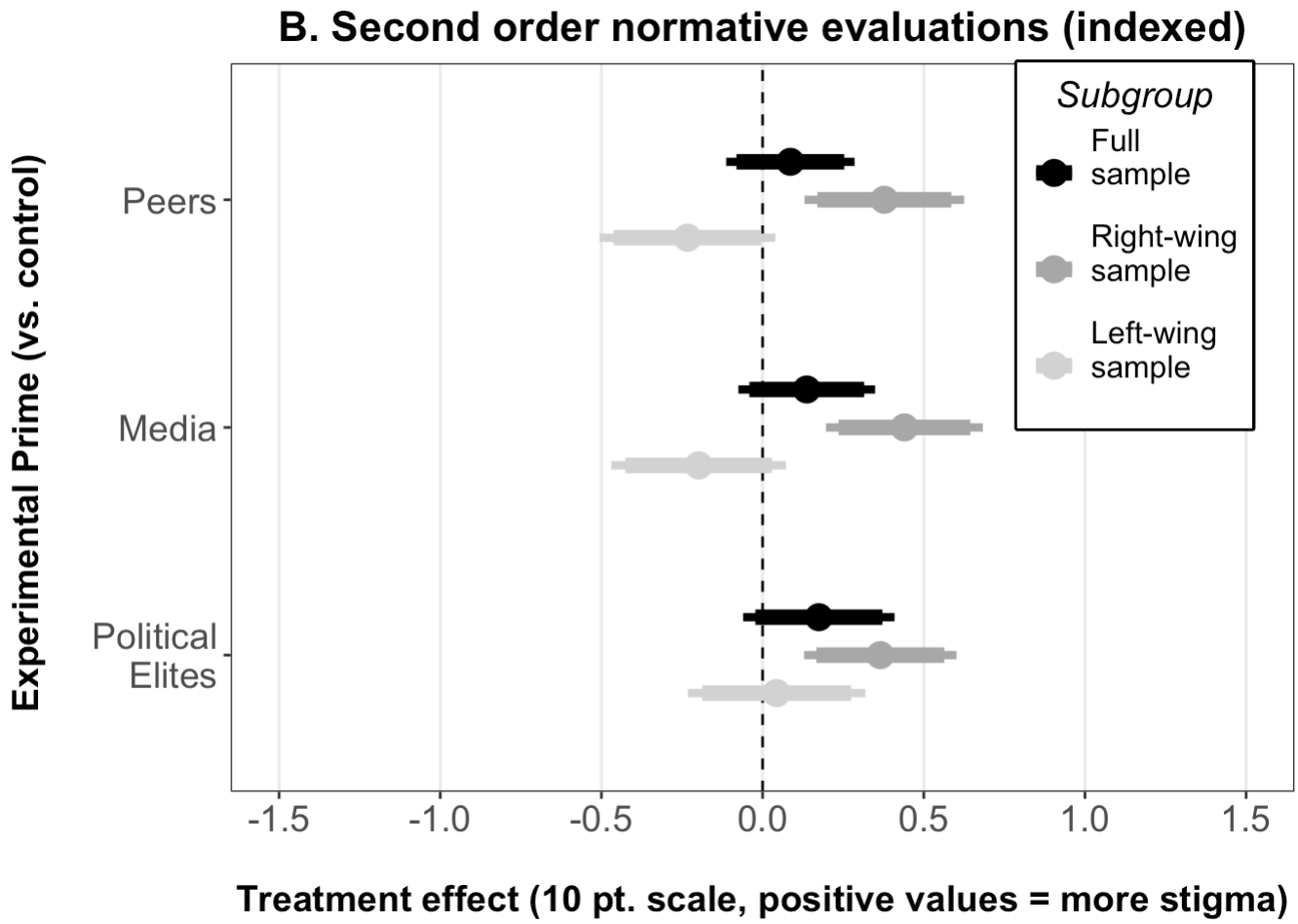

```
# FIGURE A3
get_plot(reg_coefs(ensured, outcome = "first.accept", controls = FALSE),
         mytitle = "A. First order normative evaluations", range = 1.5)
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



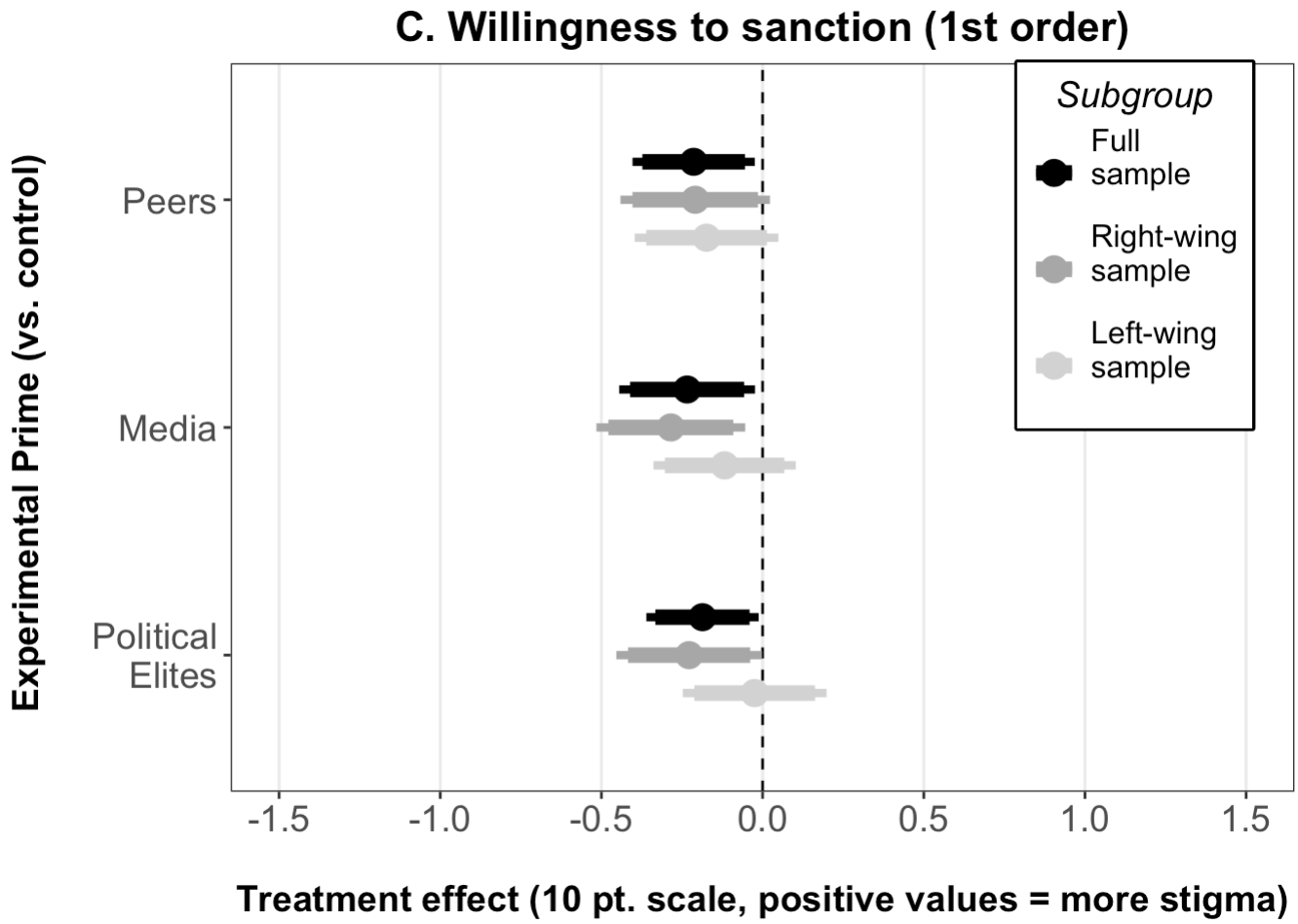

```
get_plot(reg_coefs(ensured, outcome = "second.accept", controls = FALSE),
         mytitle = "B. Second order normative evaluations (indexed)", range = 1.5)
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



