Study S1

January 28, 2025

Items

Read Data 2
Variable Names 4
Demographics 5
Pooled Analysis 6
Primary Analysis 7
Robustness 8
Secondary Analysis 10
Figure S3 Code 12
System of Simultaneous Equations 15

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_beJTVpRBi2UsN8y',
                     label = T,
                     convert = F,
                     force_request = T)
} else {
  # Read the processed data directly from CSV
  d0 <- read.csv('StudyS1.csv', check.names = F)
}

# Define the categories
females <- c('Mary Barra (CEO of General Motors)', 'Oprah Winfrey (CEO of Oprah Winfrey
↪  Network)',
             'Janet Yellen (U.S. Secretary Treasury)', 'Jane Fraser (CEO of Citigroup)',
             'Rosalind Brewer (CEO of Walgreens)')

ceos <- c("Tim Cook (CEO of Apple)", "Elon Musk (CEO of Tesla)", "Oprah Winfrey (CEO of
↪  Oprah Winfrey Network)",
          "Mary Barra (CEO of General Motors)", "Jane Fraser (CEO of Citigroup)", "Marvin
          ↪  Ellison (CEO of Lowe's)",
          "Warren Buffet (CEO of Berkshire Hathaway)", "Charles Koch (CEO of Koch
          ↪  Industries)",
          "Stewart Butterfield (CEO of Slack)", "Jack Ma (CEO of Alibaba)", "Rosalind
          ↪  Brewer (CEO of Walgreens)",
          "Jeff Weiner (CEO of LinkedIn)")

techs <- c("Bill Gates (Co-founder of Microsoft)", "Mark Zuckerberg (Co-founder of
↪  Facebook)",
           "Tim Cook (CEO of Apple)", "Elon Musk (CEO of Tesla)", "Larry Page (Co-founder
           ↪  of Google)",
           "Sergey Brin (Co-founder of Google)", "Jeff Bezos (Founder of Amazon)",
           "Stewart Butterfield (CEO of Slack)", "Jack Ma (CEO of Alibaba)",
           "Jeff Weiner (CEO of LinkedIn)")

founders <- c("Bill Gates (Co-founder of Microsoft)", "Mark Zuckerberg (Co-founder of
↪  Facebook)",
              "Larry Page (Co-founder of Google)", "Sergey Brin (Co-founder of Google)",
              "Sean Combs (Founder of Bad Boy Entertainment)", "Daymond John (Founder of
              ↪  FUBU)",
              "Jeff Bezos (Founder of Amazon)", "Michael Bloomberg (Co-founder of
              ↪  Bloomberg LP)",
              "Phil Knight (Co-founder of Nike)")

if(USE_API) {
  d0 <- qual_data |>
    mutate(ec_2 = tolower(ec_2)) |>
    filter(workerId!="", selection_6!="", ec_2 %in% c("one one", "\"one one\""),
    ↪  Finished==1)  |>
```

