Study 3

April 06, 2025

Items

Read Data

```
# Set this to TRUE if you have API access, FALSE if using CSV
USE_API <- TRUE

if(USE_API) {
  ## Pull directly from Qualtrics API
  qual_data <- fetch_survey(surveyID='SV_8uMcrtMJwiIR6EC',
                    label = T,
                    convert = F,
                    start_date = "2023-08-11",
                    force_request = T)
} else {
  # Read the processed data directly from CSV
  d0 <- read.csv('Study3.csv', check.names = F)
  num_excluded <- unique(d0$num_excluded_total)
}

# Define the categories
men <- c("Larry Page (Co-founder of Google)", "Mark Zuckerberg (Co-founder of Facebook)",
↪  "Daymond John (Founder of FUBU)", "Marvin Ellison (CEO of Lowe's)", "Tim Cook (CEO of
↪  Apple)", "Sean Combs (Founder of Bad Boy Entertainment)", "Bill Gates (Co-founder of
↪  Microsoft)", "Garrett Camp (Co-founder of Uber)", "Charles Koch (CEO of Koch
↪  Industries)", "James Gorman (CEO of Morgan Stanley)", "Jeff Bezos (Founder of
↪  Amazon)", "Jeff Weiner (CEO of LinkedIn)", "Michael Bloomberg (Co-founder of
↪  Bloomberg LP)", "Phil Knight (Co-founder of Nike)", "Sergey Brin (Co-founder of
↪  Google)", "Stewart Butterfield (CEO of Slack)", "Warren Buffet (CEO of Berkshire
↪  Hathaway)", "Sundar Pichai (CEO of Alphabet Inc.)", "James Dimon (CEO of JP Morgan)")

women <- c("Jane Fraser (CEO of Citigroup)", "Oprah Winfrey (CEO of Oprah Winfrey
↪  Network)", "Delphine Arnault (CEO of Christian Dior)", "Michelle Buck (CEO of The
↪  Hershey Company)", "Mary Barra (CEO of General Motors)", "Rosalind Brewer (CEO of
↪  Walgreens)", "Anne Wojcicki (CEO of 23andMe)", "Arianna Huffington (Co-founder of
↪  Huffington Post)", "Karen Lynch (CEO of CVS Health)", "Tricia Griffith (CEO of
↪  Progressive)", "Tory Burch (Founder of Tory Burch)", "Carol Tome (CEO of UPS)", "Leah
↪  Busque (Founder of TaskRabbit)", "Whitney Wolfe Herd (Founder of Bumble)", "Corie
↪  Barry (CEO of Best Buy)", "Melanie Perkins (Founder of Canva)", "Kathy Warden (CEO of
↪  Northrupp Grumman)", "Julia Hartz (Founder of EventBrite)", "Safra Katz (CEO of
↪  Oracle)")

technologists <- c('Stewart Butterfield (CEO of Slack)', 'Bill Gates (Co-founder of
↪  Microsoft)', 'Jeff Bezos (Founder of Amazon)', 'Jeff Weiner (CEO of LinkedIn)',
↪  'Larry Page (Co-founder of Google)', 'Mark Zuckerberg (Co-founder of Facebook)',
↪  'Sergey Brin (Co-founder of Google)', 'Tim Cook (CEO of Apple)', 'Garrett Camp
↪  (Co-founder of Uber)', 'Anne Wojcicki (CEO of 23andMe)', 'Leah Busque (Founder of
↪  TaskRabbit)', 'Whitney Wolfe Herd (Founder of Bumble)', 'Corie Barry (CEO of Best
↪  Buy)', 'Melanie Perkins (Founder of Canva)', 'Kathy Warden (CEO of Northrupp