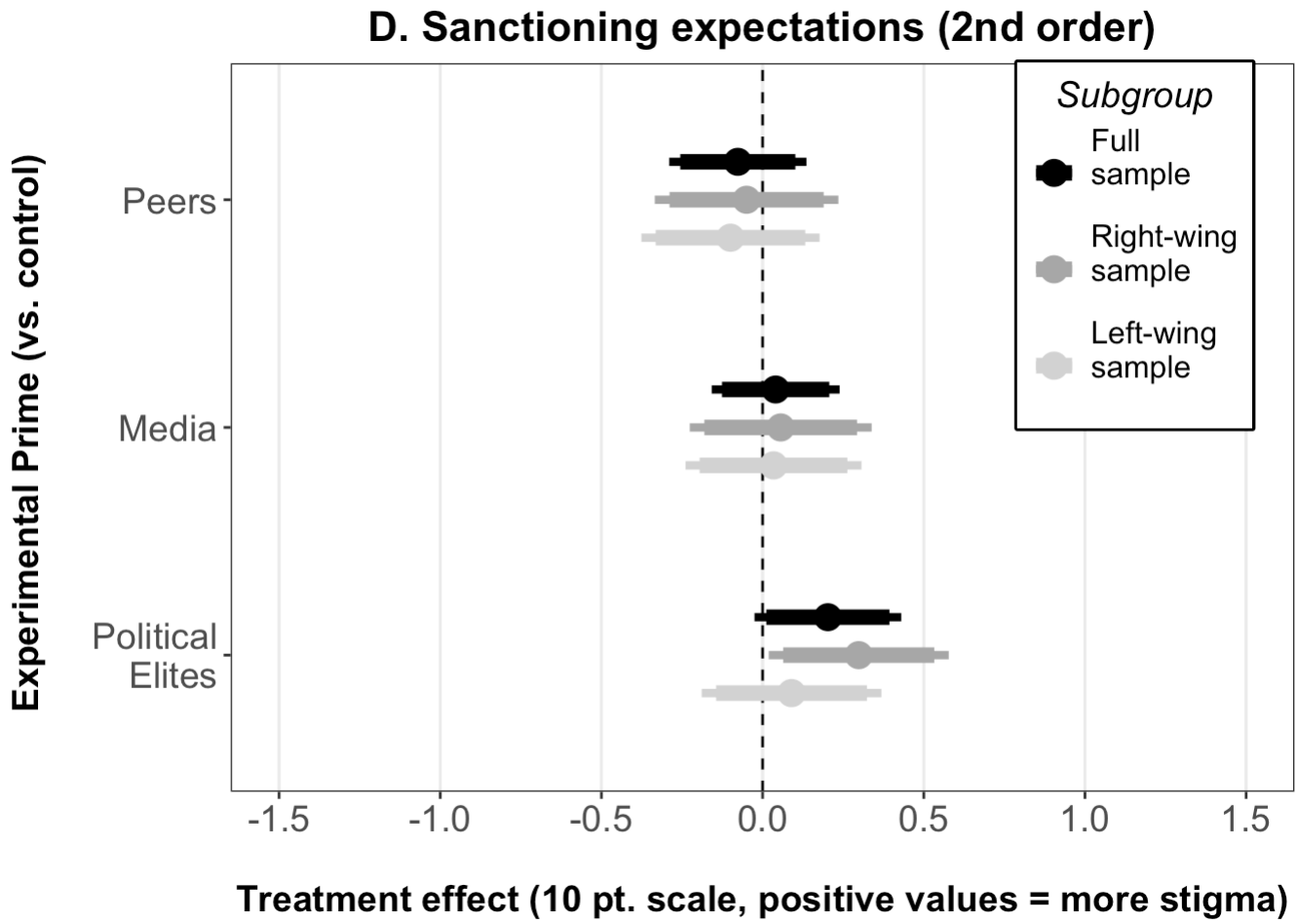

```
get_plot(reg_coefs(ensured, outcome = "first.sanction.index", controls = FALSE),
         mytitle = "C. Willingness to sanction (1st order)", range = 1.5)
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



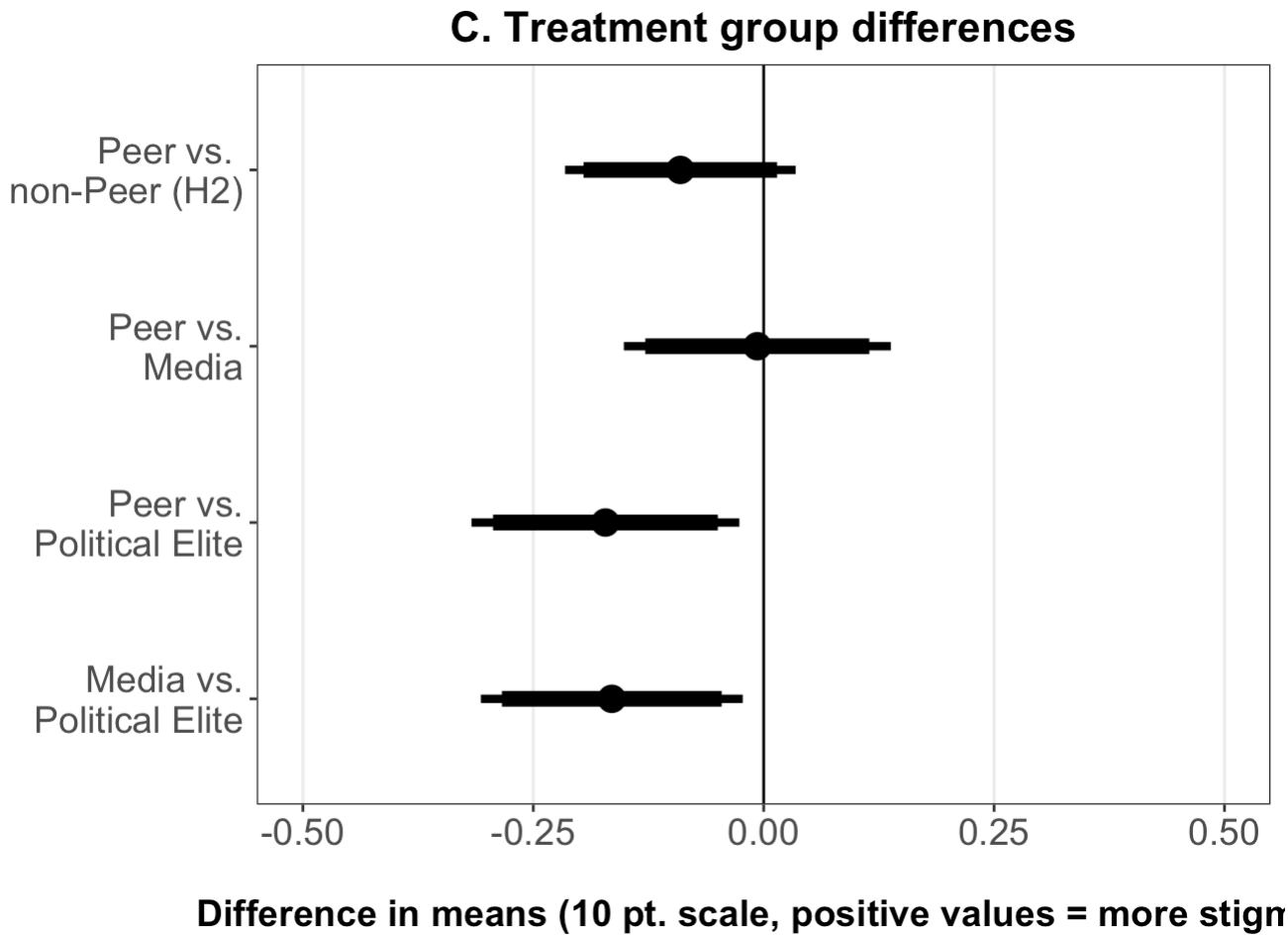

```
get_plot(reg_coefs(ensured, outcome = "second.sanction.index", controls = FALSE),
         mytitle = "D. Sanctioning expectations (2nd order)", range = 1.5)
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```




```
# FIGURE A4
get_paired(ensured, controls = FALSE) + labs(title = "C. Treatment group differences")
  theme(plot.title = element_text(size = 16, face = 'bold', hjust = 0.5))
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



A.5 Descriptive Data

