Original Replication File

Overview

This document provides the code necessary to replicate the results of “Getting the Message? Choice, Self-Selection, and the Efficacy of Social Movement Arguments” It consist of the following sections

- **Setup** Sets up R environment:
  - Sets working directory and `knitr` options for display
  - Loads libraries (`tidyverse` packages, `car`, `Hmisc`, `kableExtra`)
  - Loads data (`df_mtg.rda`, `df_qg.rda`, `power_simulations.rda`)
- **Functions** Defines a set of custom functions to:
  - Calculate treatment effects (`diff_fn()`, `acte_fn()`, `cacte_fn()`)
  - Display treatment effects (`balance_fn()`, `plot_balance_fn()`, `effects_fn()`, `plot_effects_fn()`, `format_ci_fn()`, `table_fn()`,`table_app_fn()`)
  - Conduct power simulations (`data_fn()`, `power_fn()`, `sim_power_fn()`, `display_power_fn()`)
- **Main Figures** Produces Figures 1-6 as seen in text using functions defined above
- **Main Table** Produces Tables 1-2 as seen in text using functions defined above
- **Online Appendix** Produces tables and figures from Online Appendices C-F using functions defined above

Note: Each power simulations displayed in Figure 2 and Appendix C takes approximately 30-40 minutes to complete. The replication file loads the cached results of a round of power simulations. To conduct simulations, uncomment code.

Setup

```
# Set working directory
wd <- "C:/Users/hanna/Documents/GitHub/ps231b_reproduction_group4/original
reproduction package"
setwd(wd)

# Load libraries

# Uncomment to install packages
# if(!require('tidyverse')){install.packages('tidyverse')}
# if(!require('car')){install.packages('car')}
# if(!require('Hmisc')){install.packages('Hmisc')}
# if(!require('kableExtra')){install.packages('kableExtra')}
# if(!require('sessioninfo')){install.packages('sessioninfo')}
```

```r
library(tidyverse)
library(car)
library(Hmisc)
library(kableExtra)
library(sessioninfo)
library(formatR)

# Set knitr output options
knitr::opts_chunk$set(message = F, warning = F, fig.height = 6, cache = T)
options(knitr.table.format = "latex")

# Load data
load("df_mtg.rda")
load("df_qg.rda")

# Load results of power simulations
load("power_simulations.rda")


# Display session info
sessioninfo::session_info()

## - Session info -----------------------------------------------------------
----
##  setting  value
##  version  R version 4.1.3 (2022-03-10)
##  os       Windows 10 x64 (build 19042)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/Los_Angeles
##  date     2022-04-09
##  pandoc   2.11.4 @ C:/Program Files/RStudio/bin/pandoc/ (via rmarkdown)
##
## - Packages ---------------------------------------------------------------
----
##  package      * version date (UTC) lib source
##  abind          1.4-5   2016-07-21 [1] CRAN (R 4.1.1)
##  assertthat     0.2.1   2019-03-21 [1] CRAN (R 4.1.1)
##  backports      1.4.1   2021-12-13 [1] CRAN (R 4.1.2)
##  base64enc      0.1-3   2015-07-28 [1] CRAN (R 4.1.0)
##  broom          0.7.12  2022-01-28 [1] CRAN (R 4.1.3)
##  car          * 3.0-12  2021-11-06 [1] CRAN (R 4.1.2)
##  carData      * 3.0-5   2022-01-06 [1] CRAN (R 4.1.2)
##  cellranger     1.1.0   2016-07-27 [1] CRAN (R 4.1.1)
##  checkmate      2.0.0   2020-02-06 [1] CRAN (R 4.1.2)
```

```
##  cli            3.1.0   2021-10-27 [1] CRAN (R 4.1.2)
##  cluster        2.1.3   2022-03-28 [1] CRAN (R 4.1.3)
##  colorspace     2.0-3   2022-02-21 [1] CRAN (R 4.1.3)
##  crayon         1.5.1   2022-03-26 [1] CRAN (R 4.1.3)
##  data.table     1.14.2  2021-09-27 [1] CRAN (R 4.1.2)
##  DBI            1.1.2   2021-12-20 [1] CRAN (R 4.1.3)
##  dbplyr         2.1.1   2021-04-06 [1] CRAN (R 4.1.1)
##  digest         0.6.29  2021-12-01 [1] CRAN (R 4.1.3)
##  dplyr        * 1.0.8   2022-02-08 [1] CRAN (R 4.1.3)
##  ellipsis       0.3.2   2021-04-29 [1] CRAN (R 4.1.1)
##  evaluate       0.15    2022-02-18 [1] CRAN (R 4.1.3)
##  fansi          1.0.3   2022-03-24 [1] CRAN (R 4.1.3)
##  fastmap        1.1.0   2021-01-25 [1] CRAN (R 4.1.1)
##  forcats      * 0.5.1   2021-01-27 [1] CRAN (R 4.1.1)
##  foreign        0.8-82  2022-01-13 [1] CRAN (R 4.1.2)
##  formatR      * 1.12    2022-03-31 [1] CRAN (R 4.1.3)
##  Formula      * 1.2-4   2020-10-16 [1] CRAN (R 4.1.1)
##  fs             1.5.2   2021-12-08 [1] CRAN (R 4.1.3)
##  generics       0.1.2   2022-01-31 [1] CRAN (R 4.1.3)
##  ggplot2      * 3.3.5   2021-06-25 [1] CRAN (R 4.1.1)
##  glue           1.6.2   2022-02-24 [1] CRAN (R 4.1.3)
##  gridExtra      2.3     2017-09-09 [1] CRAN (R 4.1.2)
##  gtable         0.3.0   2019-03-25 [1] CRAN (R 4.1.1)
##  haven          2.4.3   2021-08-04 [1] CRAN (R 4.1.1)
##  Hmisc        * 4.6-0   2021-10-07 [1] CRAN (R 4.1.2)
##  hms            1.1.1   2021-09-26 [1] CRAN (R 4.1.2)
##  htmlTable      2.4.0   2022-01-04 [1] CRAN (R 4.1.2)
##  htmltools      0.5.2   2021-08-25 [1] CRAN (R 4.1.1)
##  htmlwidgets    1.5.4   2021-09-08 [1] CRAN (R 4.1.1)
##  httr           1.4.2   2020-07-20 [1] CRAN (R 4.1.1)
##  jpeg           0.1-9   2021-07-24 [1] CRAN (R 4.1.1)
##  jsonlite       1.8.0   2022-02-22 [1] CRAN (R 4.1.3)
##  kableExtra   * 1.3.4   2021-02-20 [1] CRAN (R 4.1.1)
##  knitr          1.38    2022-03-25 [1] CRAN (R 4.1.3)
##  lattice      * 0.20-45 2021-09-22 [1] CRAN (R 4.1.2)
##  latticeExtra   0.6-29  2019-12-19 [1] CRAN (R 4.1.2)
##  lifecycle      1.0.1   2021-09-24 [1] CRAN (R 4.1.2)
##  lubridate      1.8.0   2021-10-07 [1] CRAN (R 4.1.2)
##  magrittr       2.0.3   2022-03-30 [1] CRAN (R 4.1.3)
##  Matrix         1.4-1   2022-03-23 [1] CRAN (R 4.1.3)
##  modelr         0.1.8   2020-05-19 [1] CRAN (R 4.1.1)
##  munsell        0.5.0   2018-06-12 [1] CRAN (R 4.1.1)
##  nnet           7.3-17  2022-01-13 [1] CRAN (R 4.1.3)
##  pillar         1.7.0   2022-02-01 [1] CRAN (R 4.1.3)
##  pkgconfig      2.0.3   2019-09-22 [1] CRAN (R 4.1.1)
##  png            0.1-7   2013-12-03 [1] CRAN (R 4.1.1)
##  purrr        * 0.3.4   2020-04-17 [1] CRAN (R 4.1.1)
##  R6             2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
##  RColorBrewer   1.1-3   2022-04-03 [1] CRAN (R 4.1.3)
##  readr        * 2.1.2   2022-01-30 [1] CRAN (R 4.1.3)
```

```
##  readxl         1.4.0   2022-03-28 [1] CRAN (R 4.1.3)
##  reprex         2.0.1   2021-08-05 [1] CRAN (R 4.1.1)
##  rlang          1.0.2   2022-03-04 [1] CRAN (R 4.1.3)
##  rmarkdown      2.13    2022-03-10 [1] CRAN (R 4.1.3)
##  rpart          4.1.16  2022-01-24 [1] CRAN (R 4.1.3)
##  rstudioapi     0.13    2020-11-12 [1] CRAN (R 4.1.1)
##  rvest          1.0.2   2021-10-16 [1] CRAN (R 4.1.2)
##  scales         1.1.1   2020-05-11 [1] CRAN (R 4.1.1)
##  sessioninfo  * 1.2.2   2021-12-06 [1] CRAN (R 4.1.2)
##  stringi        1.7.6   2021-11-29 [1] CRAN (R 4.1.2)
##  stringr      * 1.4.0   2019-02-10 [1] CRAN (R 4.1.1)
##  survival     * 3.3-1   2022-03-03 [1] CRAN (R 4.1.3)
##  svglite        2.1.0   2022-02-03 [1] CRAN (R 4.1.3)
##  systemfonts    1.0.4   2022-02-11 [1] CRAN (R 4.1.3)
##  tibble       * 3.1.6   2021-11-07 [1] CRAN (R 4.1.3)
##  tidyr        * 1.2.0   2022-02-01 [1] CRAN (R 4.1.3)
##  tidyselect     1.1.2   2022-02-21 [1] CRAN (R 4.1.3)
##  tidyverse    * 1.3.1   2021-04-15 [1] CRAN (R 4.1.1)
##  tzdb           0.3.0   2022-03-28 [1] CRAN (R 4.1.3)
##  utf8           1.2.2   2021-07-24 [1] CRAN (R 4.1.1)
##  vctrs          0.4.0   2022-03-30 [1] CRAN (R 4.1.3)
##  viridisLite    0.4.0   2021-04-13 [1] CRAN (R 4.1.1)
##  webshot        0.5.2   2019-11-22 [1] CRAN (R 4.1.1)
##  withr          2.5.0   2022-03-03 [1] CRAN (R 4.1.3)
##  xfun           0.30    2022-03-02 [1] CRAN (R 4.1.3)
##  xml2           1.3.3   2021-11-30 [1] CRAN (R 4.1.2)
##  yaml           2.3.5   2022-02-21 [1] CRAN (R 4.1.2)
##
##  [1] C:/Users/hanna/Documents/R/win-library/4.1
##  [2] C:/Program Files/R/R-4.1.3/library
##
## --------------------------------------------------------------------------
----
```

Functions

Functions to Calculate Treatment effects

- `diff_fn()`: Estimate differences in means

```
# Difference in Means Function
diff_fn <- function(the_data, dv1="Y",c,weights=F,...){
  # REQUIRES
    require(Hmisc)
  # INPUTS:
    # the_data: data frame
    # dv1: outcome
    # c: object containing names of treatment conditions
    # weights: boolean indicating whether to calculated weighted ATE
  # OUTPUTS:
```

```
    # result: vector containing Difference in Means, SE, 95% ci, 90%ci, and 
p-value

  tmp <- as.data.frame(the_data[the_data$treatment%in%c, ])

  if(weights==F){
    mu1 <- with(tmp, mean(tmp[treatment==c[1], dv1],na.rm=T))
    mu2 <- with(tmp, mean(tmp[treatment==c[2], dv1],na.rm=T))
    sd1 <- with(tmp, sd(tmp[treatment==c[1], dv1],na.rm=T))
    sd2 <- with(tmp, sd(tmp[treatment==c[2], dv1],na.rm=T))
  }
  if(weights==T){
    mu1 <- with(tmp, Hmisc::wtd.mean(
      tmp[treatment==c[1],
dv1],na.rm=T,weights=tmp[treatment==c[1],"weights"])
      )
    mu2 <- with(tmp, Hmisc::wtd.mean(
      tmp[treatment==c[2],
dv1],na.rm=T,weights=tmp[treatment==c[2],"weights"])
      )
    sd1 <- sqrt(with(tmp, Hmisc::wtd.var(
      tmp[treatment==c[1], dv1],na.rm=T,weights =
tmp[treatment==c[1],"weights"])
      ))
    sd2 <- sqrt(with(tmp, Hmisc::wtd.var(
      tmp[treatment==c[2], dv1],na.rm=T,weights =
tmp[treatment==c[2],"weights"])
      ))
  }
  # Calculate Difference
  diff <- mu2-mu1

  # Calculate N
  n1 <- with(tmp, sum(!is.na(tmp[treatment==c[1],
dv1])*tmp[treatment==c[1],"weights"]))
  n2 <- with(tmp, sum(!is.na(tmp[treatment==c[2],
dv1])*tmp[treatment==c[2],"weights"]))

   # SE of Difference
  se <- sqrt( sd1^2/n1 + sd2^2/n2)

  # Degrees of Freedom
  the_df <- (sd1^2/n1+sd2^2/n2)^2/((sd1^4)/(n1^2*(n1-1))+ (sd2^4)/(n2^2*(n2-
1)))

  # 95% CI
  ll <- diff - qt(.975,the_df)*se
  ul <- diff + qt(.975,the_df)*se
  # 90% CI
```

```r
  ll90 <- diff - qt(.95,the_df)*se
  ul90 <- diff + qt(.95,the_df)*se
  # t-stat
  stat <- diff/se
  # p-value
  pval = 2 * pt(-abs(stat),the_df)
  # Combine results
  results <- c(Difference = diff, SE = se, ll = ll, ul = ul,
               ll90=ll90,ul90=ul90, pval = pval)
  return(results)
}
```

- acte_fn(): Estimate ACTEs using delta method

```r
# ACTE function
acte_fn <- function(dat,dv2="Y",z,w,...){
  # REQUIRES
    require(car)
    require(Hmisc)

  # INPUTS:
    # dat: data frame
    # dv2: outcome
    # z: object containing treatment groups to calculate ACTE-Select or ACTE-
Avoid
    # w: weight argument passed to diff_fn
  # OUTPUTS:
  # result: vector containing ATE, SE, 95% ci, 90%ci, and p-value

  df <- dat
  N <- dim(df)[1]
  # N assigned to random assignment
  n_exp <- sum(df$C=="Experiment")
  # N assigned to choice
  n_choice <- sum(df$C == "Choice")
  # N avoiding treatment
  n_ch_a <- sum(df$C=="Choice" & df$avoid01==1)
  # N selecting treatment
  n_select <- sum(df$C=="Choice" & df$avoid01==0, na.rm=T)
  # N avoiding treatment
  n_avoid <- sum(df$C=="Choice" & df$avoid01==1,na.rm = T)
  # N avoiding treatment who receive no treatment
  n_control <- sum(df$D_ch == "Control",na.rm = T)

  # Weights to reflect fact
  df$weights <- rep(1,N)
  df$weights[df$C=="Choice" & df$avoid01==0 & df$D_ch == "Control"] <-
1/(n_select/n_choice)
  df$weights[df$C=="Choice" & df$avoid01==1 & df$D_ch == "Control"] <-
1/(n_control/n_avoid)
```

```
  # Calculate ACTE using Delta Method
    #ACTE-Select: c = c_acte_s = c("Control","Selection")
    #ACTE-Avoid:  c = c_acte_a = c("Selection","Treatment")

  c_acte_s <- c("Control","Selection")
  c_acte_a <- c("Selection","Treatment")

  tmp <- diff_fn(the_data=df, dv1=dv2,c=z, weights=w)
  x <- as.numeric(tmp["Difference"])
  se_x <- as.numeric(tmp["SE"])
  # ACTE-Select
  if(z[1]=="Control"){
    y <- summary(lm(select01~1,df[df$C=="Choice",]))$coef[1,1]
    se_y <- summary(lm(select01~1,df[df$C=="Choice",]))$coef[1,2]}
  # ACTE-Avoid
  if(z[2]=="Treatment"){
    y <- summary(lm(avoid01~1,df[df$C=="Choice",]))$coef[1,1]
    se_y <- summary(lm(avoid01~1,df[df$C=="Choice",]))$coef[1,2]}
  mvec <- c(x=x, y= y)
  V <- diag(c(se_x,se_y)^2)
  est <- car::deltaMethod(mvec,"x/y",V,level=.95)
  est90 <- car::deltaMethod(mvec,"x/y",V,level=.90)
  stat <- as.numeric(est[1])/as.numeric(est[2])

  # Return results
  results <- c(Difference = as.numeric(est[1]),
               SE = as.numeric(est[2]),
               ll = as.numeric(est[3]), ul = as.numeric(est[4]),
               ll90 = as.numeric(est90[3]), ul90 = as.numeric(est90[4]),
               pval = 2 * pnorm(-abs(stat)))
  return(results)
}
```