↪  Grumman)', 'Safra Katz (CEO of Oracle)', 'Sundar Pichai (CEO of Alphabet Inc.)')

founders <- c('Sergey Brin (Co-founder of Google)', "Arianna Huffington (Co-founder of
↪  Huffington Post)", 'Jeff Bezos (Founder of Amazon)', 'Bill Gates (Co-founder of
↪  Microsoft)', 'Larry Page (Co-founder of Google)', 'Daymond John (Founder of FUBU)',
↪  'Mark Zuckerberg (Co-founder of Facebook)', 'Michael Bloomberg (Co-founder of
↪  Bloomberg LP)', 'Phil Knight (Co-founder of Nike)', 'Sean Combs (Founder of Bad Boy
↪  Entertainment)', 'Garrett Camp (Co-founder of Uber)', 'Tory Burch (Founder of Tory
↪  Burch)', 'Leah Busque (Founder of TaskRabbit)', 'Melanie Perkins (Founder of Canva)',
↪  'Julia Hartz (Founder of EventBrite)', "Whitney Wolfe Herd (Founder of Bumble)")
```

```r
women_set <- set_names(set_names(women, women), "1")
men_set <- set_names(set_names(men, men), "0")

if(USE_API) {
  d0 <- qual_data |>
    filter(!is.na(`choice-7`), !is.na(workerId), Finished==1) |>
    mutate(
      female_pick = case_when(`choice-7` %in% women ~ 1,
                              TRUE ~ 0),
           tech_pick = case_when(`choice-7` %in% technologists ~ 1,
                              TRUE ~ 0),
           founder_pick = case_when(`choice-7` %in% founders ~ 1,
                              TRUE ~ 0),
           gender_feedback = ifelse(str_detect(feedback_categories, fixed("Women")), 1,
           ↪  0),
           sample_list = str_split(sample, pattern = "\\|\\|"),
      founder_comp = map_dbl(sample_list, ~ mean(. %in% founders)),
      tech_comp = map_dbl(sample_list, ~ mean(. %in% technologists)),
      base_gender = rowSums(across(`choice-1`:`choice-6`, ~ . %in% women)),
      male_pool = case_when(pool == 'men' ~ 1,
                        TRUE ~ 0),
      gender_code = case_when(gender=="Man" ~ 1, TRUE ~ 0),
      race_code = case_when(race=="White / Caucasian" ~ 1, TRUE ~ 0)
    ) |>
    mutate(across(starts_with("choice") & !ends_with("7"),
                          ~ifelse(.x %in% women_set, 1, 0))) |>
    dplyr::select(female_pick:gender_feedback, founder_comp:male_pool,
↪   `choice-1`:`choice-7`, race, gender, age, gender_code, race_code) |>
    slice(1:1000) # pre-registered sample size

  # Calculate the number of excluded participants
  num_excluded <- nrow(qual_data) - nrow(d0)

  # Save num_excluded in d0
  d0$num_excluded_total <- num_excluded  # As a column

  # Write the API-pulled data into a CSV file
  write.csv(d0, 'Study3.csv', row.names = FALSE, quote = TRUE)
}

# Create the pool-specific dataframes
d0_male_pool <- d0 |>
  filter(male_pool==1)

d0_female_pool <- d0 |>
  filter(male_pool==0)
```