```
# Reshape dataset to stack variables for distribution
outcome_dist <- data.frame(
  var = c(rep("Index", nrow(sbt(data, data$treat == 0))),
          rep("1st order
un-acceptability", nrow(sbt(data, data$treat == 0))),
          rep("2nd order
un-acceptability", nrow(sbt(data, data$treat == 0))),
          rep("1st order
willingness to
sanctioning", nrow(sbt(data, data$treat == 0))),
          rep("2nd order
willingness to
sanctioning", nrow(sbt(data, data$treat == 0)))),
  responses = c(sbt(data, data$treat == 0)$gen.stigma, sbt(data, data$treat == 0)$firs
                sbt(data, data$treat == 0)$first.sanction.index, sbt(data, data$treat
```

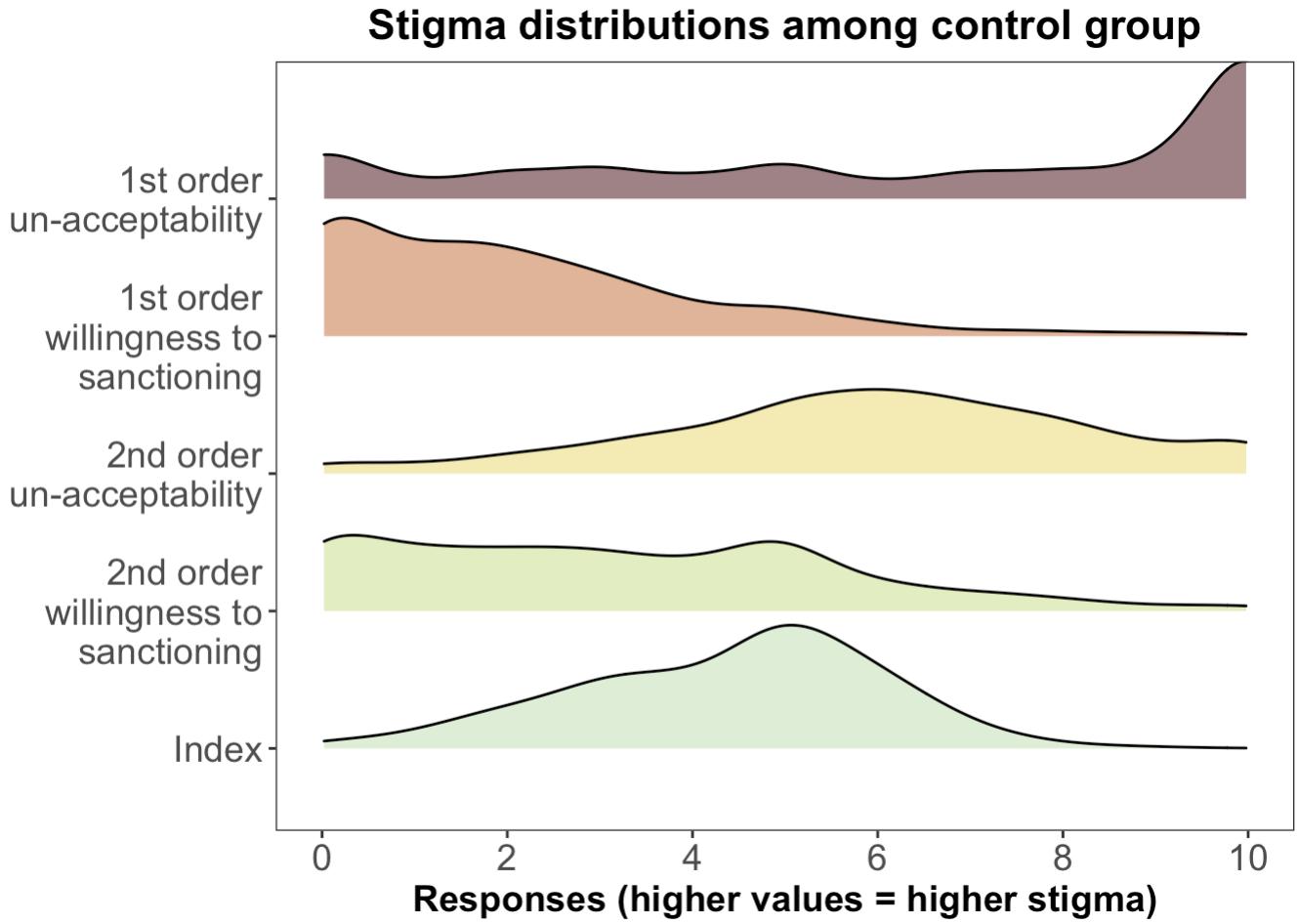
```
Warning in as.data.frame.numeric(x[[i]], optional = TRUE, nm = "x[[i]]", :
Direct call of 'as.data.frame.numeric()' is deprecated.  Use
'as.data.frame.vector()' or 'as.data.frame()' instead
```

```
# FIGURE A5
ggplot(data = outcome_dist) +
  geom_density_ridges(aes(x = responses, y = var, fill = var), alpha = 0.6, scale = 1)
```

```
  scale_fill_manual(values = c("#561D25", "#CE8147", "#ECDD7B", "#D3E298", "#CDE7BE"))
  scale_x_continuous(limits = c(0, 10), breaks = seq(0, 10, 2)) +
  labs( fill = "Outcome", x = "Responses (higher values = higher stigma)",
        title = "Stigma distributions among control group") +
  theme_bw() +
  theme(axis.title.x = element_text(size = 14, face = 'bold'), axis.text = element_tex
        axis.title.y = element_blank(), plot.title = element_text(size = 16, face = 'b
        panel.grid = element_blank(), legend.position = "none") +
  scale_y_discrete(limits = rev)
```

```
Picking joint bandwidth of 0.513
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_density_ridges()`).
```




```
# TABLE A3 – Means and Standard Deviations
paste("Generalized Index",
      round(mean(sbt(data, treat == 0)$gen.stigma, na.rm = TRUE),2), round(sd(sbt(data
      round(mean(data$gen.stigma, na.rm = TRUE),2), round(sd(data$gen.stigma, na.rm =
```

```
[1] "Generalized Index & 4.39 & 1.65 & 4.31 & 1.61"
```

```
paste("1st order (un)acceptability",
      round(mean(sbt(data, treat == 0)$first.accept, na.rm = TRUE),2), round(sd(sbt(da
      round(mean(data$first.accept, na.rm = TRUE),2), round(sd(data$first.accept, na.r
```

```
[1] "1st order (un)acceptability & 6.36 & 3.68 & 6.1 & 3.64"
```

```
paste("1st order sanctioning",
      round(mean(sbt(data, treat == 0)$first.sanction.index, na.rm = TRUE),2), round(s
      round(mean(data$first.sanction.index, na.rm = TRUE),2), round(sd(data$first.sanc
```

```
[1] "1st order sanctioning & 2.08 & 2.02 & 2.06 & 2.02"
```

```
paste("2nd order (un)acceptability",
      round(mean(sbt(data, treat == 0)$second.accept, na.rm = TRUE),2), round(sd(sbt(d
      round(mean(data$second.accept, na.rm = TRUE),2), round(sd(data$second.accept, na
```