- `cacte_fn()`: Estimate CACTEs

```
# CACTE function
cacte_fn <- function(d, dv3="Y", z, w2=F){
  # INPUTS:
    # d: data frame
    # dv3: outcome
    # z: object containing treatment groups to calculate ACTE-Select or ACTE-
Avoid
    # w2: weight argument passed to diff_fn
  # OUTPUTS:
    # result: Matrix containing Female and Male CACTE, SE, 95% ci, 90%ci, and
p-value

  tmp <- as.data.frame(d[d$avoid01 == 1, ])
```

```
  tmp$treatment <- tmp$D_ch

  # CACTE- Female
  cacte_female <- diff_fn(tmp, dv1=dv3,c=c("Control","Female"),weights=w2)

  # CACTE Male
  cacte_male <- diff_fn(tmp, dv1=dv3, c=c("Control","Male"),weights=w2)
  results <- rbind(cacte_female,cacte_male)
  return(results)

}
```

Functions to display results

- balance_function(): Function to calculate covariate differences in respondents selecting and avoiding treatment

```
balance_fn <- function(the_data, dv1="Y",c,weights=F,...){
  # INPUTS
    # the_data: data frame
    # dv1: Variable to calculate difference in means
    # c: which group to compare
    # weights: Calcultate weighted differences (T/F)
  # OUTPUTS
  # result: summary stats of difference in means

  tmp <- as.data.frame(the_data[the_data$balance%in%c, ])
  if(weights==F){
    mu1 <- with(tmp, mean(tmp[balance==c[1], dv1],na.rm=T))
    mu2 <- with(tmp, mean(tmp[balance==c[2], dv1],na.rm=T))
  }
  if(weights==T){
    mu1 <- with(tmp, Hmisc::wtd.mean(
      tmp[balance==c[1], dv1],na.rm=T,weights=tmp[balance==c[1],"weights"])
      )
    mu2 <- with(tmp, Hmisc::wtd.mean(
      tmp[balance==c[2], dv1],na.rm=T,weights=tmp[balance==c[2],"weights"])
      )
  }
  diff <- mu1-mu2
  if(weights==F){
    sd1 <- with(tmp, sd(tmp[balance==c[1], dv1],na.rm=T))
    sd2 <- with(tmp, sd(tmp[balance==c[2], dv1],na.rm=T))
  }
  if(weights==T){
    sd1 <- sqrt(with(tmp, Hmisc::wtd.var(
      tmp[balance==c[1], dv1],na.rm=T,weights = tmp[balance==c[1],"weights"])
      ))
    sd2 <- sqrt(with(tmp, Hmisc::wtd.var(
      tmp[balance==c[2], dv1],na.rm=T,weights = tmp[balance==c[2],"weights"])
      ))
```

```
  }
  n1 <- with(tmp, sum(!is.na(tmp[balance==c[1],
dv1])*tmp[balance==c[1],"weights"]))
  n2 <- with(tmp, sum(!is.na(tmp[balance==c[2],
dv1])*tmp[balance==c[2],"weights"]))
  se <- sqrt( sd1^2/n1 + sd2^2/n2)
  the_df <- (sd1^2/n1+sd2^2/n2)^2/((sd1^4)/(n1^2*(n1-1))+ (sd2^4)/(n2^2*(n2-
1)))
  ll <- diff - qt(.975,the_df)*se
  ul <- diff + qt(.975,the_df)*se
  ll90 <- diff - qt(.95,the_df)*se
  ul90 <- diff + qt(.95,the_df)*se
  stat <- diff/se
  pval = 2 * pt(-abs(stat),the_df)
  result <- c(Mu1= mu1, Mu2 = mu2, Difference = diff, SE = se, ll = ll, ul =
ul,
            ll90 = ll90, ul90 = ul90, pval = pval, N1=n1,N2=n2)
  return(result)
}
```

- `plot_balance_function()`: Wrapper function to display results of `balance_function()`:

```
plot_balance_fn <- function(d,
                            bal_labs = c("Female", "Non-white", "Education",
                                         "Income", "PID", "Ideology",
                                         "MeToo Familiarity",
                                         "Specific Support", "General
Support"),
                            comparison = c("Select Treatment", "Avoid
Treatment")
                            ){
  # INPUTS
    # d: data frame
    # bal_labs: Covariate labels
    # comparison: which group to compare
  # OUTPUTS
    # fig: ggplot of comparisons


  # Descriptives Differences in Selecting Treatment - Overall
  bal_gen <- data.frame(
    rbind(
      balance_fn(the_data = d, "gender", comparison),
      balance_fn(the_data = d, "non_white", comparison),
      balance_fn(the_data = d, "education", comparison),
      balance_fn(the_data = d, "income", comparison),
      balance_fn(the_data = d, "pid", comparison),
      balance_fn(the_data = d, "ideo", comparison),
      balance_fn(the_data = d, "fam_movement", comparison),
      balance_fn(the_data = d, "dv_pca_metoo", comparison),
```

```r
    balance_fn(the_data = d, "dv_pca_general", comparison)
  ))
bal_gen$Covariate <- factor(bal_labs, levels=rev(bal_labs))
rownames(bal_gen) <- bal_gen$Covariate

# Descriptives Differences in Selecting Treatment - Men
bal_gen_male <- data.frame(
  rbind(
    balance_fn(the_data = d[d$gender==0,],
               "gender", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "non_white", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "education", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "income", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "pid", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "ideo", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "fam_movement", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "dv_pca_metoo", comparison),
    balance_fn(the_data = d[d$gender==0,],
               "dv_pca_general", comparison)
  ))
bal_gen_male$Covariate <- factor(bal_labs, levels=rev(bal_labs))
rownames(bal_gen_male) <- bal_gen_male$Covariate
bal_gen_male[1,3] <-NA


# Descriptives Differences in Selecting Treatment - Women
bal_gen_female <- data.frame(
  rbind(
    balance_fn(the_data = d[d$gender==1,], "gender", comparison),
    balance_fn(the_data = d[d$gender==1,], "non_white", comparison),
    balance_fn(the_data = d[d$gender==1,], "education", comparison),
    balance_fn(the_data = d[d$gender==1,], "income", comparison),
    balance_fn(the_data = d[d$gender==1,], "pid", comparison),
    balance_fn(the_data = d[d$gender==1,], "ideo", comparison),
    balance_fn(the_data = d[d$gender==1,], "fam_movement", comparison),
    balance_fn(the_data = d[d$gender==1,], "dv_pca_metoo", comparison),
    balance_fn(the_data = d[d$gender==1,], "dv_pca_general", comparison)
  ))
bal_gen_female$Covariate <- factor(bal_labs, levels=rev(bal_labs))
rownames(bal_gen_female) <- bal_gen_female$Covariate
bal_gen_female[1,3] <-NA
```

```r
# Descriptives Differences in Selecting Treatment - Democrats
bal_gen_dem <- data.frame(
  rbind(
    balance_fn(the_data = d[d$pid<4,], "gender", comparison),
    balance_fn(the_data = d[d$pid<4,], "non_white", comparison),
    balance_fn(the_data = d[d$pid<4,], "education", comparison),
    balance_fn(the_data = d[d$pid<4,], "income", comparison),
    balance_fn(the_data = d[d$pid<4,], "pid", comparison),
    balance_fn(the_data = d[d$pid<4,], "ideo", comparison),
    balance_fn(the_data = d[d$pid<4,], "fam_movement", comparison),
    balance_fn(the_data = d[d$pid<4,], "dv_pca_metoo", comparison),
    balance_fn(the_data = d[d$pid<4,], "dv_pca_general", comparison)
  ))
bal_gen_dem$Covariate <- factor(bal_labs, levels=rev(bal_labs))
rownames(bal_gen_dem) <- bal_gen_dem$Covariate

# Descriptives Differences in Selecting Treatment - Republicans
bal_gen_rep <- data.frame(
  rbind(
    balance_fn(the_data = d[d$pid>4,], "gender", comparison),
    balance_fn(the_data = d[d$pid>4,], "non_white", comparison),
    balance_fn(the_data = d[d$pid>4,], "education", comparison),
    balance_fn(the_data = d[d$pid>4,], "income", comparison),
    balance_fn(the_data = d[d$pid>4,], "pid", comparison),
    balance_fn(the_data = d[d$pid>4,], "ideo", comparison),
    balance_fn(the_data = d[d$pid>4,], "fam_movement", comparison),
    balance_fn(the_data = d[d$pid>4,], "dv_pca_metoo", comparison),
    balance_fn(the_data = d[d$pid>4,], "dv_pca_general", comparison)
  ))
bal_gen_rep$Covariate <- factor(bal_labs, levels=rev(bal_labs))
rownames(bal_gen_rep) <- bal_gen_rep$Covariate

# Create data frame for plotting
fig_df <- rbind(
  data.frame(
    bal_gen,
    Group = "Overall",
    Type = "Overall"
  ),
  data.frame(
    bal_gen_male,
    Group = "Men",
    Type = "By Gender"
  ),
  data.frame(
    bal_gen_female,
    Group = "Women",
    Type = "By Gender"
  ),
  data.frame(
```

```r
    bal_gen_rep,
    Group = "Republicans",
    Type = "By Partisanship"
  ),
  data.frame(
    bal_gen_female,
    Group = "Democrats",
    Type = "By Partisanship"
  )
)

# Set labels as factors for plotting
fig_df$Type <- factor(fig_df$Type, levels = unique(fig_df$Type))
fig_df$Group <- factor(fig_df$Group , levels = unique(fig_df$Group ))

# Create Figure
fig <- fig_df %>%
  ggplot(
    aes(Covariate, Difference, col = Group, shape = Group)
  )+
  geom_hline(
    yintercept = 0, linetype = "dashed",alpha = .5
  )+
  facet_wrap(
    ~Type, ncol = 3, drop = T)+
  geom_point(
    aes(shape=Group),
    position = position_dodge(width = .5), size=2
  )+
  geom_linerange(
    aes(ymin=ll,ymax=ul),
    size=.3,
    position = position_dodge(width = .5))+
  geom_linerange(
    aes(ymin=ll90,ymax=ul90),
    size=.6,
    position = position_dodge(width = .5))+
  theme(
    axis.text.x = element_text(angle = 0, hjust = 1)
  )+
  ylab("Difference\n(Treatment Selectors - Treatment Avoiders)")+
  coord_flip()+
  theme_bw()+
  scale_color_grey(start = 0, end = .75)+
  scale_shape_manual(values = c(16, 17, 15, 23, 4))+
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  )
```

```
  # Display Figure
  fig

}
```

- `effects_fn()`: Wrapper function to calculated ATE, ACTEs, CACTEs, and CATEs

```
effects_fn <- function(the_dat,
                       the_dv,
                       the_lab = c("ATE",
                                   "ACTE:","Select Treatment","Avoid
Treatment",
                                   "CACTE:","Female Treatment","Male
Treatment",
                                   "CATE:",
                                   "White","Non-White",
                                   "Male","Female",
                                   "Democrat","Independent","Republican",
                                   "Liberal","Moderate","Conservative",
                                   "College Degree ","No Degree",
                                   "Familiar","Unfamiliar"
                       ),...){
  # INPUTS:
    # the_dat: data frame
    # dv1: outcome
    # the_lab: object containing row names
  # OUTPUTS:
    # result: dataframe containing ATE, ACTES, CACTEs, CATE


  # Calculate ATE, ACTE, CACTEs
  tmp <- rbind(
    diff_fn(the_dat, dv1=the_dv,c("Control","Treatment")),
    rep(NA,7),
    acte_fn(the_dat, dv2=the_dv,c("Control","Selection"),w = T),
    acte_fn(the_dat, dv2=the_dv,c("Selection","Treatment"),w = T),
    rep(NA,7),
    cacte_fn(the_dat, dv3=the_dv),
    rep(NA,7),
    diff_fn(the_data =
the_dat[the_dat$non_white==0,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$non_white==1,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$gender==0,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$gender==1,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$pid<4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$pid==4,],the_dv,c("Control","Treatment")),
```

```
    diff_fn(the_data =
the_dat[the_dat$pid>4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$ideo<4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$ideo==4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$ideo>4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$education>4,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$education<5,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$fam_movement>2,],the_dv,c("Control","Treatment")),
    diff_fn(the_data =
the_dat[the_dat$fam_movement<3,],the_dv,c("Control","Treatment"))
    )

  results <- data.frame(tmp)

  # Format labels
  results$Estimate <- as.character(the_lab)
  results$Estimate <- factor(results$Estimate,levels=rev(the_lab))
  results$Estimand <- c("ATE",
                        rep("ACTE",3),
                        rep("CACTE",3),
                        rep("CATE",15)

  )
  results$Estimand <- factor(results$Estimand, levels
=c("ATE","ACTE","CACTE","CATE"))

  return(results)
}
```

- `plot_effects_fn()`: Wrapper to plot results of `effects_fn()`

```
plot_effects_fn <- function(d, dv){
  # INPUTS:
    # d: data frame
    # dv: outcome
  # OUTPUTS:
    # fig: ggplot of results

  # Create dataframe of effects for plotting
  fig_df <- rbind(
    data.frame(
      effects_fn(d, dv),
      Group = "Overall",
      Type = "Overall"
```

```r
  ),
  data.frame(
    effects_fn(d[d$gender==0,], dv),
    Group = "Men",
    Type = "By Gender"),
  data.frame(
    effects_fn(d[d$gender==1,], dv),
    Group = "Women",
    Type = "By Gender"),
  data.frame(
    effects_fn(d[d$pid > 4,], dv),
    Group = "Republicans",
    Type = "By Partisanship"),
  data.frame(
    effects_fn(d[d$pid < 4,], dv),
    Group = "Democrats",
    Type = "By Partisanship")
)

# Set labels as factors for plotting
fig_df$Type <- factor(fig_df$Type, levels = unique(fig_df$Type))
fig_df$Group <- factor(fig_df$Group , levels = unique(fig_df$Group ))


# Creat Figure
fig <- fig_df %>%
  filter(Estimand!="CATE") %>%
  ggplot(
    aes(Estimate, Difference,col=Group,shape=Group)
  )+
  geom_hline(yintercept = 0,linetype="dashed",alpha=.5)+
  facet_grid(~Type)+
  geom_point(
    aes(shape=Group),
    position = position_dodge(width = .5),size=2
  )+
  geom_linerange(
    aes(ymin=ll,ymax=ul),
    size=.3,
    position = position_dodge(width = .5)
  )+
  geom_linerange(
    aes(ymin=ll90,ymax=ul90),
    size=.6,
    position = position_dodge(width = .5)
  )+
  ylab("Difference\n(Treatment Selectors - Treatment Avoiders)")+
  coord_flip()+
  theme_bw()+
  theme(
```

```
      panel.grid.minor = element_blank(),
      legend.position = "bottom"
    )
  # Format axis labels
  fig <- fig +
    theme(
      axis.text.y = element_text(
        face = ifelse(rev(fig$data$Estimate) %in% c("ATE", "ACTE:", "CACTE:",
"CATE:"),
                      "bold", "italic")
      )
    ) +
    scale_shape_manual(values = c(16, 17,15,23,4))

  return(fig)


}
```