Variable Names

Variable	Description
gender_feedback	Binary indicator of whether a participant was randomly assigned to gender feedback condition.
female_pick	Binary indicator of whether a participant selected a female panelist for their seventh selection.
tech_pick	Binary indicator of whether a participant selected a technologist for their seventh selection.
founder_pick	Binary indicator of whether a participant selected a founder for their seventh selection.
founder_comp	Percentage of Founders in the base rate.
tech_comp	Percentage of Technologists in the base rate.
male_pool	Binary indicator of whether a participant was randomly assigned to male-dominated pool.
choice-1 to choice-7	The selected panelists
gender	Self-selected gender.
race	Self-selected race.
age	Self-entered age.
gender_code	Dummy code for gender (male = 1).
race_code	Dummy code for race (white = 1).

Demographics

```
## Excluded Participants: 126
```

```
##                         Percentage gender
## 1 Another gender not listed here:    0.2
## 2                               Man   50.9
## 3                        Non-binary    0.6
## 4                             Woman   48.3
```

```
##                           Percentage Race
## 1 American Indian or Alaskan Native  1.1
## 2          Asian / Pacific Islander  7.9
## 3         Black or African American  9.1
## 4                 Hispanic / Latinx  2.9
## 5                 White / Caucasian 79.0
```

```
## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     43.3   12.6
```

```
## Mean (num of initial women selected):  1.4
```

```
## SD (num of initial women selected):  1.24
```

```
## Percentage (initial women selected):  0.2333333
```

```
## SD (initial women selected):  0.2066667
```

```
## # A tibble: 2 x 2
##   gender_feedback  mean
##             <dbl> <dbl>
## 1               0 0.242
## 2               1 0.226
```

```
##
##  Welch Two Sample t-test
##
## data:  base_gender/6 by gender_feedback
## t = 0.88271, df = 497.98, p-value = 0.3778
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.01996204  0.05253206
## sample estimates:
## mean in group 0 mean in group 1
##       0.2422402       0.2259552
```

```
## Mean (num of initial women selected):  3.52
```

```
## SD (num of initial women selected):  1.27
```

```
## Percentage (initial women selected):  0.5866667

## SD (initial women selected):  0.2116667

## # A tibble: 2 x 2
##   gender_feedback  mean
##             <dbl> <dbl>
## 1               0 0.598
## 2               1 0.576

##
##  Welch Two Sample t-test
##
## data:  base_gender/6 by gender_feedback
## t = 1.1597, df = 497.96, p-value = 0.2467
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.01525303  0.05920040
## sample estimates:
## mean in group 0 mean in group 1
##       0.5976096       0.5756359
```

Primary Analysis

Women underrepresented Pool

```
# primary model, no encouragement
r_man <- lm(female_pick ~ gender_feedback, data=d0_male_pool)

# Display the summary with robust standard errors
robust_summary(r_man)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback, data = d0_male_pool)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5375 -0.3360 -0.3360  0.4625  0.6640
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.33603    0.03018  11.135  < 2e-16 ***
## gender_feedback  0.20152    0.04360   4.622 4.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4868 on 498 degrees of freedom
## Multiple R-squared:  0.04124,    Adjusted R-squared:  0.03931
## F-statistic: 21.42 on 1 and 498 DF,  p-value: 4.713e-06
```

```
robust_confint(r_man)
```

```
##                     2.5 %    97.5 %
## (Intercept)     0.2767423 0.3953225
## gender_feedback 0.1158526 0.2871814
```

Men underrepresented Pool

```
# primary model, no encouragement
r_woman <- lm(female_pick ~ gender_feedback, data=d0_female_pool)

# Display the summary with robust standard errors
robust_summary(r_woman)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback, data = d0_female_pool)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7291 -0.6185  0.2709  0.3815  0.3815
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.72908    0.02816  25.887  < 2e-16 ***
## gender_feedback -0.11061    0.04182  -2.645  0.00842 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4664 on 498 degrees of freedom
## Multiple R-squared:  0.01392,    Adjusted R-squared:  0.01194
## F-statistic:  7.03 on 1 and 498 DF,  p-value: 0.008271
```

```
robust_confint(r_woman)
```

```
##                      2.5 %      97.5 %
## (Intercept)      0.6737476  0.78441969
## gender_feedback -0.1927666 -0.02845297
```

Interaction by Pool

```
# primary model, no encouragement
r3 <- lm(female_pick ~ gender_feedback*male_pool, data=d0)

# Display the summary with robust standard errors
robust_summary(r3)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback * male_pool, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7291 -0.5375  0.2709  0.3815  0.6640
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.72908    0.02816  25.887  < 2e-16 ***
## gender_feedback           -0.11061    0.04182  -2.645  0.00829 **
## male_pool                 -0.39305    0.04128  -9.522  < 2e-16 ***
## gender_feedback:male_pool  0.31213    0.06041   5.167 2.88e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4767 on 996 degrees of freedom
## Multiple R-squared:  0.08316,    Adjusted R-squared:  0.0804
## F-statistic: 30.11 on 3 and 996 DF,  p-value: < 2.2e-16
```

```
robust_confint(r3)
```

```
##                               2.5 %      97.5 %
## (Intercept)                0.6738150  0.78435237
## gender_feedback           -0.1926666 -0.02855292
## male_pool                 -0.4740537 -0.31204882
## gender_feedback:male_pool  0.1935778  0.43067582
```