```
[1] "2nd order (un)acceptability & 5.91 & 2.35 & 5.93 & 2.22"
```

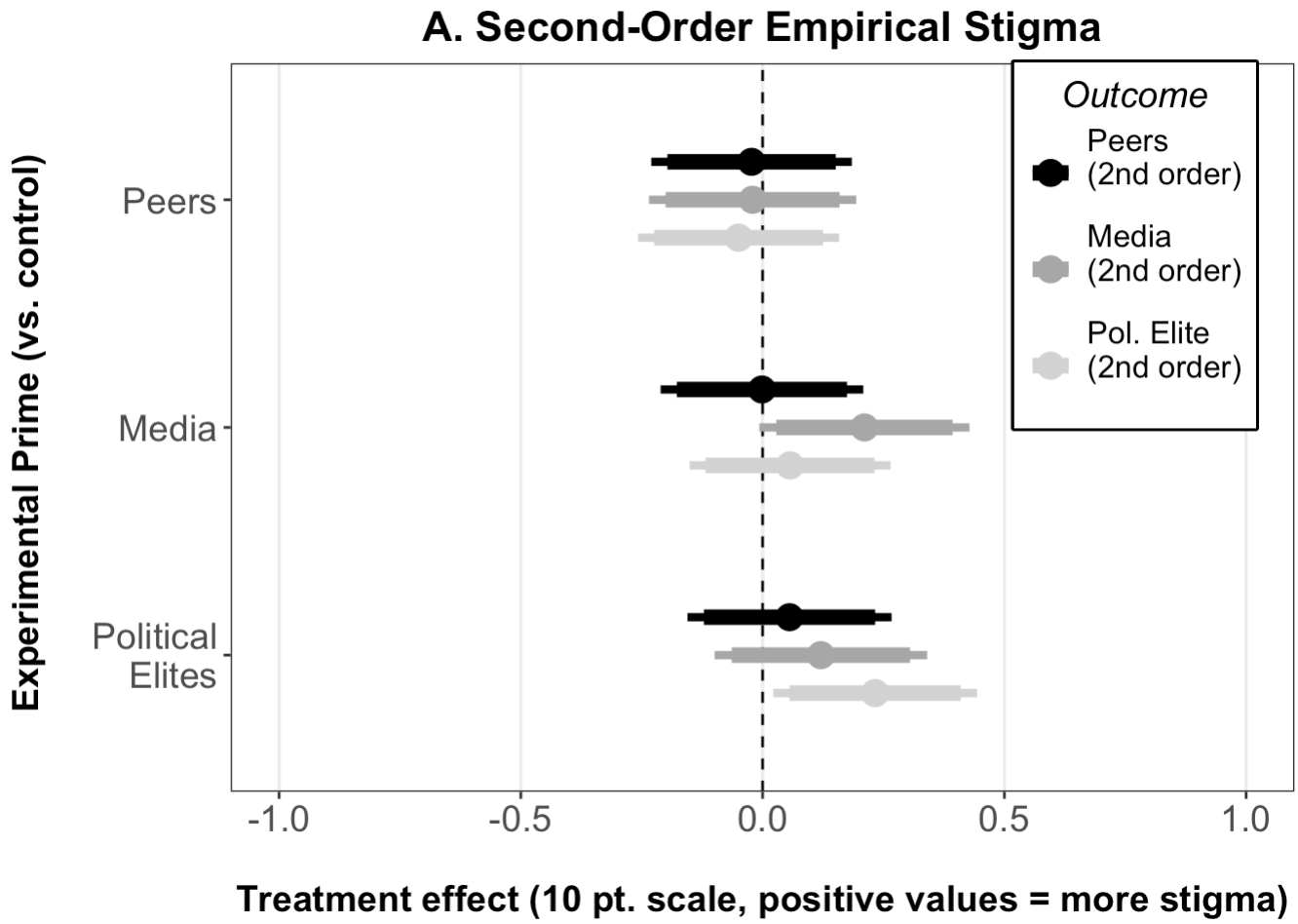
```
paste("2nd order sanctioning",
      round(mean(sbt(data, treat == 0)$second.sanction.index, na.rm = TRUE),2), round(
      round(mean(data$second.sanction.index, na.rm = TRUE),2), round(sd(data$second.sa
```

```
[1] "2nd order sanctioning & 3.18 & 2.41 & 3.16 & 2.38"
```

A.6 Including Models with Controls

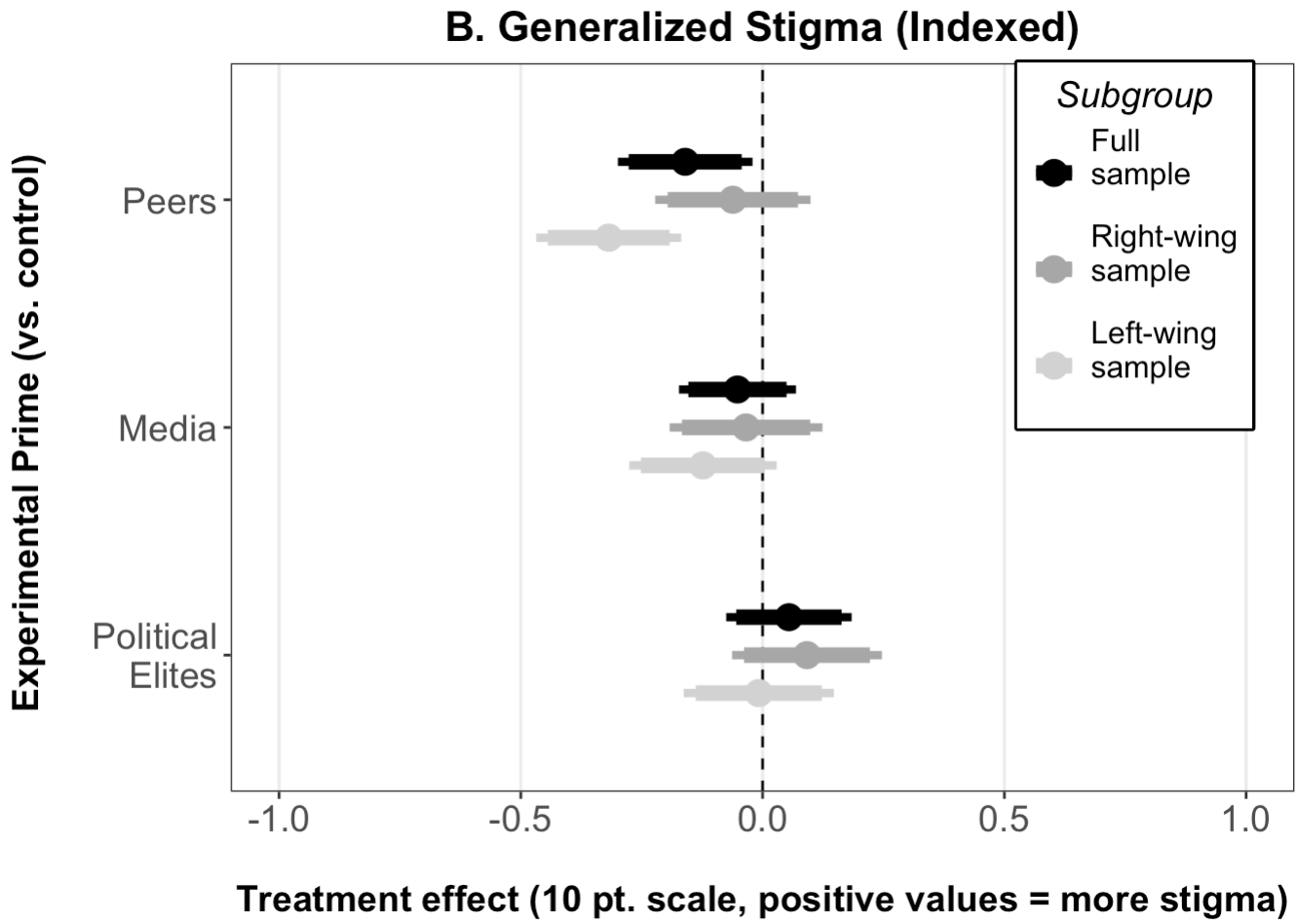
```
# FIGURE A6
get_plot(validation_coefs(data, controls = TRUE), mytitle = "A. Second-Order Empirical
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