- format_ci_fn(): Helper function to format CIs

```
format_ci_fn <- function(est){
  paste("[",sprintf("%.2f",est[,"ll"]),", ",
sprintf("%.2f",est[,"ul"]),"]",sep="")
}
```

- `table_fn()`: Wrapper to format results of `plot_effects_fn()` as LaTeX table

```
table_fn <- function(d, the_cap=NULL){
  # INPUTS:
    # d: a dataframe of effects
  # OUTUTS
    # A LaTeX table
  tab_df <- d %>%filter(Estimand!="CATE") %>% na.omit()


  tab <- data.frame(matrix(NA,nrow=10,ncol=6))
  est_seq <- seq(1,dim(tab)[1], by=2)
  ci_seq <- seq(2,dim(tab)[1], by=2)
  tab[est_seq,1] <- as.character(tab_df[tab_df$Group == "Overall",
"Estimate"])
  tab[ci_seq,1] <- ""

  tab[est_seq,2] <- sprintf("%.2f",tab_df[tab_df$Group == "Overall",
"Difference"])
  tab[ci_seq,2] <- format_ci_fn(tab_df[tab_df$Group == "Overall",])

  tab[est_seq,3] <- sprintf("%.2f",tab_df[tab_df$Group == "Men",
"Difference"])
  tab[ci_seq, 3] <- format_ci_fn(tab_df[tab_df$Group == "Men",])
```

```
  tab[est_seq,4] <- sprintf("%.2f",tab_df[tab_df$Group == "Women",
"Difference"])
  tab[ci_seq, 4] <- format_ci_fn(tab_df[tab_df$Group == "Women",])

  tab[est_seq,5] <- sprintf("%.2f",tab_df[tab_df$Group == "Republicans",
"Difference"])
  tab[ci_seq, 5] <- format_ci_fn(tab_df[tab_df$Group == "Republicans",])

  tab[est_seq,6] <- sprintf("%.2f",tab_df[tab_df$Group == "Democrats",
"Difference"])
  tab[ci_seq, 6] <- format_ci_fn(tab_df[tab_df$Group == "Democrats",])

  colnames(tab) <- c("","Overall","Men","Women","Republicans","Democrats")

  kable(
    tab,
    booktabs = TRUE,
    caption = the_cap,
    digits=2,
    align = c("l",rep("c",3))
  ) %>%
    kable_styling(latex_options = c("HOLD_position",font_size=10)) %>%
    kableExtra::group_rows("ATE",1,2) %>%
    kableExtra::group_rows("ACTE",3,6) %>%
    kableExtra::group_rows("CACTE",7,10)%>%
    footnote(general = "The table provides point estimates and 95% confidence
intervals for treatment effect estimated from the full sample and separately
by gender and partisanship",
             threeparttable = T,
             fixed_small_size = T)
}
```

- `table_app_fn()`: Wrapper to format results of `effects_fn()` as LaTeX tables for appendix

```
table_app_fn <- function(d, dv, g,eg,the_cap="Treatment Effects"){
  # INPUTS:
    # d: data frame
    # dv: outcome
    # g: variable to group by (Quotes)
    # eg: variable to group by (No quotes)
    # the_cap: caption for table
  # OUTPUTS:
    # tab: table of results
  tmp <- effects_fn(d,dv)%>%filter(Estimand!="CATE")
  tmp <- tmp[-c(2,5,8),-c(5:7,9)]
```

```r
  the_vals <- na.omit(unlist(unique(d[,g])))

  enquo_g <- enquo(eg)
  quo_g <- quo_name(enquo(eg))

  tmp1 <- effects_fn(d[d[,quo_g]==the_vals[1],],
dv)%>%filter(Estimand!="CATE")
  tmp1[,g] <- the_vals[1]
  tmp1 <- tmp1[-c(2,5,8),-c(5:7,9)]

  tmp2 <- effects_fn(d[d[,quo_g]==the_vals[2],],
dv)%>%filter(Estimand!="CATE")
  tmp2[,g] <- the_vals[2]
  tmp2 <- tmp2[-c(2,5,8),-c(5:7,9)]



  d_m <- d %>%filter(gender==0)
  tmp1_m <- effects_fn(d_m[d_m[,quo_g]==the_vals[1],],
dv)%>%filter(Estimand!="CATE")
  tmp1_m[,g] <- the_vals[1]
  tmp1_m <- tmp1_m[-c(2,5,8),-c(5:7,9)]

  tmp2_m <- effects_fn(d_m[d_m[,quo_g]==the_vals[2],],
dv)%>%filter(Estimand!="CATE")
  tmp2_m[,g] <- the_vals[2]
  tmp2_m <- tmp2_m[-c(2,5,8),-c(5:7,9)]

  d_f <- d %>%filter(gender==1)
  tmp1_f <- effects_fn(d_f[d_f[,quo_g]==the_vals[1],],
dv)%>%filter(Estimand!="CATE")
  tmp1_f[,g] <- the_vals[1]
  tmp1_f <- tmp1_f[-c(2,5,8),-c(5:7,9)]

  tmp2_f <- effects_fn(d_f[d_f[,quo_g]==the_vals[2],],
dv)%>%filter(Estimand!="CATE")
  tmp2_f[,g] <- the_vals[2]
  tmp2_f <- tmp2_f[-c(2,5,8),-c(5:7,9)]


  tab <- data.frame(matrix(NA,nrow=dim(tmp)[1]*2,ncol=7))
  est_seq <- seq(1,dim(tab)[1], by=2)
  ci_seq <- seq(2,dim(tab)[1], by=2)

  tab[est_seq,1] <- as.character(tmp[,c("Estimate")])
  tab[ci_seq,1] <- ""

  tab[est_seq,2] <- sprintf("%.2f",tmp1[,c("Difference")])
  tab[ci_seq, 2] <- format_ci_fn(tmp1)
```

```
  tab[est_seq, 3] <- sprintf("%.2f",tmp2[,c("Difference")])
  tab[ci_seq, 3] <- format_ci_fn(tmp2)

  tab[est_seq,4] <- sprintf("%.2f",tmp1_m[,c("Difference")])
  tab[ci_seq, 4] <- format_ci_fn(tmp1_m)
  tab[est_seq, 5] <- sprintf("%.2f",tmp2_m[,c("Difference")])
  tab[ci_seq, 5] <- format_ci_fn(tmp2_m)


  tab[est_seq,6] <- sprintf("%.2f",tmp1_f[,c("Difference")])
  tab[ci_seq, 6] <- format_ci_fn(tmp1_f)
  tab[est_seq, 7] <- sprintf("%.2f",tmp2_f[,c("Difference")])
  tab[ci_seq, 7] <- format_ci_fn(tmp2_f)
  colnames(tab) <- c(" ",the_vals[1],the_vals[2],
                     the_vals[1],the_vals[2],
                     the_vals[1],the_vals[2])

  tab <- kable(tab,
               booktabs = TRUE,
               caption = the_cap,
               digits=2,
               align = c("l",rep("c",6))) %>%
    kable_styling(latex_options = c("HOLD_position",font_size=10)) %>%
    kableExtra::group_rows("ATE",1,2) %>%
    kableExtra::group_rows("ACTE",3,6) %>%
    kableExtra::group_rows("CACTE",7,10)%>%
    add_header_above(c(" " = 1, "Full Sample" = 2, "Men" = 2, "Women" = 2))
  return(tab)

}
```

Functions to conduct power simulations

- data_fn(): Simulate data for power simulations

```
# Generate simulated data
data_fn <- function(N=1000,
                    sigma = 2,
                    p_treat=.5,
                    prop_select=.5,
                    p_treat_select = c(.25,.5,.25),
                    tau_st = .5,
                    tau_af = -.5,
                    tau_am = .5,
                    select_effect = 0,
                    ...
){
  # INPUTS:
    # N: sample size
    # sigma: SE of outcome
    # p_treat: Prob of treatment assignment in experiment condition
```

```
    # prop_select: Proportion selecting treatment in choice conditoin
    # p_treat_select: Prob of treatment assignment for CACTE
    # tau_st: effect among those selecting treatment
    # tau_af: effect among those avoiding treatment assigned to receive
female treatment
    # tau_am: effect among those avoiding treatment assigned to receive mael
treatment
    # select_effect: creates correlation between outcome and selecting
treatment
  # OUTPUTS:
    # df: data frame of simualted responses

  select <- rbinom(n=N, size=1,prob = prop_select)
  avoid <- as.numeric(select!=1)

  # Baseline
  Y0 <- rnorm(n=N, mean = 0, sd = sigma) + select_effect * select

  # Condition
  C <- sample(c("Choice","Experiment"),size=N,
              prob = c(prop_select, 1- prop_select),replace = T)
  n_exp <- sum(C=="Experiment")
  n_ch_a <- sum(C=="Choice" & avoid==1)

  # Treatment status in experimental arm
  D_exp <- rep("Selection",N)
  D_exp[C=="Experiment"] <- sample(c("Treatment","Control"),size=n_exp,
                                   prob = c(p_treat,1-p_treat), replace = T)

  D_ch <- rep("Experiment",N)
  D_ch[C=="Choice" & avoid==1] <- sample(c("Treatment","Control",
"Alternative"),
                                         size=n_ch_a,
                                         prob = p_treat_select,replace = T)
  n_choice <- sum(C == "Choice")
  n_select <- sum(C=="Choice" & avoid==0)
  n_avoid <- sum(C=="Choice" & avoid==1)
  n_control <- sum(D_ch == "Control")
  weights <- rep(1,N)
  weights[C=="Choice" & avoid==0 & D_ch == "Control"] <-1/(n_select/n_choice)
  weights[C=="Choice" & avoid==1 & D_ch == "Control"] <-1/(n_control/n_avoid)

  # Potential outcome is conditional on preferences
  Y1 <- Y0 + tau_st*(select == 1 & D_exp == "Treatment") + tau_af*(avoid == 1
& D_exp == "Treatment" ) +
    tau_st*(select == 1 & C == "Choice")+
    tau_af*(avoid == 1 & D_ch == "Treatment" ) + tau_am*(avoid == 1 & D_ch ==
"Alternative" )
```

```
  # Observed Outcome

  Y <- rep(NA,N)
  Y[C=="Experiment"] <-
Y0[C=="Experiment"]*(D_exp[C=="Experiment"]=="Control") +
    Y1[C=="Experiment"]*(D_exp[C=="Experiment"]=="Treatment")
  Y[C=="Choice" & select == 1] <- Y1[C=="Choice" & select == 1]
  Y[C=="Choice" & select == 0 & D_ch == "Control"] <- Y0[C=="Choice" & select
== 0 & D_ch == "Control"]
  Y[C=="Choice" & select == 0 & D_ch == "Treatment"] <- Y1[C=="Choice" &
select == 0 & D_ch == "Treatment"]
  Y[C=="Choice" & select == 0 & D_ch == "Alternative"] <- Y1[C=="Choice" &
select == 0 & D_ch == "Alternative"]


  treatment = rep(NA, N)
  treatment[C=="Experiment" & D_exp == "Treatment"] <- "Treatment"
  treatment[C=="Experiment" & D_exp == "Control"] <- "Control"
  treatment[C=="Choice"] <- "Selection"
  treatment[C=="Choice" & D_ch == "Treatment" ] <- NA
  treatment[C=="Choice" & D_ch == "Alternative" ] <- NA

  avoid01 <- NA
  avoid01[C == "Choice" & avoid == 1] <- 1
  avoid01[C == "Choice" & avoid == 0] <- 0

  select01 <- NA
  select01[C == "Choice" & avoid == 0] <- 1
  select01[C == "Choice" & avoid == 1] <- 0

  df <- data.frame(Y0,Y1,Y,true_diff = Y1-Y0,
                   C, treatment, select, avoid,
                   select01,avoid01,
                   D_exp, D_ch, weights)
  return(df)
}
```

- power_fn(): Calculate power for given treatment effects

```
# Calculate power for given parameters
power_fn <- function(sims,
                     p_N=1000,
                     p_sigma = 1,
                     p_p_treat=.5,
                     p_prop_select=.5,
                     p_p_treat_select = c(.25,.5,.25),
                     p_tau_st = .5,
                     p_tau_af = -.5,
```

```
                    p_tau_am = .5,
                    p_select_effect = 0,

                    ...){
  # INPUTS:
    # sims: number of simulations
    # p_*: arguments passed to data_fn()
  # OUTPUTS:
    # results: results of power simulation

  ate <- rep(NA,sims)
  acte_s <- rep(NA,sims)
  acte_a <- rep(NA,sims)
  cacte_female <- rep(NA,sims)
  cacte_male <- rep(NA,sims)

  sig_ate <- rep(NA,sims)
  sig_acte_s <- rep(NA,sims)
  sig_acte_a <- rep(NA,sims)
  sig_cacte_female <- rep(NA,sims)
  sig_cacte_male <- rep(NA,sims)
  cor_select <- rep(NA,sims)


  for(i in 1:sims){
    df <- data_fn(N=p_N,sigma= p_sigma ,
                 p_treat=p_p_treat,
                 prop_select = p_prop_select,
                 p_treat_select = p_p_treat_select ,
                 tau_st= p_tau_st ,
                 tau_af= p_tau_af ,
                 tau_am= p_tau_am ,
                 select_effect= p_select_effect )
    sig_ate[i] <- diff_fn(df,dv1="Y",c=c("Control","Treatment"),weights =
T)["pval"]
    sig_acte_s[i] <- acte_fn(df,z=c("Control","Selection"), w=T)["pval"]
    sig_acte_a[i] <- acte_fn(df,z=c("Selection","Treatment"), w=T)["pval"]
    sig_cacte_female[i] <- cacte_fn(df)[1,"pval"]
    sig_cacte_male[i] <- cacte_fn(df)[2,"pval"]
    cor_select[i] <- cor(df$Y,df$select)

    ate[i] <- diff_fn(df,dv1="Y",c=c("Control","Treatment"))["Difference"]
    acte_s[i] <- acte_fn(df,z=c("Control","Selection"), w=T)["Difference"]
    acte_a[i] <- acte_fn(df,z=c("Selection","Treatment"),w = T)["Difference"]
    cacte_female[i] <- cacte_fn(df)[1,"Difference"]
    cacte_male[i] <- cacte_fn(df)[2,"Difference"]
  }

  pow_ate <- mean(sig_ate<.05)
```

```
  pow_acte_s <- mean(sig_acte_s<.05)
  pow_acte_a <- mean(sig_acte_a<.05)
  pow_cacte_female <-  mean(sig_cacte_female<.05)
  pow_cacte_male <- mean(sig_cacte_male<.05)
  mn_cor_select <- mean(cor_select)

  mn_ate <- mean(ate)
  mn_acte_s <- mean(acte_s)
  mn_acte_a <- mean(acte_a)
  mn_cacte_female <-  mean(cacte_female)
  mn_cacte_male <- mean(cacte_male)


  results <- rbind(
    c(mn_ate,
      mn_acte_s,
      mn_acte_a,
      mn_cacte_female,
      mn_cacte_male,mn_cor_select),

    c(pow_ate,
      pow_acte_s,
      pow_acte_a,
      pow_cacte_female,
      pow_cacte_male,
      NA
    ))

  return(results)

}
```