Robustness

```
## robust to demographic controls
### when women are underrepresented

r3 <- lm(female_pick ~ gender_feedback + gender_code + race_code + age,
↪  data=d0_male_pool)

# Display the robust_summary with robust standard errors
robust_summary(r3)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + gender_code + race_code +
##     age, data = d0_male_pool)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5860 -0.3784 -0.3163  0.4792  0.6926
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.3889032  0.0913795   4.256 2.49e-05 ***
## gender_feedback  0.2029675  0.0440425   4.608 5.17e-06 ***
## gender_code     -0.0264340  0.0442683  -0.597    0.551
## race_code       -0.0344376  0.0554936  -0.621    0.535
## age             -0.0002793  0.0018123  -0.154    0.878
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4878 on 495 degrees of freedom
## Multiple R-squared:  0.04291,    Adjusted R-squared:  0.03517
## F-statistic: 5.548 on 4 and 495 DF,  p-value: 0.0002245
```

```
robust_confint(r3)
```

```
##                       2.5 %      97.5 %
## (Intercept)      0.209363720 0.568442608
## gender_feedback  0.116434155 0.289500837
## gender_code     -0.113410941 0.060542978
## race_code       -0.143469672 0.074594446
## age             -0.003840005 0.003281428
```

```
## logistic regression
# Fit the logistic regression model
r4 <- glm(female_pick ~ gender_feedback, family = binomial, data=d0_male_pool)

# Odds ratio
tidy_r4 <- tidy(r4, exponentiate = TRUE, conf.int = T)
print(tidy_r4)
```

```
## # A tibble: 2 x 7
##   term            estimate std.error statistic     p.value conf.low conf.high
##   <chr>              <dbl>     <dbl>     <dbl>       <dbl>    <dbl>     <dbl>
## 1 (Intercept)        0.506     0.135     -5.06 0.000000429    0.387     0.657
## 2 gender_feedback    2.30      0.185      4.51 0.00000659     1.60      3.31
```

```
summary(r4)
```

```
##
## Call:
## glm(formula = female_pick ~ gender_feedback, family = binomial,
##     data = d0_male_pool)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.6810     0.1347  -5.056 4.29e-07 ***
## gender_feedback   0.8315     0.1845   4.506 6.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 685.44  on 499  degrees of freedom
## Residual deviance: 664.66  on 498  degrees of freedom
## AIC: 668.66
##
## Number of Fisher Scoring iterations: 4
```

```
## robust to demographic controls
### when women are overrepresented

r5 <- lm(female_pick ~ gender_feedback + gender_code + race_code + age,
↪  data=d0_female_pool)

# Display the robust_summary with robust standard errors
robust_summary(r5)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + gender_code + race_code +
##     age, data = d0_female_pool)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7877 -0.6141  0.2575  0.3510  0.4890
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.722981   0.091418   7.908 1.71e-14 ***
## gender_feedback -0.111257   0.041987  -2.650  0.00831 **
## gender_code     -0.029459   0.042758  -0.689  0.49117
## race_code        0.100341   0.053700   1.869  0.06228 .
```

```
## age             -0.001370   0.001715  -0.799  0.42480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4657 on 495 degrees of freedom
## Multiple R-squared:  0.02275,    Adjusted R-squared:  0.01485
## F-statistic:  2.88 on 4 and 495 DF,  p-value: 0.0223
```

```
robust_confint(r5)
```

```
##                       2.5 %       97.5 %
## (Intercept)      0.543365403  0.902597014
## gender_feedback -0.193751731 -0.028761432
## gender_code     -0.113467244  0.054550010
## race_code       -0.005166878  0.205849478
## age             -0.004740774  0.002000215
```

```
## logistic regression
# Fit the logistic regression model
r6 <- glm(female_pick ~ gender_feedback, family = binomial, data=d0_female_pool)