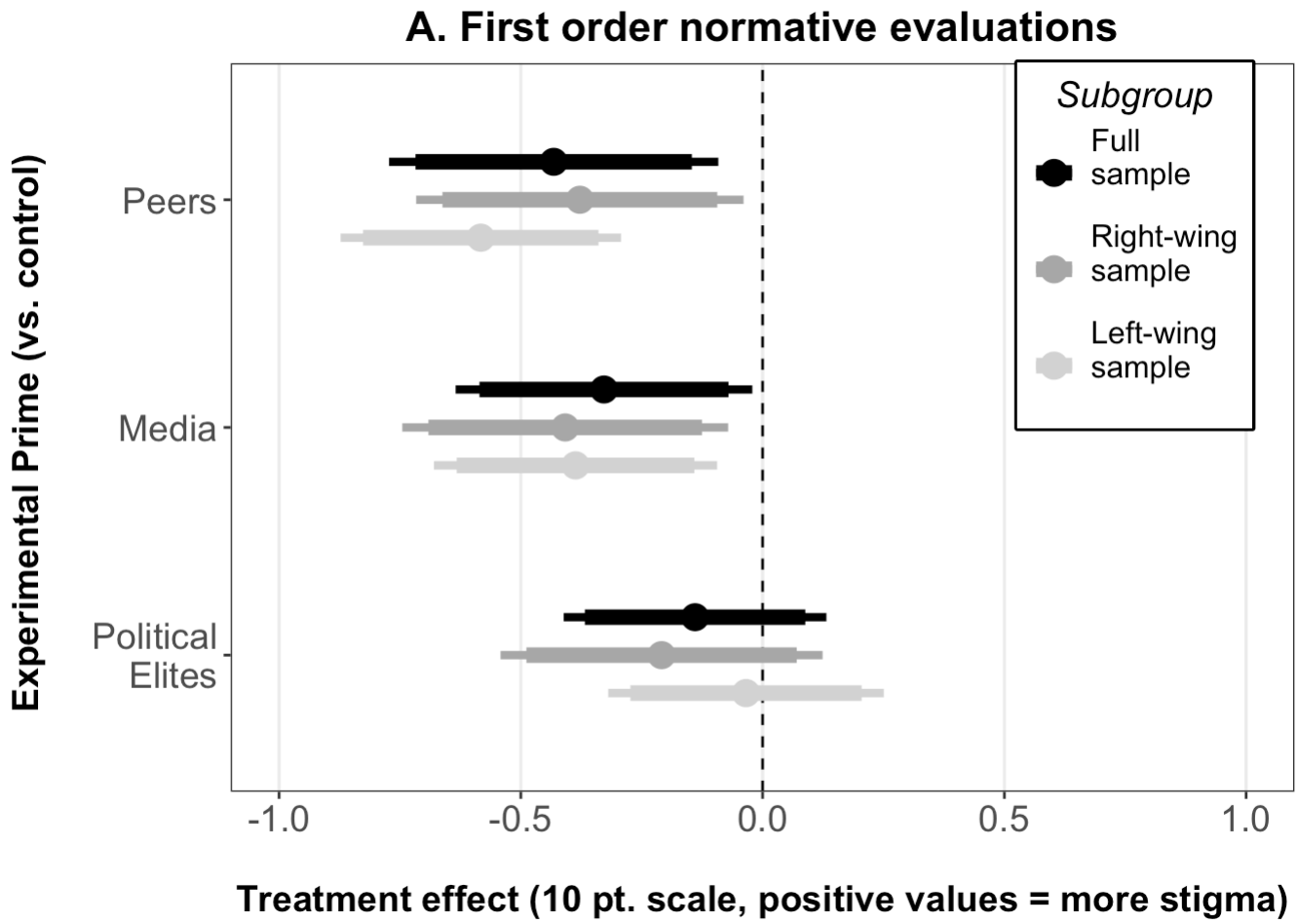

```
get_plot(reg_coefs(data, controls = TRUE), mytitle = "B. Generalized Stigma (Indexed)"
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



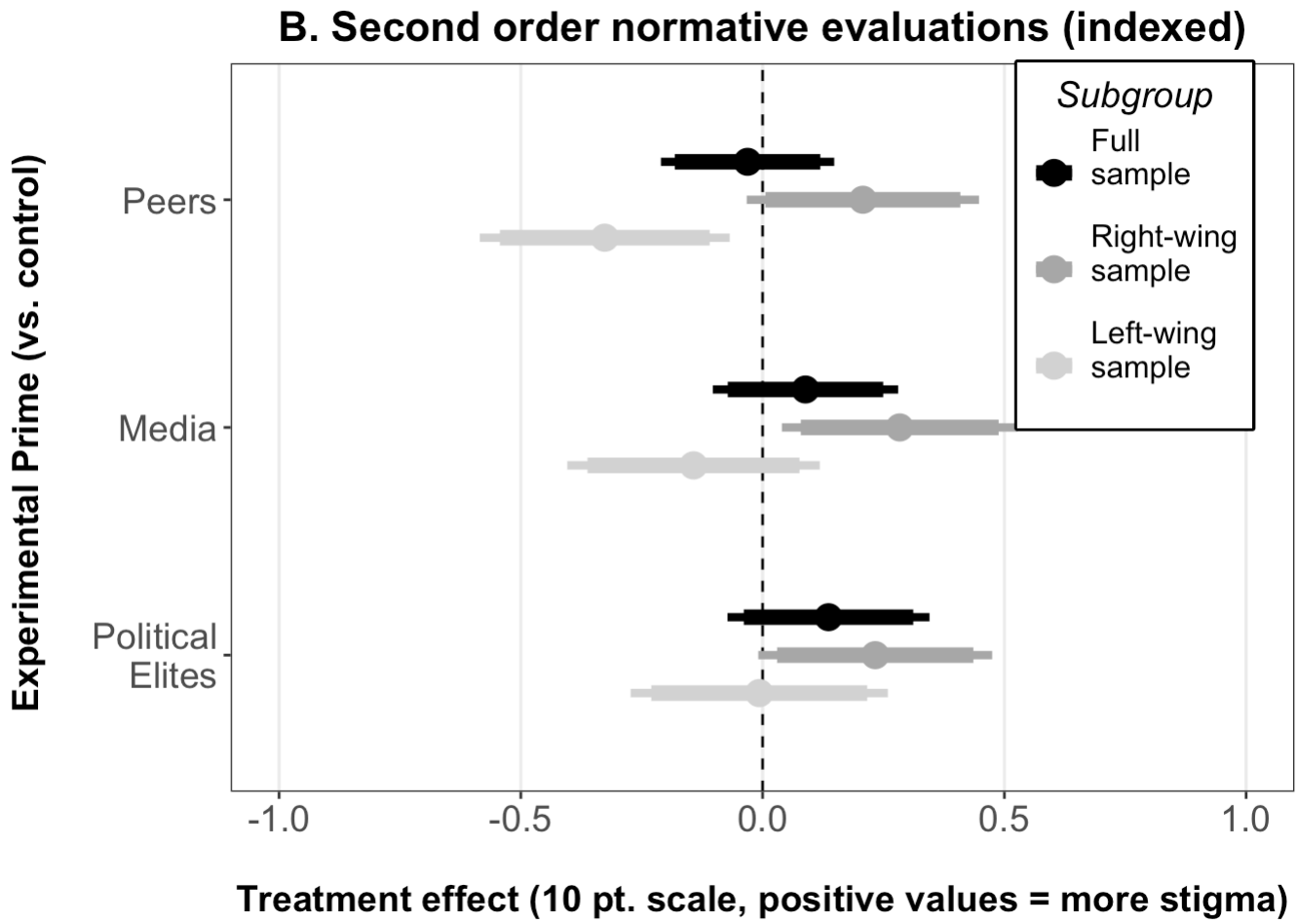

```
# FIGURE A7
get_plot(reg_coefs(data, outcome = "first.accept", controls = TRUE),
         mytitle = "A. First order normative evaluations")
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