- sim_power_fn(): Conduct power simulations over a range of treatment effects

```
# Simulate power for range of treatment effects
sim_power_fn <- function(
  s_sims = 500,
  s_N=1000,
  s_sigma = 1,
  s_p_treat=.5, #
  s_prop_select=.5,
  s_p_treat_select = c(.25,.5,.25),
  s_tau_st = .5,
  s_tau_af = -.5,
  s_tau_am = .5,
  s_select_effect = 0




) {
```

```r
# INPUTS:
  # s_*: arguments passed to power_fn()
# OUTPUTS:
# results: list containing
#            - data frame of  power simulations
#            - summary statitistics
#            - range of correlation between selecting treatment and outcome

# Create matrix to store values
power_mat <- matrix(NA,nrow=5, ncol = length(s_tau_st),
                    dimnames = list(c("ATE","ACTE-Select","ACTE-Avoid",
                                      "CACTE-Female","CACTE-Male"),
                                    s_tau_st
                    )
)
bias_mat <- matrix(NA,nrow=5, ncol = length(s_tau_st),
                   dimnames = list(c("ATE","ACTE-Select","ACTE-Avoid",
                                     "CACTE-Female","CACTE-Male"),
                                   s_tau_st
                   )
)
tmp <- c()
ave_cor <- c()
tmp_df <- data.frame(Estimate=NULL,Tau=NULL)
df <- data.frame(Estimate= NULL,
                 Tau_Select = NULL,
                 Tau_Avoid = NULL ,
                 Tau_Alt = NULL ,
                 Power = NULL )
# Loop over possible values
for(i in 1:length(s_tau_st)){
  tmp <- power_fn(sims = s_sims,
                  p_N = s_N,
                  p_sigma  = s_sigma,
                  p_p_treat = s_p_treat,
                  p_prop_select = s_prop_select,
                  p_p_treat_select  = s_p_treat_select,
                  p_tau_st  = s_tau_st[i],
                  p_tau_af  = s_tau_af[i],
                  p_tau_am  = s_tau_am[i],
                  p_select_effect  = s_select_effect
  )
  ave_cor[i] <- tmp[1,6]
  power_mat[,i] <- tmp[2,1:5]
  tmp2 <- data.frame(Estimate= c("ATE","ACTE-Select","ACTE-Avoid",
                                 "CACTE-Female","CACTE-Male"),
                     Type = c("ATE","ACTE","ACTE","CACTE","CACTE"),
                     Tau_Select = rep(s_tau_st[i],5),
                     Tau_Avoid = rep(s_tau_af[i],5),
                     Tau_Alt = rep(s_tau_am[i],5),
```

```
                          Power = tmp[2,1:5]
    )
    df <- rbind(tmp2,df)
  }
  return(list(df,power_mat,ave_cor))

}
```

- display_power_sim_fn(): Wrapper to display results of power simulations as figure and table

```
# Display results of power simulations
display_power_sim_fn <- function(
  p_s_sims = 1000,
  p_s_N=1000,
  p_s_sigma = 1,
  p_s_p_treat=.4,
  p_s_prop_select=.5,
  p_s_p_treat_select = c(.25,.5,.25),
  p_s_tau_st = .5,
  p_s_tau_af = -.5,
  p_s_tau_am = .5,
  p_s_select_effect = 0,
  lab_effects = "Effects = Equal & offsetting"

){
  # INPUTS:
  # p_s_*: arguments passed to sim_power_fn()
  # OUTPUTS:
  # results: list containing
  #            - figure displaying results
  #            - table formatted for LaTeX

  pow <- sim_power_fn(
    s_sims = p_s_sims,
    s_N= p_s_N,
    s_sigma = p_s_sigma,
    s_p_treat= p_s_p_treat,
    s_prop_select= p_s_prop_select,
    s_p_treat_select = p_s_p_treat_select,
    s_tau_st = p_s_tau_st,
    s_tau_af = p_s_tau_af,
    s_tau_am = p_s_tau_am,
    s_select_effect = p_s_select_effect


  )
  rhos.min <- round(range(pow[[3]]),2) [1]
  rhos.max <- round(range(pow[[3]]),2) [2]
```

```r
  p <- pow[[1]] %>%
    mutate(
      Estimate=factor(Estimate,
                      levels = c("ATE","ACTE-Select","ACTE-Avoid",
                                 "CACTE-Female","CACTE-Male")
      ),
      Type = factor(Type,
                    levels = c("ATE","ACTE","CACTE")
      )
    )%>%
    ggplot(aes(Tau_Select,Power, col=Estimate,linetype=Type))+
    geom_line()+
    ylim(0,1.05) +
    xlim(0,.7) +
    geom_hline(yintercept = .8,linetype = "dashed",col="grey")+
    xlab(expression(tau[Select]))+
    ylab("Power (Probability of Statistical Significance)")+
    annotate(geom = "text",
             hjust = 0,
             y = 1.05,
             x = 0,
             label = paste("Simulations =",scales::comma(p_s_sims))
    )+
    annotate(geom = "text",
             hjust = 0,
             y = 1,
             x = 0,
             label = lab_effects
    )+
    annotate(geom = "text",
             hjust = 0,
             y = .95,
             x = 0,
             label =paste("Prop Select =", round(p_s_prop_select,2))
    ) +
    annotate(geom = "text",
             hjust = 0,
             y = .90,
             x = 0,
             label = paste("Cor = [",rhos.min,", ",rhos.max,"]",sep="")
    )
  tab <- kable(pow[[2]],
               caption = "Power Analysis",
               format = "latex",
               booktabs=T,
               linesep = "",
               digits=2) %>%
    add_header_above(c("","Hypothesized Effect Among
Selectors"=dim(pow[[2]])[2])) %>%
    kable_styling(latex_options = c("hold_position",font_size=10))
```

```
  return(list(p,tab))


}
```

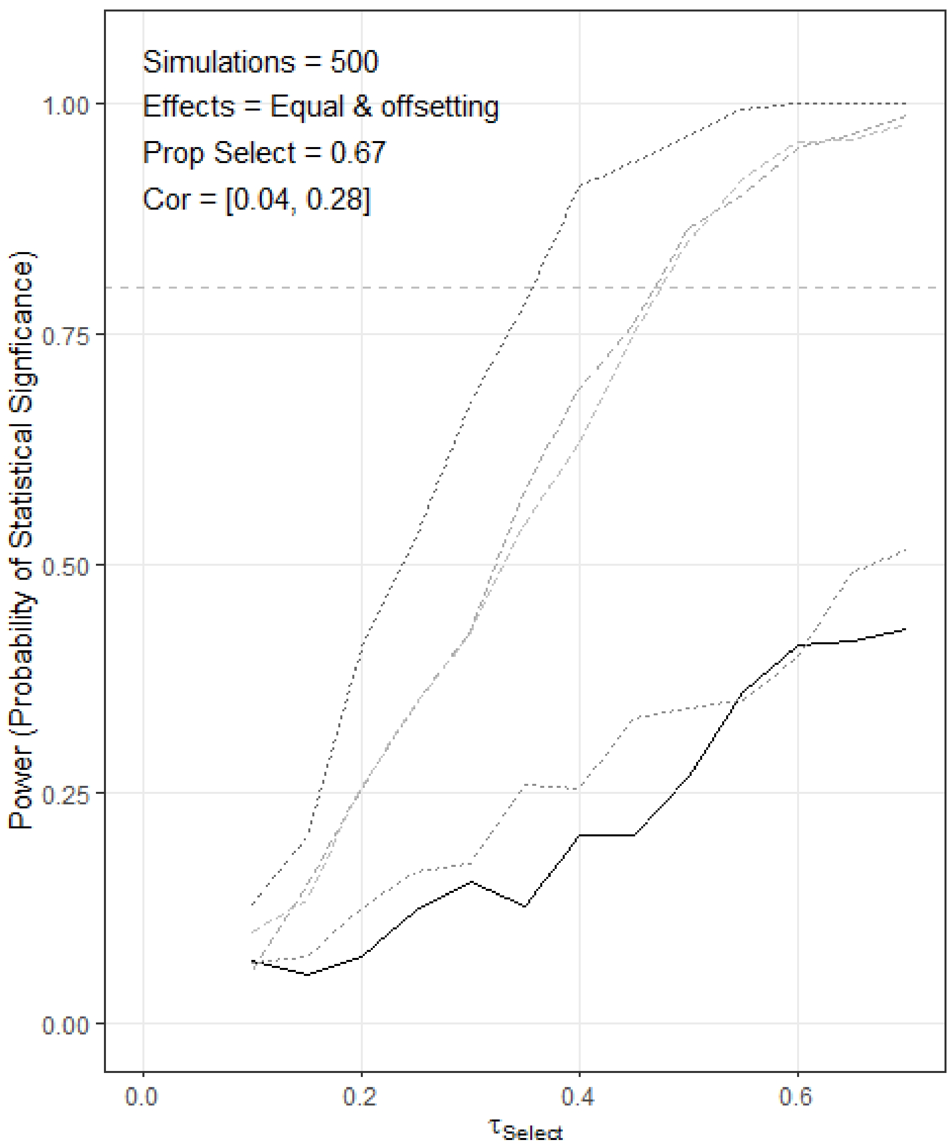
Main Figures

Figure 1: Triply Randomized Parallel Design

Note: Figure created using Adobe Illustrator

Figure 2: Statistical Power with More Selectors than Avoiders, Equal and Offsetting Effects

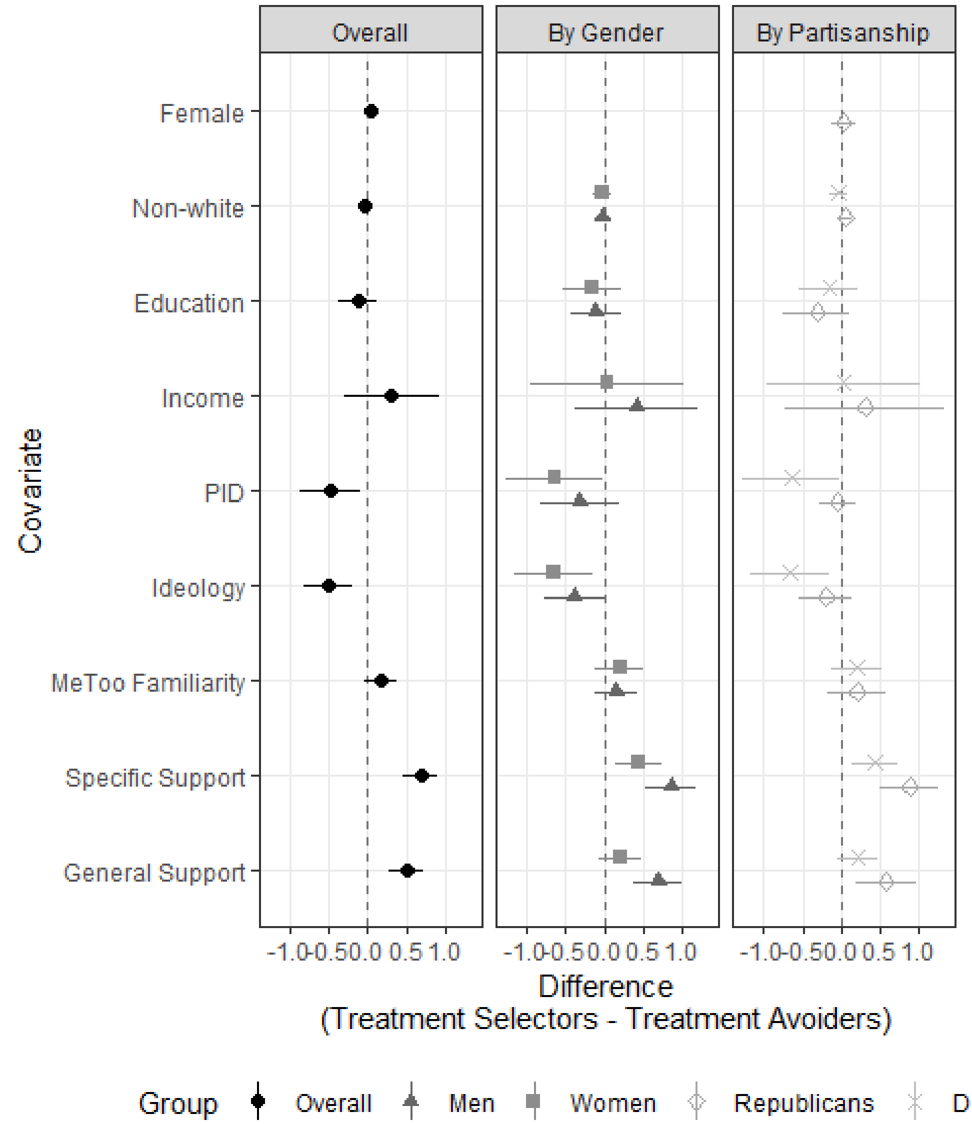
```
#Uncomment to run. 500 Simulation takes ~ 30-40 minutes
# Set random seed
#set.seed(123)

#fig2_power_sim <- display_power_sim_fn(
#                        p_s_sims = 500,
#                        p_s_prop_select = 2/3,
#                        p_s_tau_st = seq(.1,.7,by=.05),
#                        p_s_tau_af = seq(-.1,-.7,by=-.05),
#                        p_s_tau_am = seq(.1,.7,by=.05)
#                        )
#print(fig2_power_sim)

# Format Figure 2
fig2 <- fig2_power_sim [[1]]+
  theme_bw()+
  theme(
    panel.grid.minor = element_blank(),
  )+
  scale_color_grey(start = 0, end = .75)

# Display Figure 2
fig2
```



Statistical Power with More Selectors than Avoiders, Equal and Offsetting Effects

Figure 3: Who is Likely to Seek Out or Avoid the Message of the #MeToo Movement?

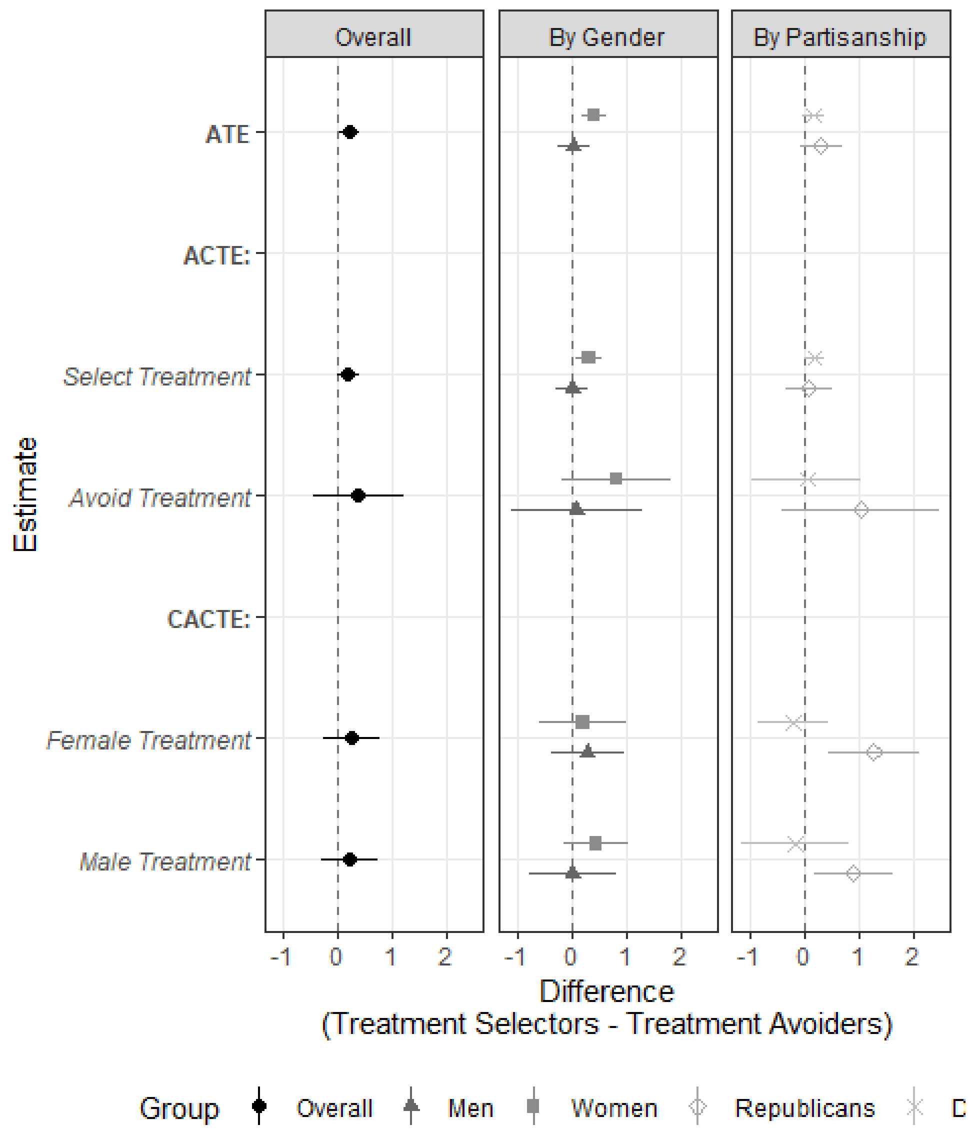
```r
# Create Figure 3
fig3 <- plot_balance_fn(df_mtg)

# Display Figure 3
fig3
```



Who is Likely to Seek Out or Avoid the Message of the #MeToo Movement?