# Odds ratio
tidy_r6 <- tidy(r6, exponentiate = TRUE, conf.int = T)
print(tidy_r6)
```

```
## # A tibble: 2 x 7
##   term            estimate std.error statistic  p.value conf.low conf.high
##   <chr>              <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)        2.69      0.142      6.97 3.16e-12    2.05      3.58
## 2 gender_feedback    0.602     0.193     -2.63 8.58e- 3    0.412     0.878
```

```
summary(r6)
```

```
##
## Call:
## glm(formula = female_pick ~ gender_feedback, family = binomial,
##     data = d0_female_pool)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.9900     0.1420   6.971 3.16e-12 ***
## gender_feedback  -0.5069     0.1928  -2.629  0.00858 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 631.31  on 499  degrees of freedom
## Residual deviance: 624.33  on 498  degrees of freedom
## AIC: 628.33
##
## Number of Fisher Scoring iterations: 4
```

```r
### interaction model

r7 <- lm(female_pick ~ gender_feedback*male_pool + gender_code + race_code + age,
↪  data=d0)

# Display the robust_summary with robust standard errors
robust_summary(r7)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback * male_pool + gender_code +
##     race_code + age, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7663 -0.5255  0.2635  0.3988  0.7126
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.7562307  0.0668830  11.307  < 2e-16 ***
## gender_feedback            -0.1118210  0.0418681  -2.671  0.00769 **
## male_pool                  -0.3920465  0.0413892  -9.472  < 2e-16 ***
## gender_code                -0.0293015  0.0306656  -0.956  0.33955
## race_code                   0.0337741  0.0386178   0.875  0.38202
## age                        -0.0009134  0.0012402  -0.736  0.46160
## gender_feedback:male_pool  0.3149640  0.0605210   5.204 2.37e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4769 on 993 degrees of freedom
## Multiple R-squared:  0.08499,    Adjusted R-squared:  0.07947
## F-statistic: 15.37 on 6 and 993 DF,  p-value: < 2.2e-16
```

```r
robust_confint(r7)
```

```
##                                2.5 %       97.5 %
## (Intercept)                0.624982457  0.887478988
## gender_feedback           -0.193981069 -0.029660966
## male_pool                 -0.473266843 -0.310826154
## gender_code               -0.089478191  0.030875276
## race_code                 -0.042007889  0.109556020
## age                       -0.003347248  0.001520373
## gender_feedback:male_pool  0.196200295  0.433727691
```

```r
## logistic regression
# Fit the logistic regression model
r8 <- glm(female_pick ~ gender_feedback*male_pool, family = binomial, data=d0)

# Odds ratio
tidy_r8 <- tidy(r8, exponentiate = TRUE, conf.int = T)
print(tidy_r8)
```

```
## # A tibble: 4 x 7
##   term                  estimate std.error statistic  p.value conf.low conf.high
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)              2.69      0.142      6.97 3.16e-12    2.05      3.58
## 2 gender_feedback          0.602     0.193     -2.63 8.58e- 3    0.412     0.878
## 3 male_pool                0.188     0.196     -8.54 1.38e-17    0.127     0.275
## 4 gender_feedback:male~    3.81      0.267      5.01 5.31e- 7    2.26      6.45
```

```
summary(r8)
```

```
##
## Call:
## glm(formula = female_pick ~ gender_feedback * male_pool, family = binomial,
##     data = d0)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.9900     0.1420   6.971 3.16e-12 ***
## gender_feedback            -0.5069     0.1928  -2.629  0.00858 **
## male_pool                  -1.6710     0.1957  -8.537  < 2e-16 ***
## gender_feedback:male_pool   1.3384     0.2669   5.015 5.31e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1373.7  on 999  degrees of freedom
## Residual deviance: 1289.0  on 996  degrees of freedom
## AIC: 1297
##
## Number of Fisher Scoring iterations: 4
```