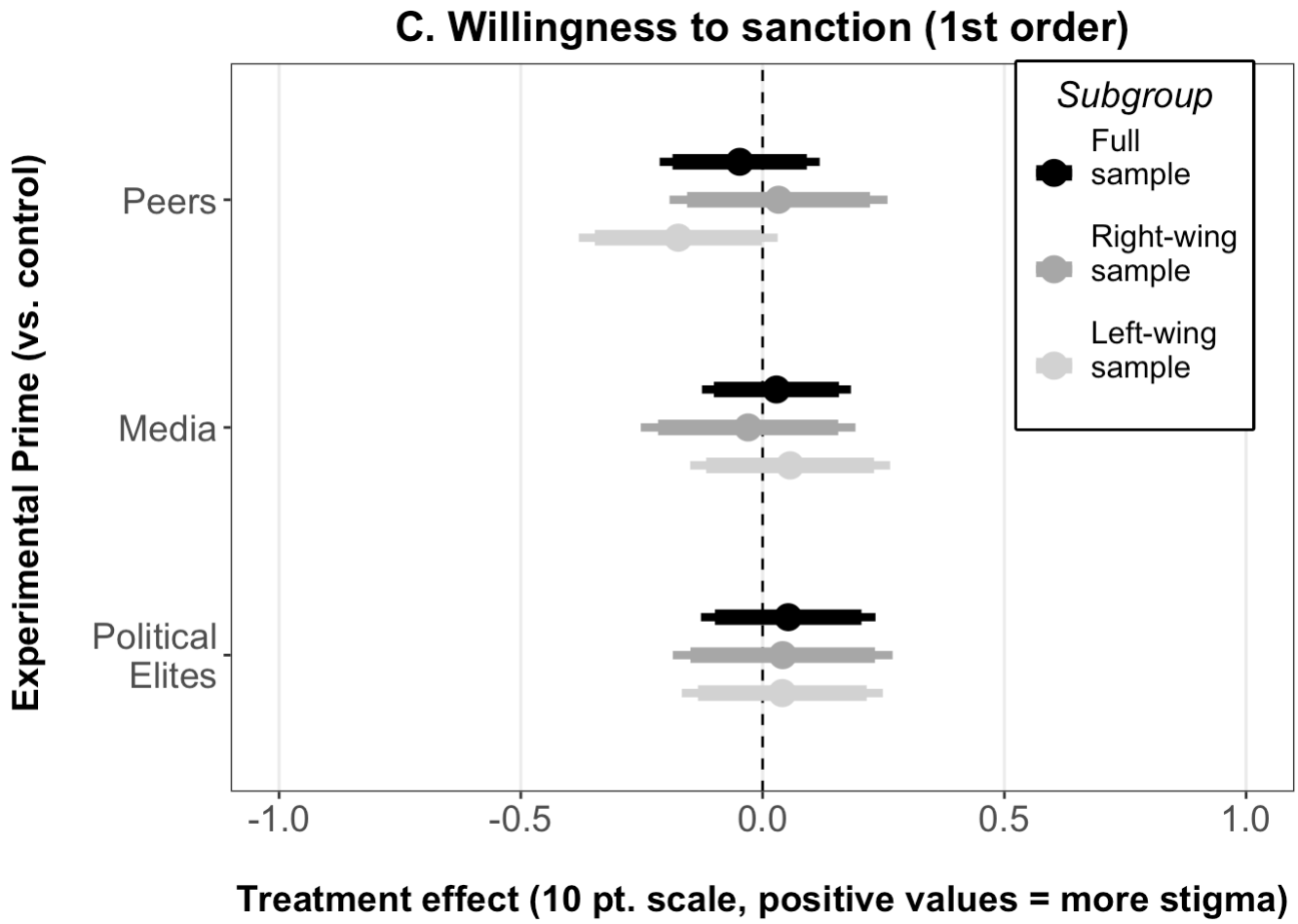

```
get_plot(reg_coefs(data, outcome = "second.accept", controls = TRUE),
         mytitle = "B. Second order normative evaluations (indexed)")
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



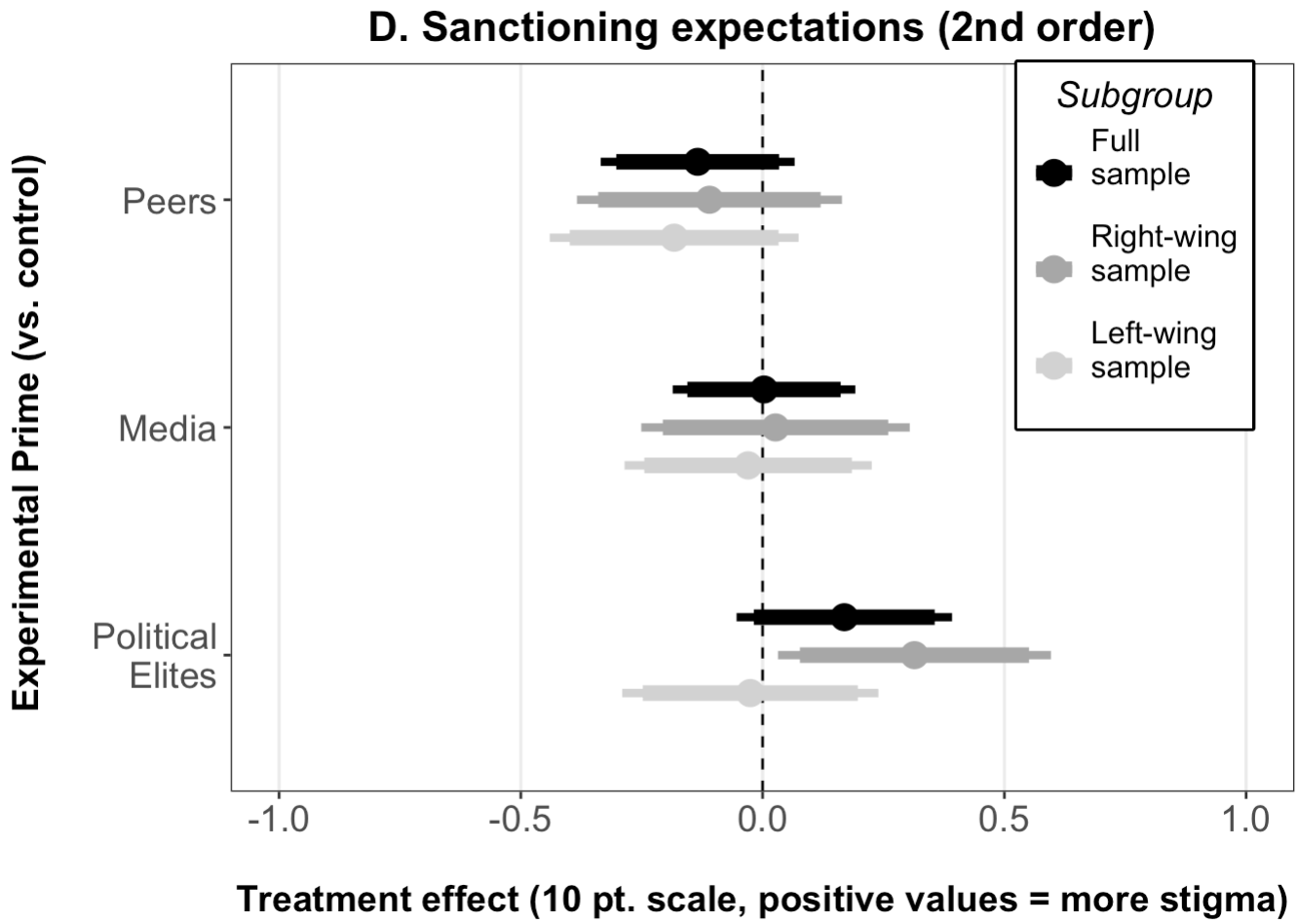

```
get_plot(reg_coefs(data, outcome = "first.sanction.index", controls = TRUE),
         mytitle = "C. Willingness to sanction (1st order)")
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