Figure 4: Heterogeneous Effects in the #MeToo MTurk Study

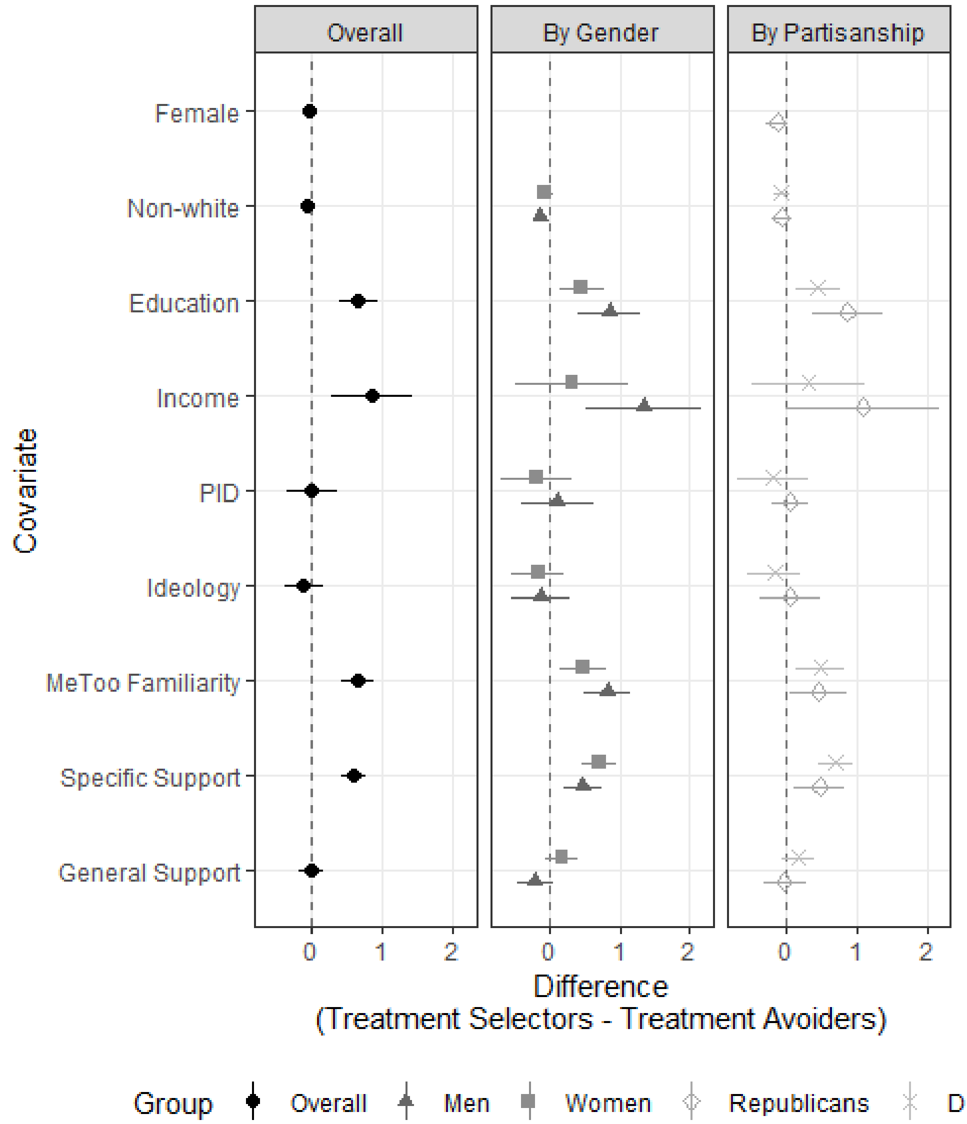
```
# Create Figure 4
fig4 <- plot_effects_fn(df_mtg,"dv_pca_metoo") + scale_color_grey(start = 0,
end = .75)

# Display Figure 4
fig4
```



Heterogeneous Effects in the #MeToo MTurk Study

Figure 5: Who is Likely to Seek Out or Avoid the Message of the #MeToo Movement in a More Nationally Representative Sample?

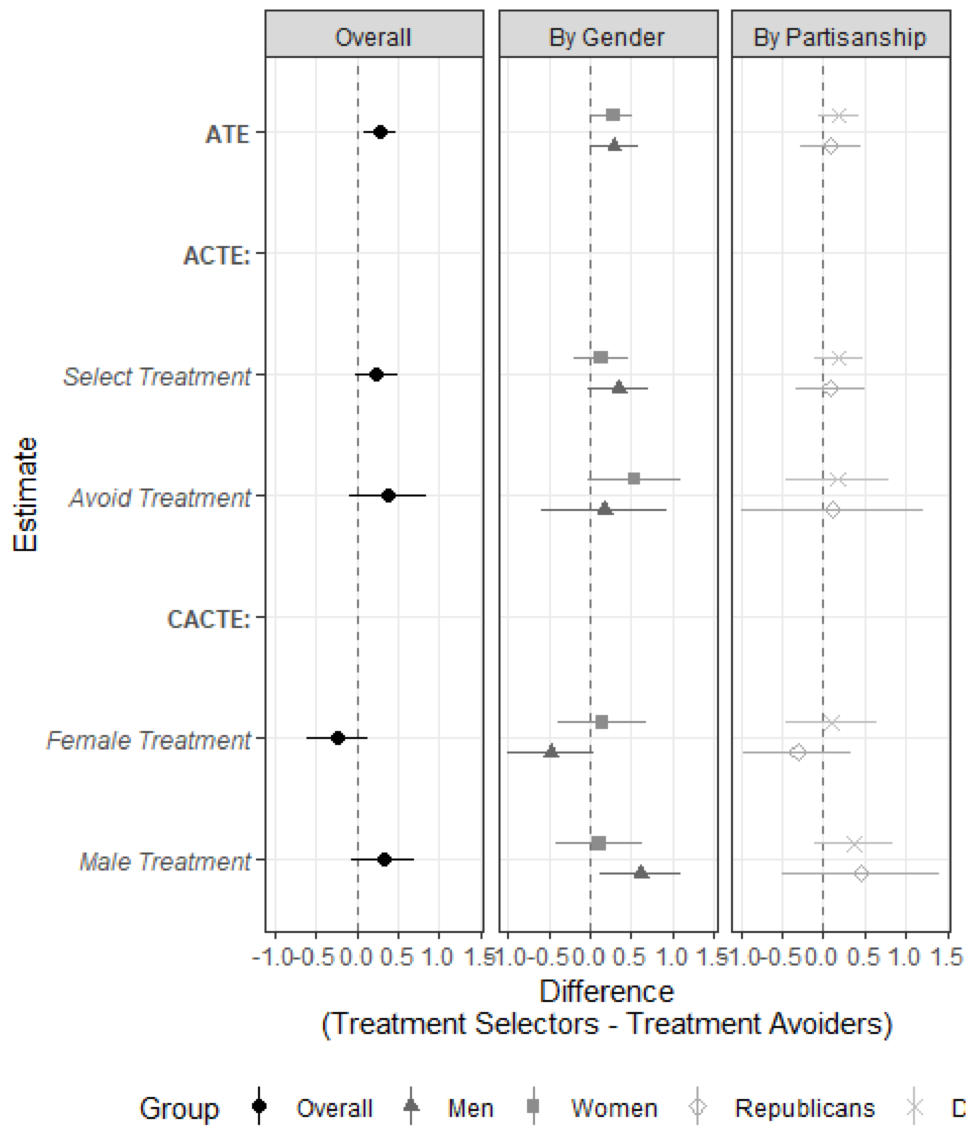
```
# Create Figure 3
fig5 <- plot_balance_fn(df_qg)

# Display Figure 3
fig5
```



Who Seeks Out or Avoids the Message of the #MeToo Movement in a More Nationally Representative Sample?

Figure 6: Heterogeneous Effects in the #MeToo Qualtrics Study

```
# Create Figure 6
fig6 <- plot_effects_fn(df_qg,"dv_pca_metoo") + scale_color_grey(start = 0,
end = .75)

# Display Figure 6
fig6
```



Heterogeneous Effects in the #MeToo Qualtrics Study

Main Tables

Table 1

```
table_fn(fig4$data,
    "Table 1: Treatment Effect Estimates on Specific Support for \\#MeToo
(MTurk Sample)")
```

Table 2

```
table_fn(fig6$data,
    "Table 2: Treatment Effect Estimates on Specific Support for \\#MeToo
(Qualtrics Sample)")
```

Online Appendix

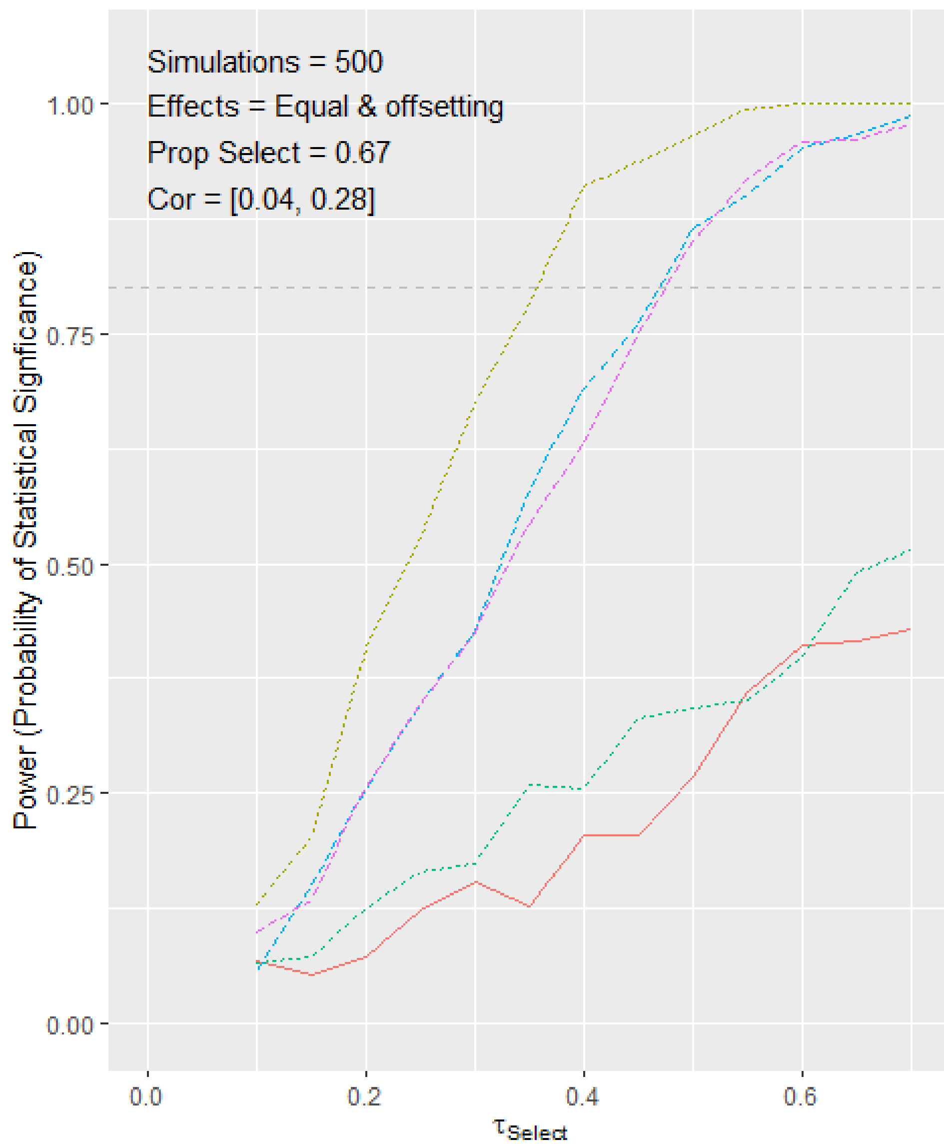
Appendix C Power Simulations

Figure and Table C.1: Statistical Power with Equal Number of Selectors than Avoiders, Equal and Offsetting Effects

```
# Uncomment to run. 500 Simulation takes ~ 30-40 minutes
# Set random seed
#set.seed(123)

#figC1_power_sim <- display_power_sim_fn(
#                       p_s_sims = 500,
#                        p_s_prop_select = .5,
#                        p_s_tau_st = seq(.1,.7,by=.05),
#                        p_s_tau_af = seq(-.1,-.7,by=-.05),
#                        p_s_tau_am = seq(.1,.7,by=.05)
#                        )
#
#figC1_power_sim[[1]]

figC1_power_sim[[2]]
```

Figure and Table C.2: Statistical Power with More Selectors than Avoiders, Equal and Offsetting Effects

```
# Same as Figure 2
fig2_power_sim[[1]]
```



Statistical Power with More Selectors than Avoiders, Equal and Offsetting Effects

```
fig2_power_sim[[2]]
```

Figure and Table C.3: Statistical Power with More Selectors than Avoiders, Equal and Offsetting Effects, and Selection Correlated with Outcome

```
# Uncomment to run. 500 Simulation takes ~ 30-40 minutes

# Set random seed
#set.seed(123)

#figC3_power_sim <- display_power_sim_fn(
#                       p_s_sims = 500,
#                        p_s_prop_select = 2/3,
#                        p_s_select_effect = 0.5,
```

```r
#                        p_s_tau_st = seq(.1,.7,by=.05),
#                        p_s_tau_af = seq(-.1,-.7,by=-.05),
#                        p_s_tau_am = seq(.1,.7,by=.05)
#                        )

#figC3_power_sim[[1]]

figC3_power_sim[[2]]

# Uncomment to save results of power simulations
#save(figC1_power_sim, fig2_power_sim,figC3_power_sim,file =
"power_simulations.rda")
```

Appendix D Descriptive Statistics

Table D.1: Descriptive Statistics for MTurk Sample

```r
the_covariates <- c("female01", "age", "income", "education", "pid", "ideo",
                    "black", "latino", "asian", "fam_movement", "avoid01")
desc_tab_mt<- c()
for(i in 1:length(the_covariates)){
  desc_tab_mt <- cbind(desc_tab_mt,
                       summary(df_mtg[,the_covariates[i]])[1:6]
                       )
}
desc_tab_mt <- t(round(desc_tab_mt,2))
rownames(desc_tab_mt) <- c("Prop. Female", "Age", "Income", "Education",
                           "Party ID", "Ideology",
                           "Prop. Black",
                           "Prop. Latinx",
                           "Prop. Asian",
                           "Familiarity with MeToo",
                           "Prop Avoiding Treatment")



desc_tab_mt_tex <- kable(desc_tab_mt,
      booktabs = TRUE,
               caption = "Descriptive Statistics for MTurk Sample",
               digits=2,
  align = "l") %>%
  kable_styling(latex_options = c("hold_position",font_size=10))
desc_tab_mt_tex
```