Figure 4 Code

```
## Man pool
dman_plot <- d0 |>
  filter(male_pool == 1) |>
  dplyr::select(gender_feedback, female_pick) |>
  group_by(gender_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(female_pick),
    sd = sd(female_pick) * 100,
    se = (sd(female_pick) / sqrt(n())) * 100
  ) |>
  mutate(gender_feedback = case_when(gender_feedback==1 ~ "Treatment",
                          TRUE ~ "Control")) |>
  rename(Condition = gender_feedback)

## Women Pool
dwoman_plot <- d0 |>
  filter(male_pool == 0) |>
  dplyr::select(gender_feedback, female_pick) |>
  group_by(gender_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(female_pick),
    sd = sd(female_pick) * 100,
    se = (sd(female_pick) / sqrt(n())) * 100
  ) |>
  mutate(gender_feedback = case_when(gender_feedback==1 ~ "Treatment",
                          TRUE ~ "Control")) |>
  rename(Condition = gender_feedback)


df_combined <- bind_rows(
  dman_plot %>% mutate(Category = "Women Underrepresented\nin Panelist Candidate Set"),
  dwoman_plot %>% mutate(Category = "Women Overrepresented\nin Panelist Candidate Set")
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('Women Underrepresented\nin Panelist
  ↪  Candidate Set', 'Women Overrepresented\nin Panelist Candidate Set')))

df_link <- df_combined %>%
  filter(Condition == "Treatment" & Category == "Women Underrepresented\nin Panelist
  ↪  Candidate Set" |
           Condition == "Control" & Category == "Women Overrepresented\nin Panelist
↪  Candidate Set")

p_combined <- ggplot(df_combined, aes(x = Condition, y = freq*100, fill = Condition)) +
  geom_bar(stat="identity", width = 0.85, position = position_dodge(width = 0.7)) +
  geom_text(aes(label=paste0(sprintf("%.1f", freq*100),"%")),
            position=position_dodge(width=0.7), vjust=5, size = 7, color = "white",
            ↪  family = "Times New Roman") +
  geom_errorbar(aes(ymin=freq*100-se, ymax=freq*100+se), width = .1, position =
  ↪  position_dodge(width = 0.7)) +
```

```
facet_wrap(~factor(Category, c('Women Underrepresented\nin Panelist Candidate Set',
↪  'Women Overrepresented\nin Panelist Candidate Set')), nrow = 1, strip.position =
↪  "bottom") +
geom_segment(data = df_combined %>% filter(Category == "Women Underrepresented\nin
↪  Panelist Candidate Set" & Condition == "Treatment"),
             aes(x = 1, xend = 2, y = freq*100 + se + 10, yend = freq*100 + se + 10),
             inherit.aes = FALSE) +
  geom_segment(data = df_combined %>% filter(Category == "Women Overrepresented\nin
  ↪  Panelist Candidate Set" & Condition == "Control"),
             aes(x = 1, xend = 2, y = freq*100 + se + 10, yend = freq*100 + se + 10),
             inherit.aes = FALSE) +
  geom_text(data = df_combined %>% filter(Category == "Women Underrepresented\nin
  ↪  Panelist Candidate Set" & Condition == "Treatment"),
             aes(x = 1.5, xend = 1.5, y = freq*100 + se + 10, yend = freq*100 + se +
             ↪  10, label = "***", size = 7),
             inherit.aes = FALSE, vjust = 0, family = "Times New Roman", size = 7) +
  geom_text(data = df_combined %>% filter(Category == "Women Overrepresented\nin
  ↪  Panelist Candidate Set" & Condition == "Control"),
             aes(x = 1.5, xend = 1.5, y = freq*100 + se + 10, yend = freq*100 + se +
             ↪  10, label = "***"),
             inherit.aes = FALSE, vjust = 0, family = "Times New Roman", size = 7) +
theme_bw() +
scale_fill_manual(values = c("#990000", "#011F5B"), labels = c("No gender feedback
↪  provided", "Gender feedback provided"), "") +
scale_y_continuous(labels = function(x) paste0(x,"%"), limits = c(0,100)) +
scale_x_discrete(labels = c("\"Control\"" = "Not\nShown", "\"Treatment\"" = "Shown")) +
labs(x = "", y = "% of Final Panelists Selected who Were Women") +
theme(text = element_text(family = "Times New Roman"),
      plot.caption = element_text(face = "italic", family = "Times New Roman"),
      legend.position = c(0.5, 0.95),
      legend.direction = "horizontal",
      legend.text = element_text(size = 18, family = "Times New Roman"),
      legend.key.size = unit(5, 'mm'),
      legend.background = element_rect(fill = "white"),
      panel.grid.minor = element_blank(),
      panel.grid = element_blank(),
      panel.border = element_rect(fill= NA, color = "white"),
      plot.background = element_rect(fill = "white"),
      panel.background = element_rect(fill = "white"),
      axis.title.x = element_text(size = 18, color = "black", family = "Times New
      ↪  Roman"),
      plot.title = element_text(hjust = 0.5, family = "Times New Roman"),
      axis.title.y = element_text(size = 18, color = "black", family = "Times New
      ↪  Roman"),
      axis.text.x = element_blank(),
      axis.ticks = element_blank(),
      axis.text.y = element_text(size = 18, color = "black", family = "Times New
      ↪  Roman"),
      strip.text = element_text(size = 18, color = "black", family = "Times New
      ↪  Roman"),
      strip.background = element_rect(colour = "white", fill = "white"))
```

```
# Save the plot with Times New Roman font
# ggsave("../Manuscript_Figures/Figure-4.pdf", plot = p_combined, width = 10, height = 8,
↪  units = "in", device = cairo_pdf, family = "Times New Roman")
```