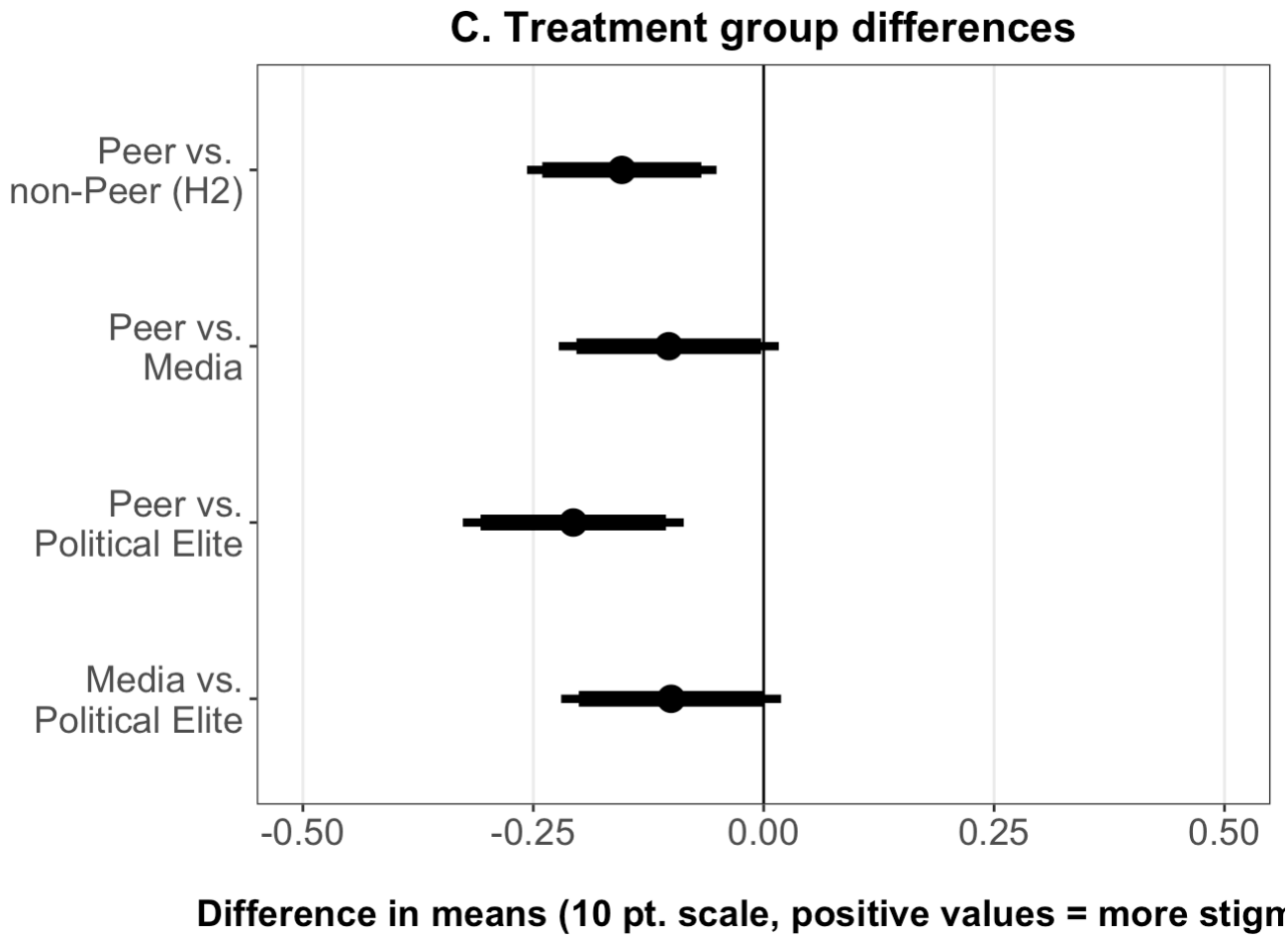

```
get_plot(reg_coefs(data, outcome = "second.sanction.index", controls = TRUE),
         mytitle = "D. Sanctioning expectations (2nd order)")
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```




```
# FIGURE A8
get_paired(data, controls = TRUE) + labs(title = "C. Treatment group differences") +
  theme(plot.title = element_text(size = 16, face = 'bold', hjust = 0.5))
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```



A.7 Checking Index Validity

A.7.1 Correlation Matrix - TABLE A4

```
round(cor(data[data$treat == 0, c("first.accept", "second.accept", "first.sanction.ind
                                  "second.sanction.index")], use = 'complete.obs'),
```

```
                      first.accept second.accept first.sanction.index
first.accept                  1.00          0.49                 0.23
second.accept                 0.49          1.00                -0.09
first.sanction.index          0.23         -0.09                 1.00
second.sanction.index        -0.01         -0.07                 0.49
                      second.sanction.index
first.accept                          -0.01
second.accept                         -0.07
first.sanction.index                   0.49
second.sanction.index                  1.00
```

A.7.2 PCA

```
# Run PCA commands on indexed outcomes
pca <- data[,c("first.accept", "second.accept", "first.sanction.index", "second.sancti
pca <- as.data.frame(scale(na.omit(pca))) # omit NAs and scale variables
```

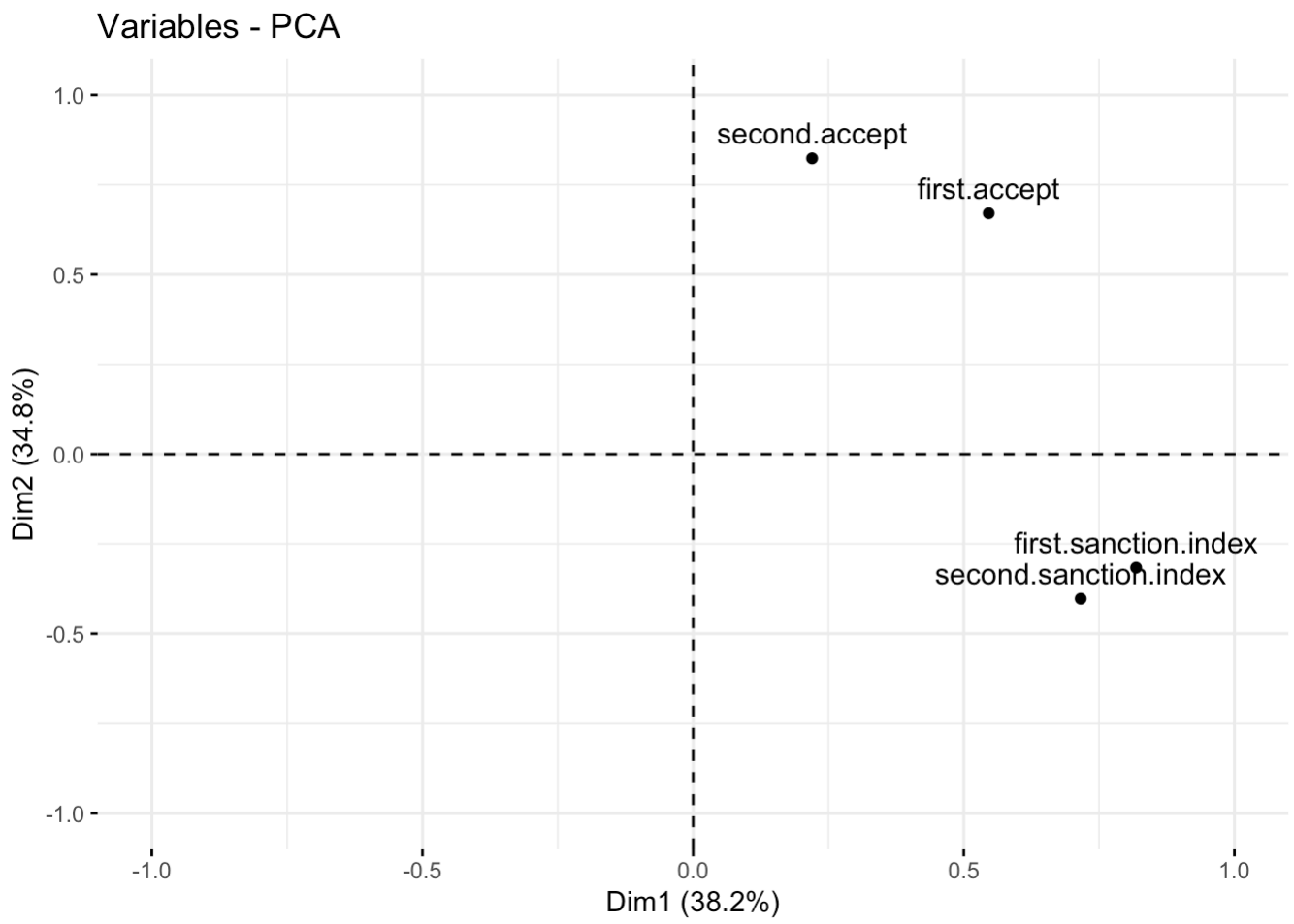
```
data.pca <- princomp(pca)
summary(data.pca) # First 2 components cumulatively explain 73% of variance
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4
Standard deviation     1.2365094 1.1789649 0.8336107 0.62084908
Proportion of Variance 0.3823082 0.3475526 0.1737582 0.09638089
Cumulative Proportion  0.3823082 0.7298609 0.9036191 1.00000000
```

```
data.pca$loadings[, 1:2] # Opposite sign loadings onto comp 2 by acceptability vs. san
```

```
                         Comp.1     Comp.2
first.accept          0.4415902  0.5687692
second.accept         0.1777497  0.6984915
first.sanction.index  0.6618846 -0.2682182
second.sanction.index 0.5790612 -0.3415702
```

```
# FIGURE A9
fviz_pca_var(data.pca, geom = c("point", "text"), xlim = c(-1, 1), ylim = c(-1, 1))
```