Table D.2: Descriptive Statistics for Qualtrics Sample

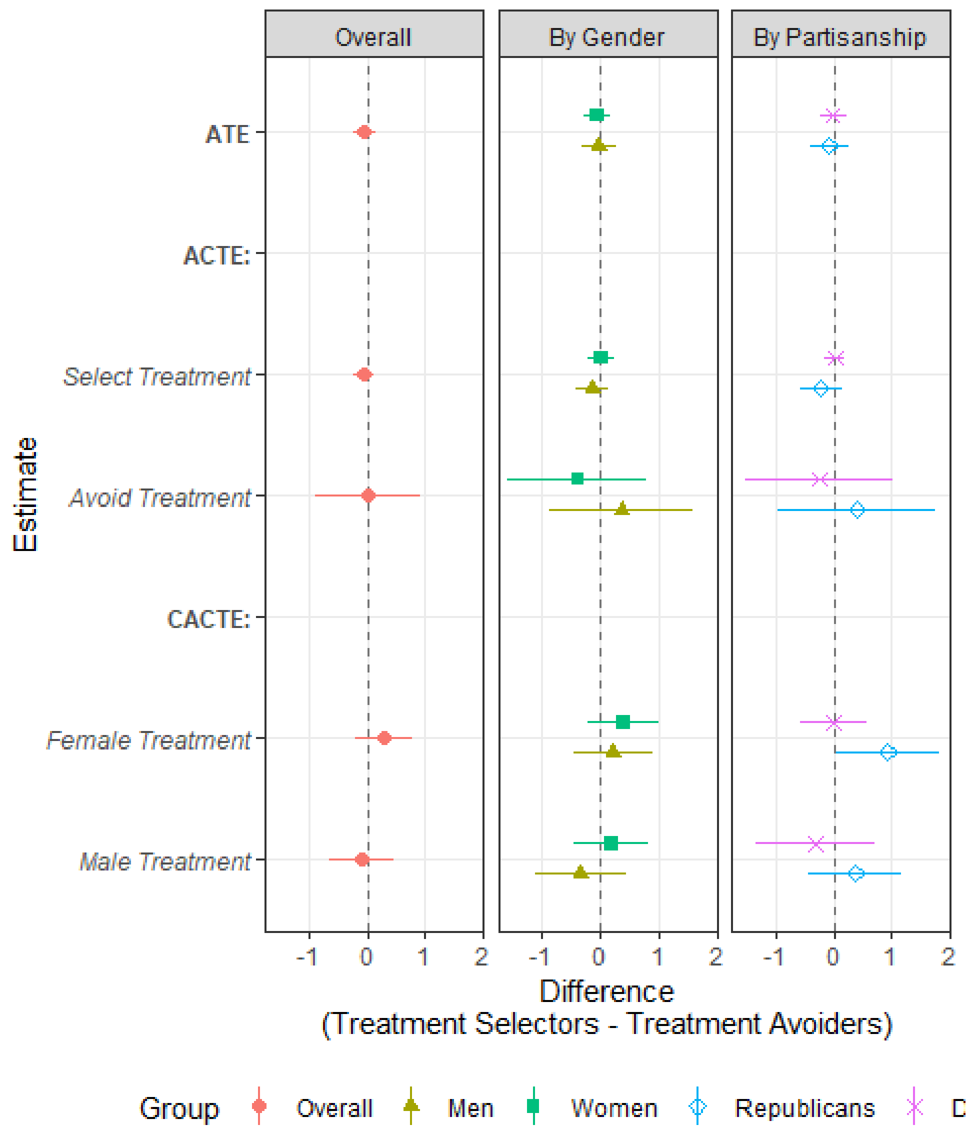
```r
desc_tab_q <- c()
df_qg <- data.frame(df_qg)
for(i in 1:length(the_covariates)){
  desc_tab_q <- cbind(desc_tab_q,
                       summary(na.omit(df_qg[,the_covariates[i]])))

}
```

```r
desc_tab_q <- t(round(desc_tab_q,2))
rownames(desc_tab_q) <- c("Prop. Female", "Age", "Income", "Education",
                           "Party ID", "Ideology",
                           "Prop. Black",
                           "Prop. Latinx",
                           "Prop. Asian",
                           "Familiarity with MeToo",
                           "Prop Avoiding Treatment")



desc_tab_q_tex <- kable(desc_tab_q,
      booktabs = TRUE,
               caption = "Descriptive Statistics for Qualtrics Sample",
               digits=2,
  align = "l") %>%
  kable_styling(latex_options = c("hold_position",font_size=10))
desc_tab_q_tex
```

Appendix E Effects on General Support for Gender Equality

Figure and Table E.1: Effects on General Support for Gender Equality (MTurk Study)

```r
# Create Figure E1
figE1 <- plot_effects_fn(df_mtg, "dv_pca_general")

# Display Figure 4
figE1
```

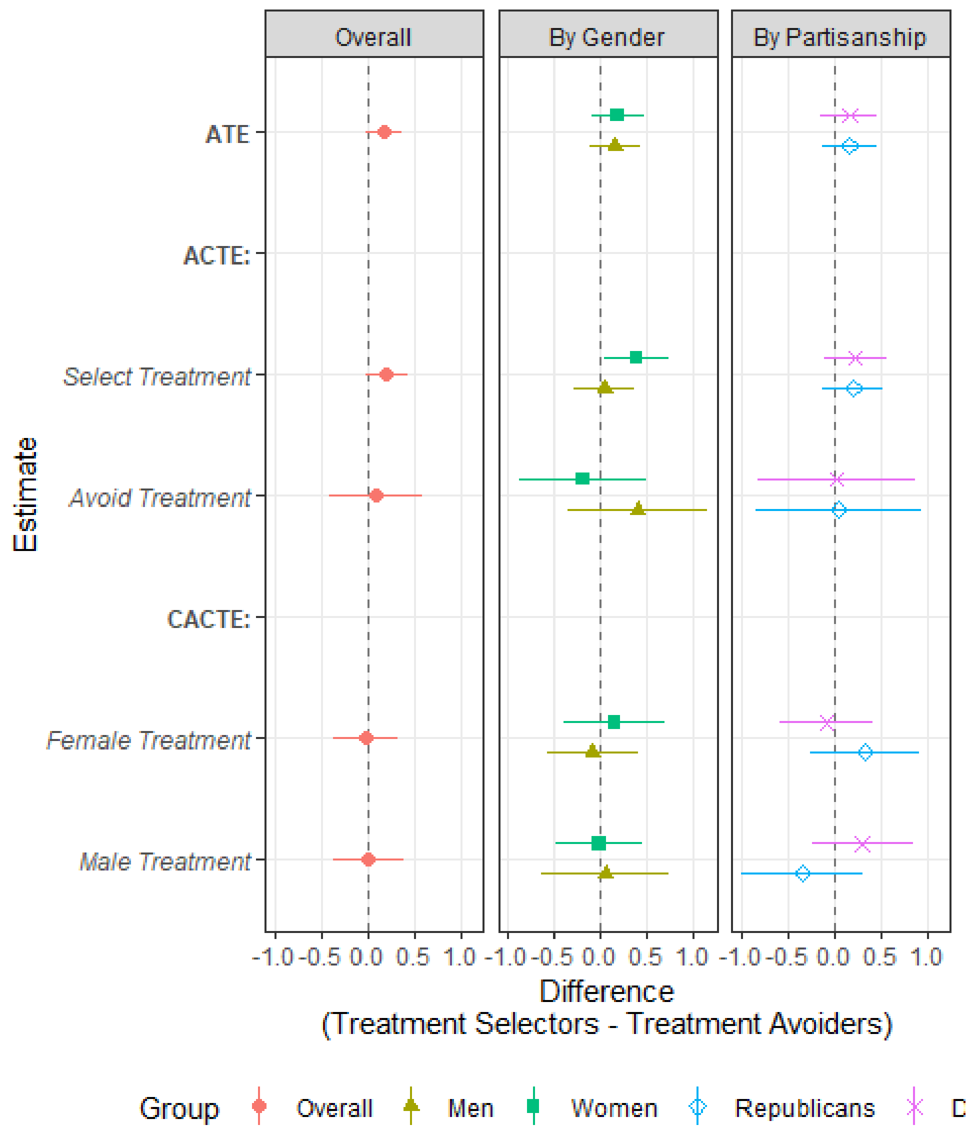


Effects on General Support for Gender Equality (MTurk Study)

```
table_fn(figE1$data,
  "Treatment Effect Estimates on General Support for Gender Equality (MTurk
Sample)"
  )
```

Figure and Table E.2: Effects on General Support for Gender Equality (Qualtrics Study)

```
# Create Figure E2
figE2 <- plot_effects_fn(df_qg, "dv_pca_general")

# Display Figure E2
figE2
```



Effects on General Support for Gender Equality (Qualtrics Study)

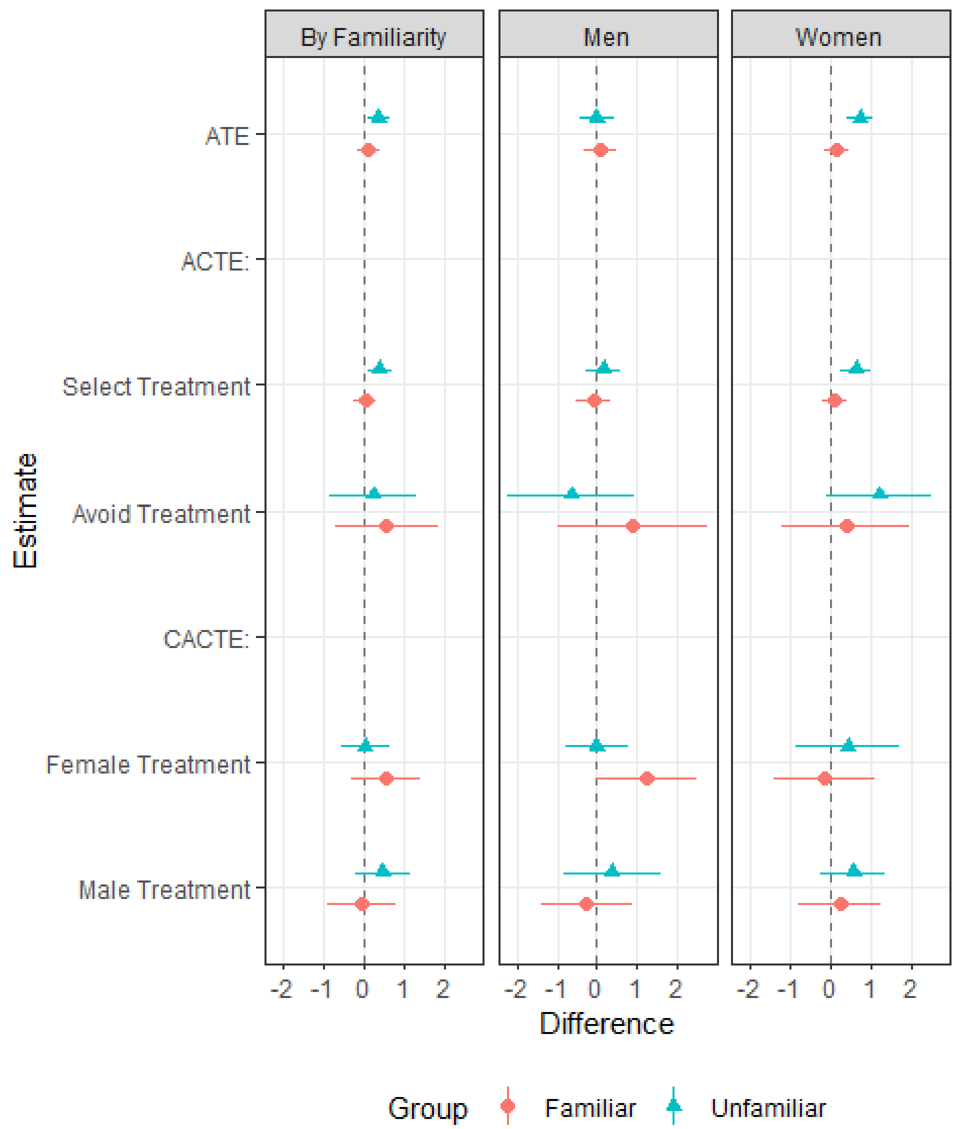
```
table_fn(figE2$data,
  "Treatment Effect Estimates on General Support for Gender Equality (MTurk Sample)"
  )
```

F Additional Analyses

Figure and Table F.1: Treatment Effect Estimates on Specific Support for #MeToo Conditional on Familiarity and Gender (MTurk Sample)

```
figF1_df <- rbind(
  data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement>2,], "dv_pca_metoo"),
  Group = "Familiar",
```

```
  Type = "By Familiarity"),
  data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement<3,], "dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "By Familiarity"),
  data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement>2 & df_mtg$gender==0,],
"dv_pca_metoo"),
  Group = "Familiar",
  Type = "Men"),
  data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement<3& df_mtg$gender==0,],
"dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "Men"),
    data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement>2 & df_mtg$gender==1,],
"dv_pca_metoo"),
  Group = "Familiar",
  Type = "Women"),
  data.frame(
  effects_fn(df_mtg[df_mtg$fam_movement<3& df_mtg$gender==1,],
"dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "Women")
  )

figF1 <- figF1_df %>%
  filter(Estimand != "CATE")%>%
  ggplot(aes(Estimate, Difference,col=Group,shape=Group))+
  geom_hline(yintercept = 0,linetype="dashed",alpha=.5)+
  facet_grid(~Type)+
  geom_point(aes(shape=Group),
               position = position_dodge(width = .5),size=2
      )+
  geom_linerange(aes(ymin=ll,ymax=ul),size=.3,
                       position = position_dodge(width = .5))+
  geom_linerange(aes(ymin=ll90,ymax=ul90),size=.6,
                     position = position_dodge(width = .5))+
  coord_flip()+
  theme_bw()+
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  )

figF1
```



Treatment Effect Estimates on Specific Support for #MeToo Conditional on Familiarity and Gender

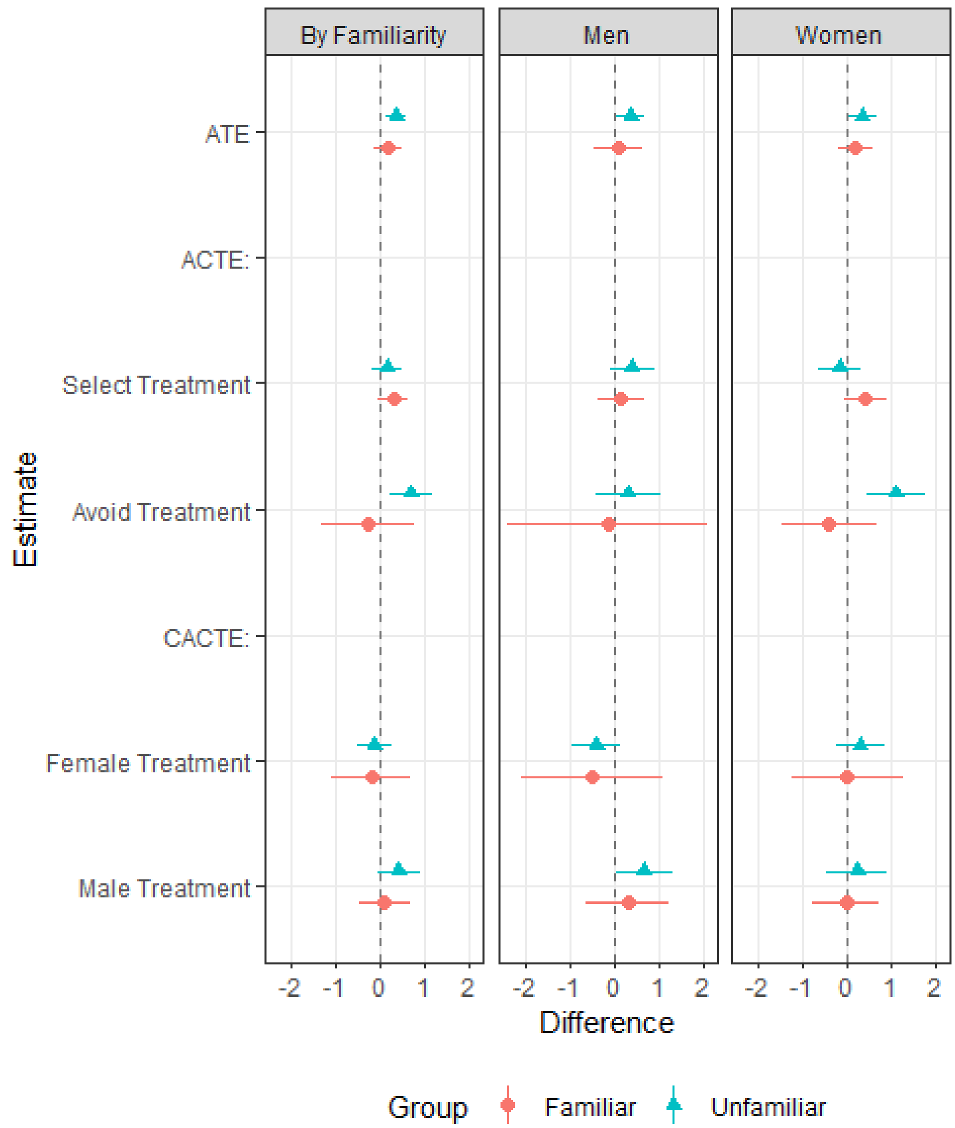
```
# Create Grouping Label
#df_mtg$Familiarity <- ifelse(df_mtg$fam_movement>2, "Familiar",
"Unfamiliar")

table_app_fn(df_mtg, "dv_pca_metoo",
         "Familiarity",
         Familiarity,
         the_cap ="Treatment Effect Estimates on Specific Support for
\\#MeToo Conditional On Familiarity and Gender (MTurk Sample)" )%>%
  footnote(general = "The table provides point estimates and 95% confidence
intervals for treatment effect estimated by level of pre-test familiarity
with the movement overall and by gender",
```

```
            threeparttable = T,
            fixed_small_size = T)
```