A.8 Alternative Specifications for Conditional Effects

```
# Create binary left-right cutoff
data$RL2 <- ifelse(data$ideology > 4, "right", "left")
```

```
# Run regressions to examine significance of interaction terms
m1 <- summary(lm(gen.stigma ~ anytreat + factor(RL2) + anytreat*factor(RL2), data = sb
m2 <- summary(lm(gen.stigma ~ anytreat + factor(RL2) + anytreat*factor(RL2), data = sb
m3 <- summary(lm(gen.stigma ~ anytreat + factor(RL2) + anytreat*factor(RL2), data = sb

# Print TABLE A5
modelsummary(list(m1, m2, m3), stars = TRUE)
```

	(1)	(2)	(3)
(Intercept)	5.276***	5.276***	5.276***
	(0.058)	(0.057)	(0.057)
anytreat	-0.292***	-0.198*	-0.046
	(0.082)	(0.081)	(0.081)
factor(RL2)right	-1.626***	-1.626***	-1.626***
	(0.078)	(0.077)	(0.077)
anytreat × factor(RL2)right	0.225*	0.186+	0.121
	(0.111)	(0.109)	(0.109)
Num.Obs.	2768	2761	2745
R2	0.217	0.224	0.232
R2 Adj.	0.216	0.223	0.231
RMSE	1.44	1.43	1.42

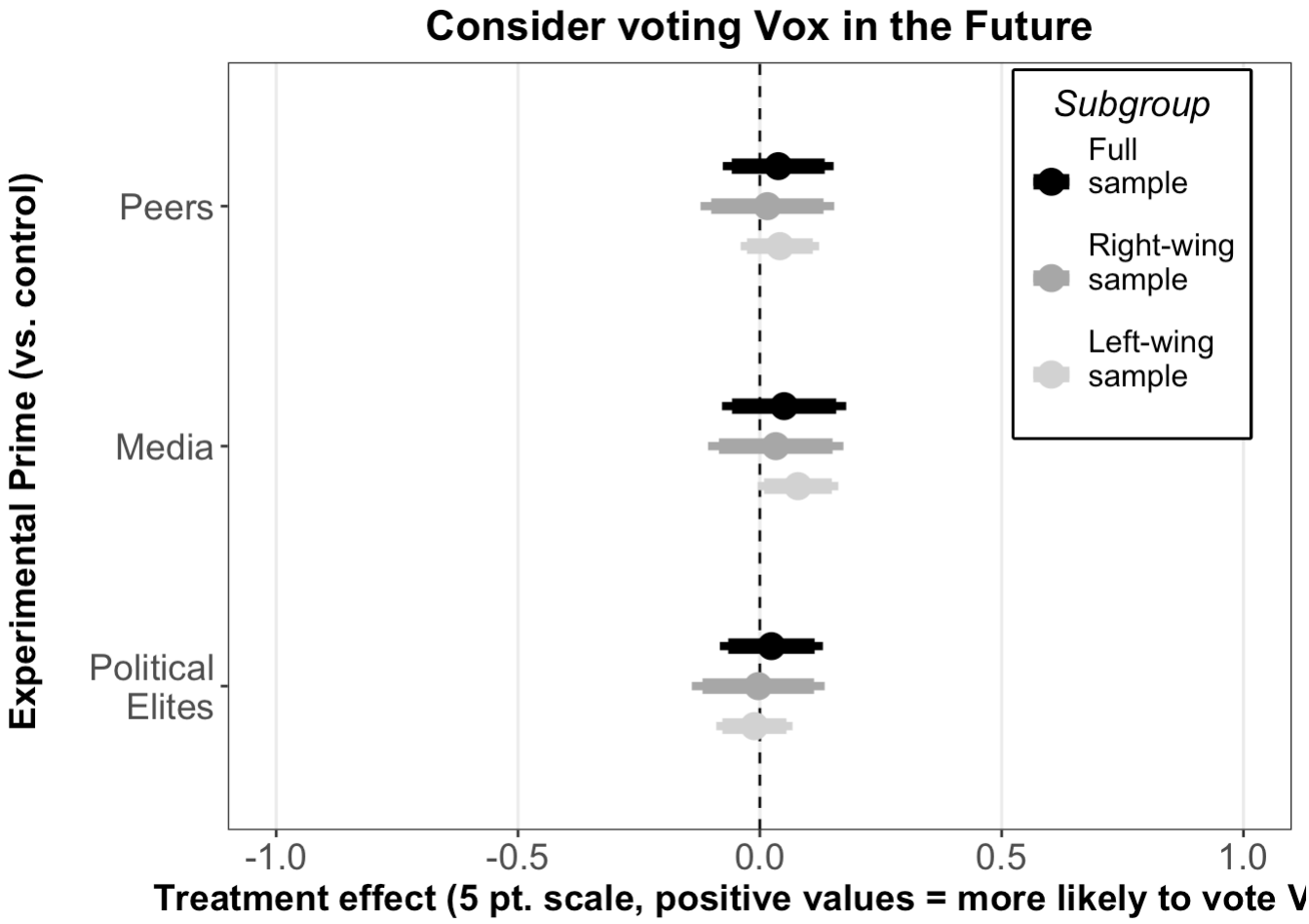
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

```
# Create 3-level ideology cutoff
data$RL3 <- ifelse(data$ideology > 6, "right", ifelse(data$ideology > 4, "center", "le

# Run regressions to examine significance of interaction terms
m1 <- summary(lm(gen.stigma ~ anytreat + factor(RL3) + anytreat*factor(RL3), data = sb
m2 <- summary(lm(gen.stigma ~ anytreat + factor(RL3) + anytreat*factor(RL3), data = sb
m3 <- summary(lm(gen.stigma ~ anytreat + factor(RL3) + anytreat*factor(RL3), data = sb

m4 <- summary(lm(gen.stigma ~ anytreat + ideology + anytreat*ideology, data = sbt(data
m5 <- summary(lm(gen.stigma ~ anytreat + ideology + anytreat*ideology, data = sbt(data
m6 <- summary(lm(gen.stigma ~ anytreat + ideology + anytreat*ideology, data = sbt(data

# Print TABLE A6
modelsummary(list(m1, m4, m2, m5, m3, m6), stars = TRUE)
```

	(1)	(2)	(3)	(4)	(5)	(6)
(Intercept)	4.107***	5.858***	4.107***	5.858***	4.107***	5.858***
	(0.074)	(0.074)	(0.073)	(0.073)	(0.072)	(0.073)
anytreat	-0.120	-0.264*	-0.071	-0.176+	0.057	-0.023
	(0.104)	(0.107)	(0.101)	(0.105)	(0.102)	(0.104)
factor(RL3)left	1.169***		1.169***		1.169***	
	(0.093)		(0.092)		(0.091)	
factor(RL3)right	-0.878***		-0.878***		-0.878***	
	(0.103)		(0.101)		(0.100)	
anytreat × factor(RL3)left	-0.172		-0.128		-0.103	
	(0.131)		(0.128)		(0.129)	
anytreat × factor(RL3)right	0.094		0.022		0.033	
	(0.145)		(0.143)		(0.141)	
ideology		-0.308***		-0.308***		-0.308***
		(0.013)		(0.013)		(0.013)
anytreat × ideology		0.019		0.013		0.007
		(0.019)		(0.019)		(0.019)
Num.Obs.	2768	2768	2761	2761	2745	2745
R2	0.253	0.259	0.263	0.265	0.272	0.281
R2 Adj.	0.251	0.259	0.262	0.265	0.270	0.280
RMSE	1.41	1.40	1.39	1.39	1.38	1.37

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

A.9 Effects on Support for Vox

```
# FIGURE A10
get_plot(reg_coefs(data, outcome = "vox.support"), mytitle = "Consider voting Vox in t
  labs(x = "Treatment effect (5 pt. scale, positive values = more likely to vote Vox)"
```

```
Warning: `position_dodge()` requires non-overlapping x intervals.
`position_dodge()` requires non-overlapping x intervals.
```