Figure and Table F.2: Treatment Effect Estimates on Specific Support for #MeToo Conditional on Familiarity and Gender (Qualtrics Study)

```
figF2_df <- rbind(
  data.frame(
  effects_fn(df_qg[df_qg$fam_movement>2,], "dv_pca_metoo"),
  Group = "Familiar",
  Type = "By Familiarity"),
  data.frame(
  effects_fn(df_qg[df_qg$fam_movement<3,], "dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "By Familiarity"),
  data.frame(
  effects_fn(df_qg[df_qg$fam_movement>2 & df_qg$gender==0,], "dv_pca_metoo"),
  Group = "Familiar",
  Type = "Men"),
  data.frame(
  effects_fn(df_qg[df_qg$fam_movement<3& df_qg$gender==0,], "dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "Men"),
    data.frame(
  effects_fn(df_qg[df_qg$fam_movement>2 & df_qg$gender==1,], "dv_pca_metoo"),
  Group = "Familiar",
  Type = "Women"),
  data.frame(
  effects_fn(df_qg[df_qg$fam_movement<3& df_qg$gender==1,], "dv_pca_metoo"),
  Group = "Unfamiliar",
  Type = "Women")
  )

figF2 <- figF2_df %>%
  filter(Estimand != "CATE")%>%
  ggplot(aes(Estimate, Difference,col=Group,shape=Group))+
  geom_hline(yintercept = 0,linetype="dashed",alpha=.5)+
  facet_grid(~Type)+
  geom_point(aes(shape=Group),
                 position = position_dodge(width = .5),size=2
      )+
  geom_linerange(aes(ymin=ll,ymax=ul),size=.3,
                       position = position_dodge(width = .5))+
  geom_linerange(aes(ymin=ll90,ymax=ul90),size=.6,
                     position = position_dodge(width = .5))+
  coord_flip()+
  theme_bw()+
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  )
```

```
figF2
```



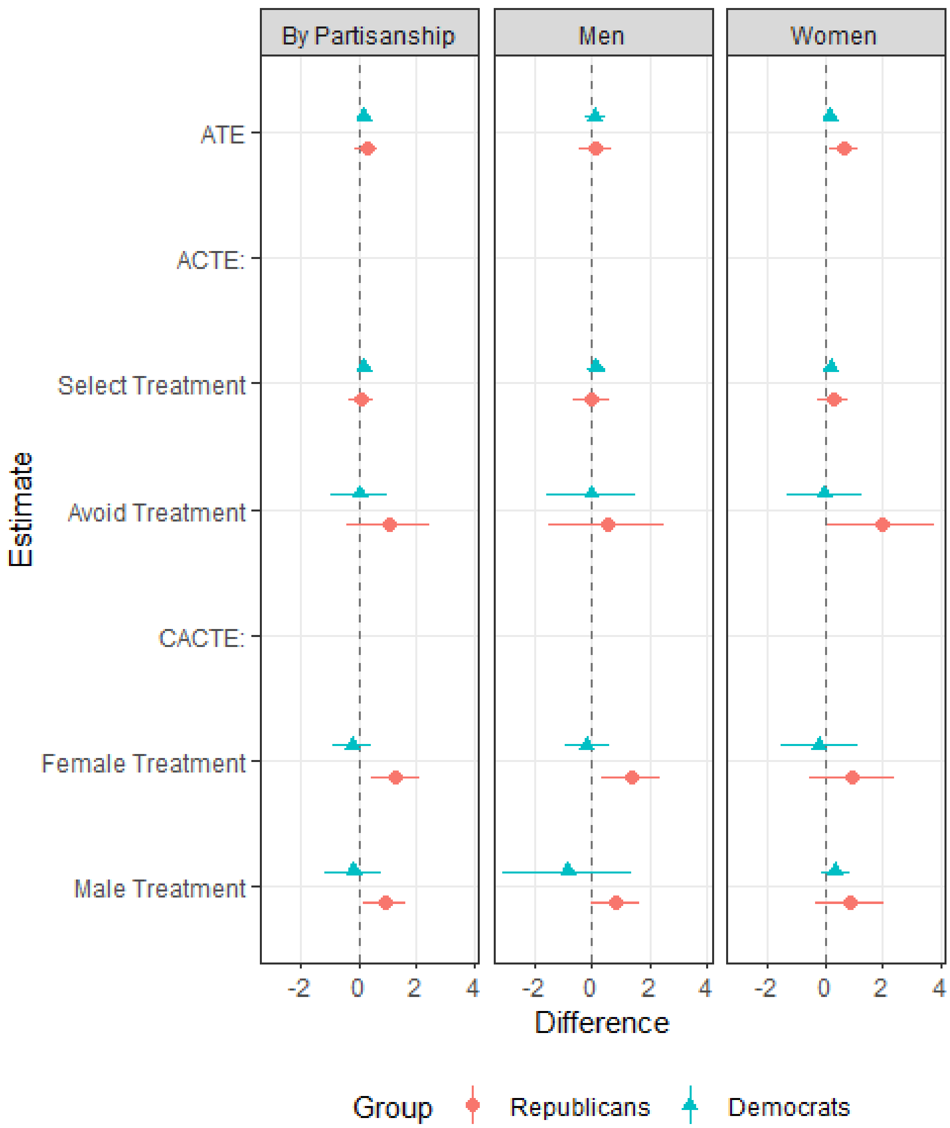
Treatment Effect Estimates on Specific Support for #MeToo Conditional on Familiarity and Gender (Qualtrics Study)

```
table_app_fn(df_qg, "dv_pca_metoo",
         "Familiarity",
         Familiarity,
         the_cap ="Treatment Effect Estimates on Specific Support for
\\#MeToo Conditional On Familiarity and Gender (Qualtrics Sample)" )%>%
  footnote(general = "The table provides point estimates and 95% confidence
intervals for treatment effect estimated by level of pre-test familiarity
with the movement overall and by gender",
           threeparttable = T,
           fixed_small_size = T)
```

Figure and Table F.3: Treatment Effect Estimates on Specific Support for #MeToo Conditional On Partisanship and Gender (MTurk Sample)

```
figF3_df <- rbind(
  data.frame(
  effects_fn(df_mtg[df_mtg$pid>4,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "By Partisanship"),
  data.frame(
  effects_fn(df_mtg[df_mtg$pid<4,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "By Partisanship"),
  data.frame(
  effects_fn(df_mtg[df_mtg$pid>4 & df_mtg$gender==0,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "Men"),
  data.frame(
  effects_fn(df_mtg[df_mtg$pid<4& df_mtg$gender==0,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "Men"),
    data.frame(
  effects_fn(df_mtg[df_mtg$pid>4 & df_mtg$gender==1,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "Women"),
  data.frame(
  effects_fn(df_mtg[df_mtg$pid<4& df_mtg$gender==1,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "Women")
  )

figF3_df$Group <- factor(figF3_df$Group,
                         levels = unique(figF3_df$Group) )

figF3 <- figF3_df %>%
  filter(Estimand != "CATE")%>%
  ggplot(aes(Estimate, Difference,col=Group,shape=Group))+
  geom_hline(yintercept = 0,linetype="dashed",alpha=.5)+
  facet_grid(~Type)+
  geom_point(aes(shape=Group),
                position = position_dodge(width = .5),size=2
      )+
  geom_linerange(aes(ymin=ll,ymax=ul),size=.3,
                      position = position_dodge(width = .5))+
  geom_linerange(aes(ymin=ll90,ymax=ul90),size=.6,
                    position = position_dodge(width = .5))+
  coord_flip()+
  theme_bw()+
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
```

```
  )

figF3
```



Treatment Effect Estimates on Specific Support for #MeToo Conditional On Partisanship and Gender (MTurk Sample)

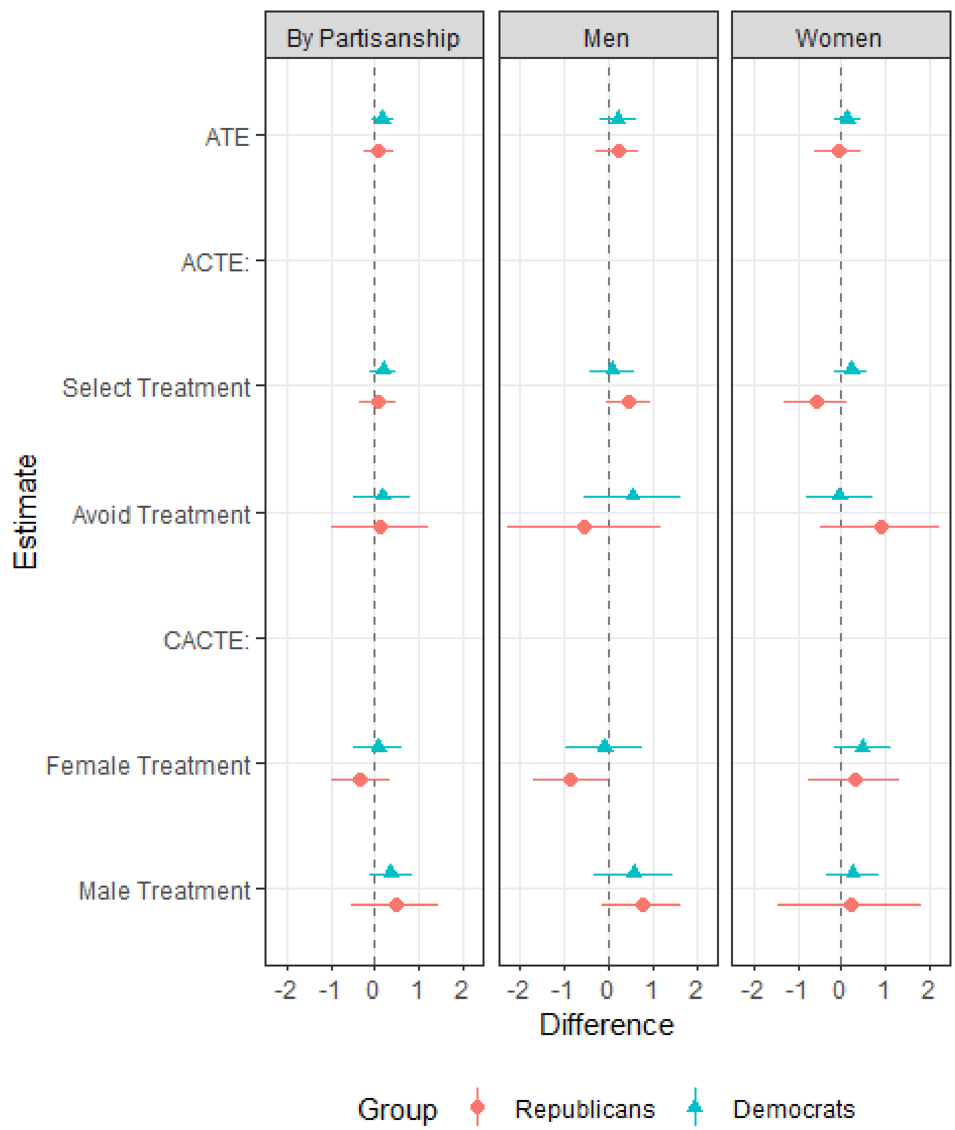
```
# Create Grouping Label

table_app_fn(df_mtg, "dv_pca_metoo",
         "Partisanship",
         Partisanship,
         the_cap ="Treatment Effect Estimates on Specific Support for
\\#MeToo Conditional On Partisanship and Gender (MTurk Sample)" )%>%
  footnote(general = "The table provides point estimates and 95% confidence
intervals for treatment effect estimated by partisanship overall and
```

```
partisanship by gender",
           threeparttable = T,
           fixed_small_size = T)
```

Figure and Table F.4: Treatment Effect Estimates on Specific Support for #MeToo Conditional On Partisanship and Gender (Qualtrics Sample)

```
figF4_df <- rbind(
  data.frame(
  effects_fn(df_qg[df_qg$pid>4,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "By Partisanship"),
  data.frame(
  effects_fn(df_qg[df_qg$pid<4,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "By Partisanship"),
  data.frame(
  effects_fn(df_qg[df_qg$pid>4 & df_qg$gender==0,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "Men"),
  data.frame(
  effects_fn(df_qg[df_qg$pid<4& df_qg$gender==0,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "Men"),
    data.frame(
  effects_fn(df_qg[df_qg$pid>4 & df_qg$gender==1,], "dv_pca_metoo"),
  Group = "Republicans",
  Type = "Women"),
  data.frame(
  effects_fn(df_qg[df_qg$pid<4& df_qg$gender==1,], "dv_pca_metoo"),
  Group = "Democrats",
  Type = "Women")
  )

figF4_df$Group <- factor(figF4_df$Group,
                         levels = unique(figF4_df$Group) )

figF4 <- figF4_df %>%
  filter(Estimand != "CATE")%>%
  ggplot(aes(Estimate, Difference,col=Group,shape=Group))+
  geom_hline(yintercept = 0,linetype="dashed",alpha=.5)+
  facet_grid(~Type)+
  geom_point(aes(shape=Group),
               position = position_dodge(width = .5),size=2
      )+
  geom_linerange(aes(ymin=ll,ymax=ul),size=.3,
                     position = position_dodge(width = .5))+
  geom_linerange(aes(ymin=ll90,ymax=ul90),size=.6,
                     position = position_dodge(width = .5))+
  coord_flip()+
  theme_bw()+
```

```
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "bottom"
  )

figF4
```



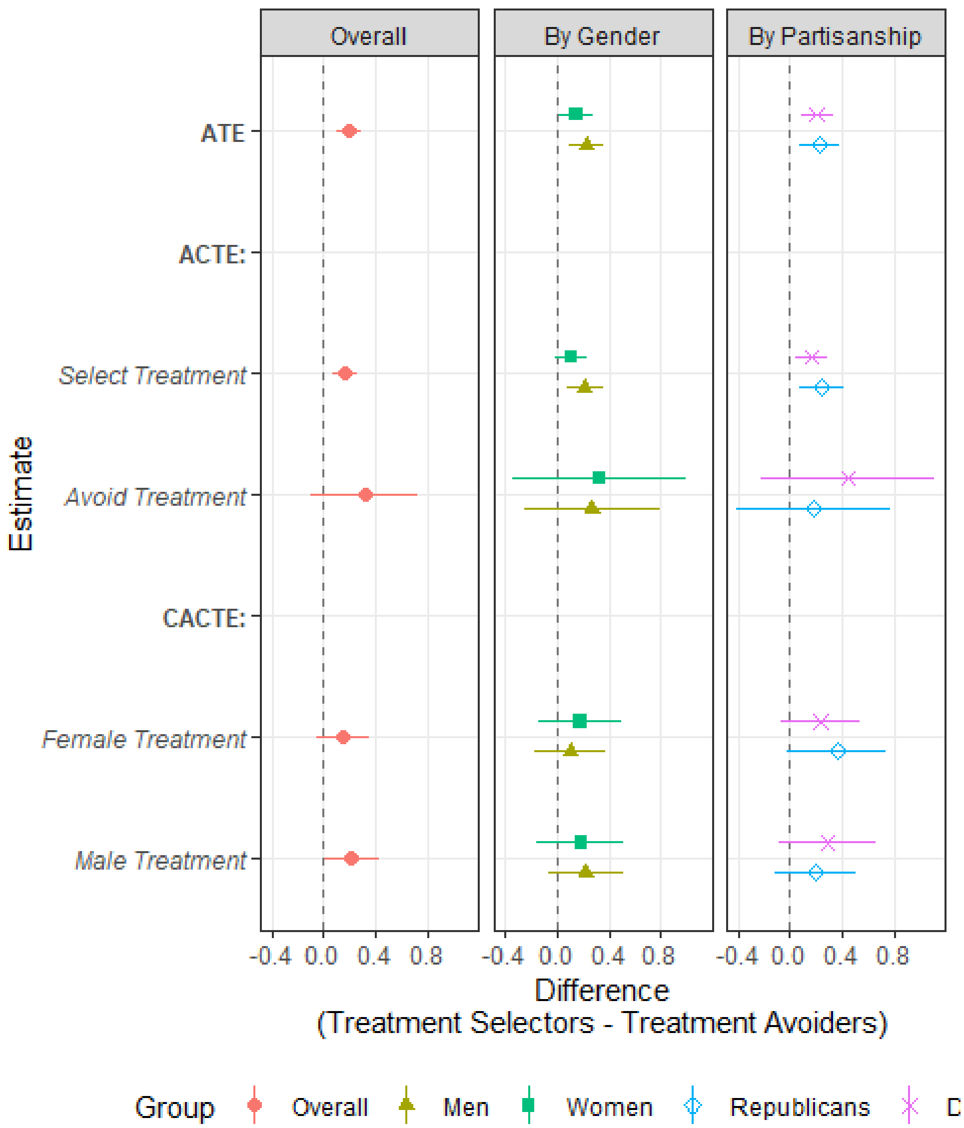
Treatment Effect Estimates on Specific Support for #MeToo Conditional On Partisanship and Gender (Qualtrics Sample)

```
# Create Grouping Label
table_app_fn(df_qg, "dv_pca_metoo",
         "Partisanship",
         Partisanship,
         the_cap ="Treatment Effect Estimates on Specific Support for
\\#MeToo Conditional On Partisanship and Gender (Qulatrics Sample)" )%>%
```

```
  footnote(general = "The table provides point estimates and 95% confidence
intervals for treatment effect estimated by partisanship overall and
partisanship by gender",
           threeparttable = T,
           fixed_small_size = T)
```

Figure and Table F.5: Treatment Effect Estimates on Knowledge of Sexual Assault Statistics (MTurk Sample)

```
# Create Figure 5
figF5 <- plot_effects_fn(df_mtg,"dv_fact01")

# Display Figure 5
figF5
```

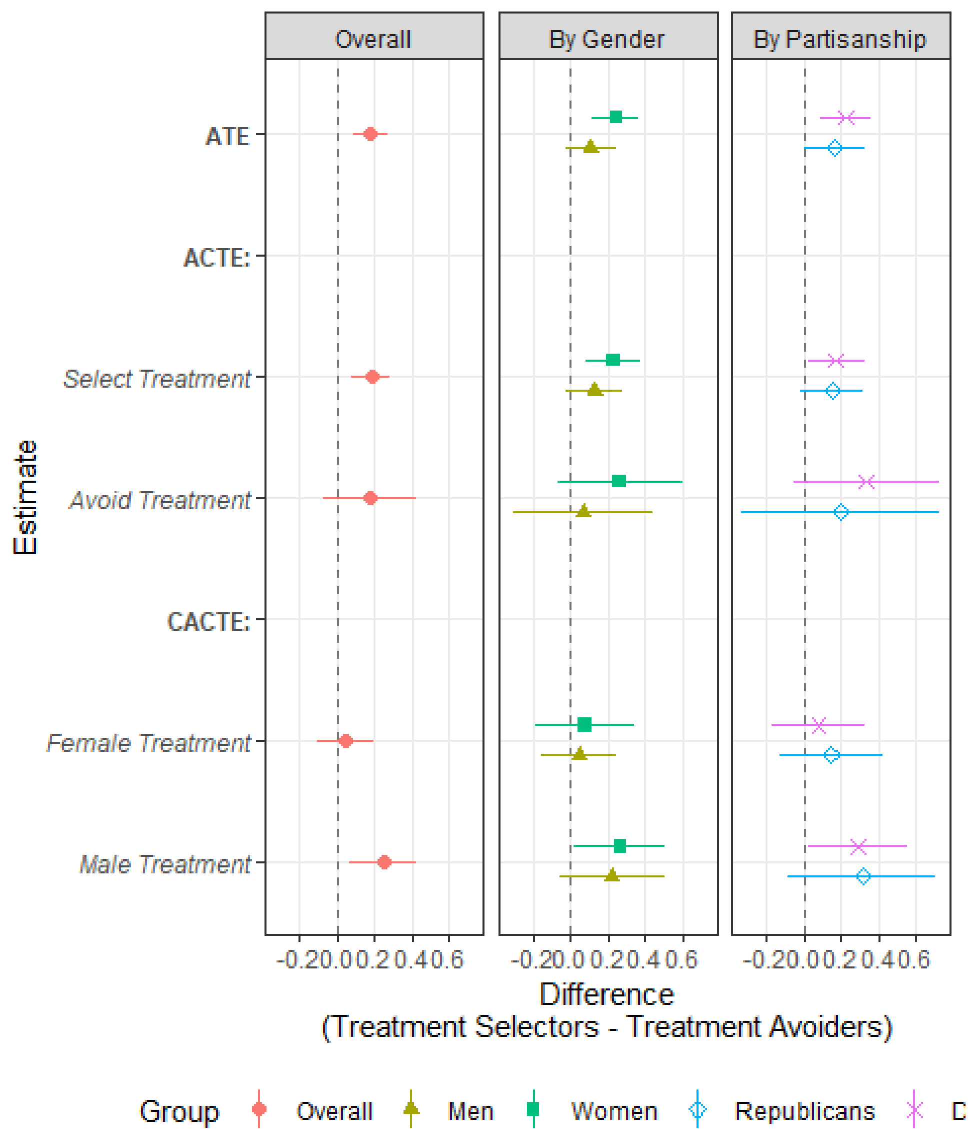


Treatment Effect Estimates on Knowledge of Sexual Assault Statistics (MTurk Sample)

```
table_fn(figF5$data,
    "Treatment Effect Estimates on Knowledge of Sexual Assault Statistics
(MTurk Sample)")
```

Figure and Table F.6: Treatment Effect Estimates on Knowledge of Sexual Assault Statistics (Qualtrics Sample)

```
# Create Figure 6
figF6 <- plot_effects_fn(df_qg,"dv_fact01")

# Display Figure 6
figF6
```

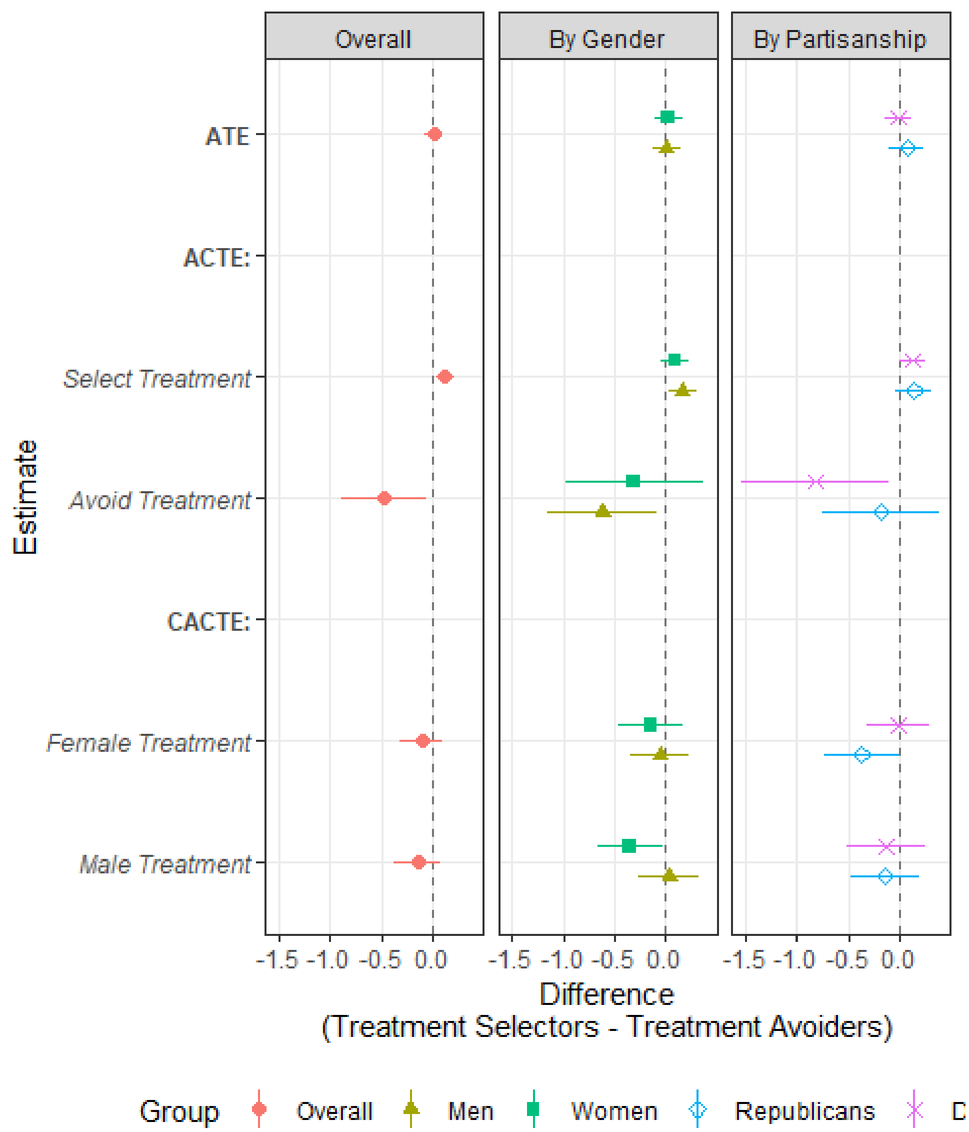


Treatment Effect Estimates on Knowledge of Sexual Assault Statistics (Qualtrics Sample)

```
table_fn(figF6$data,
    "Treatment Effect Estimates on Knowledge of Sexual Assault Statistics
(Qualtrics Sample)")
```

Figure and Table F.7: Treatment Effect Estimates on Providing Written Responses about #MeToo (MTurk Sample)

```
# Create Figure 6
figF8 <- plot_effects_fn(df_mtg,"dv_response01")

# Display Figure 6
figF8
```

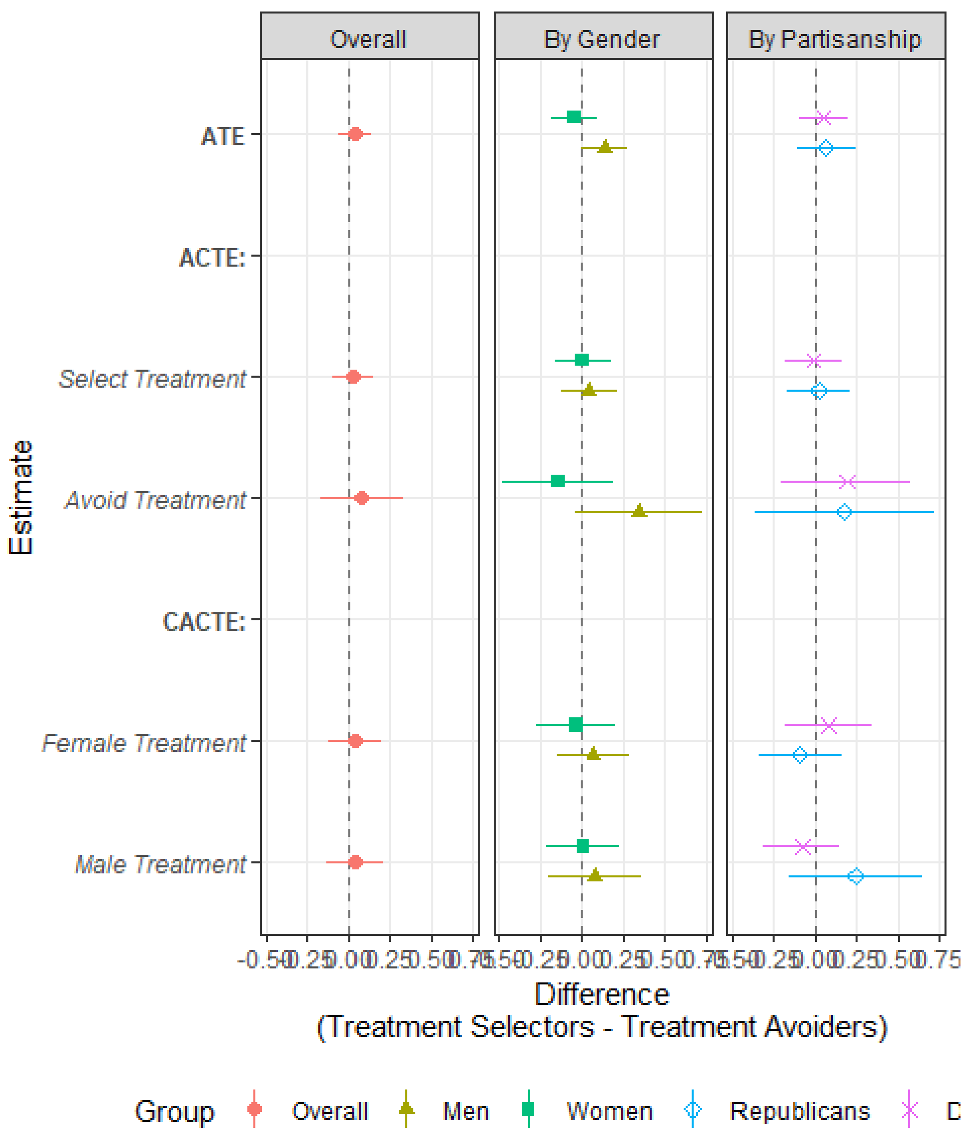


Treatment Effect Estimates on Providing Written Responses about #MeToo (MTurk Sample)

```
table_fn(figF8$data,
    "Treatment Effect Estimates on Providing Written Responses about \\#MeToo
(MTurk Sample)")
```

Figure and Table F.8: Treatment Effect Estimates on Providing Written Responses about #MeToo (Qualtrics Sample)

```
# Create Figure 6
figF8 <- plot_effects_fn(df_qg,"dv_response01")

# Display Figure 6
figF8
```



Treatment Effect Estimates on Providing Written Responses about #MeToo (Qualtrics Sample)

```
table_fn(figF8$data,
    "Treatment Effect Estimates on Providing Written Responses about \\#MeToo
(Qualtrics Sample)")
```