```r
mutate(fem_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
              ~ case_when(. %in% females ~ 1,
                          TRUE ~ 0)),
       ceo_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
              ~ case_when(. %in% ceos ~ 1,
                          TRUE ~ 0)),
        tech_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
              ~ case_when(. %in% techs ~ 1,
                TRUE ~ 0)),
        founder_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
              ~ case_when(. %in% founders ~ 1,
                TRUE ~ 0)),
       condition = case_when(group %in% c("control1", "control2") ~ "Control",
                             TRUE ~ "Treatment")) |>
mutate(fem_score = case_when(`female_count_type-1` == "are female" ~ female_count_1,
                          `female_count_type-2` == "are female" ~ female_count_2,
                          `female_count_type-3` == "are female" ~ female_count_3),
       gender_feedback = case_when(condition=="Control" ~ 0, condition=="Treatment" ~
       ↪   1),
       founder = case_when((group == "control1" & (`count_type-1` == "are founders" |
       ↪   `count_type-2` == "are founders")) | (group=="control2") |
       ↪   (group=="treatment" & (`female_count_type-1` == "are founders" |
       ↪   `female_count_type-2` == "are founders" | `female_count_type-3` == "are
       ↪   founders")) ~ 1, TRUE ~ 0),
       tech = case_when((group == "control1" & (`count_type-1` == "are in the
       ↪   technology industry" | `count_type-2` == "are in the technology
       ↪   industry")) | (group=="control2") | (group=="treatment" &
       ↪   (`female_count_type-1` == "are in the technology industry" |
       ↪   `female_count_type-2` == "are in the technology industry" |
       ↪   `female_count_type-3` == "are in the technology industry")) ~ 1, TRUE ~
       ↪   0),
       ceo = case_when((group == "control1" & (`count_type-1` == "are CEOs" |
       ↪   `count_type-2` == "are CEOs")) | (group=="control2") | (group=="treatment"
       ↪   & (`female_count_type-1` == "are CEOs" | `female_count_type-2` == "are
       ↪   CEOs" | `female_count_type-3` == "are CEOs")) ~ 1, TRUE ~ 0),
       list_two = case_when(group=="control1" ~ 1, group=="control2" ~ 0, TRUE ~
       ↪   NA_real_)) |>
mutate(female = case_when(fem_choice$bonus_ctrl2_7==1 | fem_choice$bonus_ctrl1_7==1 |
↪   fem_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
       ceo_pick = case_when(ceo_choice$bonus_ctrl2_7==1 | ceo_choice$bonus_ctrl1_7==1
       ↪   |  ceo_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
       founder_pick = case_when(founder_choice$bonus_ctrl2_7==1 |
       ↪   founder_choice$bonus_ctrl1_7==1 |  founder_choice$bonus_trt_7==1 ~ 1, TRUE
       ↪   ~ 0),
       tech_pick = case_when(tech_choice$bonus_ctrl2_7==1 |
       ↪   tech_choice$bonus_ctrl1_7==1 |  tech_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
       gender_code = case_when(gender=="Man" ~ 1, TRUE ~ 0),
       race_code = case_when(race=="White / Caucasian" ~ 1, TRUE ~ 0),
       age = as.numeric(age),
       gender = case_when(gender=="" ~ "N/A",
                          TRUE ~ gender),
       race = case_when(race=="" ~ "N/A",
                        TRUE ~ race),
           base_gender = rowSums(across(selection_1:selection_6, ~ . %in% females))) |>
    dplyr::select(
      list_two, gender_feedback, female, ceo, ceo_pick, founder, founder_pick, tech,
↪   tech_pick, base_gender, gender, race, age, gender_code, race_code
    )

  # Write the API-pulled data into a CSV file
  write.csv(d0, 'StudyS1.csv', row.names = FALSE, quote = TRUE)
}
```

Variable Names

Variable	Description
list_two	Binary indicator of whether the control received a list of two attributes (list_two=1) or not (list_two=0).
gender_feedback	Binary indicator of whether a participant was randomly assigned to gender feedback condition.
female	Binary indicator of whether a participant selected a female business leader for their seventh selection.
ceo	Binary indicator of whether a participant was randomly assigned to receive CEO feedback.
ceo_pick	Binary indicator of whether a participant selected a CEO business leader for their seventh selection.
founder	Binary indicator of whether a participant was randomly assigned to receive founder feedback.
founder_pick	Binary indicator of whether a participant selected a founder business leader for their seventh selection.
tech	Binary indicator of whether a participant was randomly assigned to receive technologist feedback.
tech_pick	Binary indicator of whether a participant selected a technologist business leader for their seventh selection.
base_gender	Count of the number of female panelists selected in the initial six selections.
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: 161
```

```
##                        Percentage gender
## 1 Another gender not listed here:    0.2
## 2                             Man   50.9
## 3                      Non-binary    1.0
## 4                           Woman   47.9
```

```
##                         Percentage Race
## 1 American Indian or Alaskan Native  0.3
## 2          Asian / Pacific Islander  7.6
## 3         Black or African American  7.9
## 4                 Hispanic / Latinx  7.1
## 5                 White / Caucasian 77.1
```

```
## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     41.2   12.0
```

Pooled Analysis

```
r0 <- lm(female ~ list_two, data=d0)

# Calculate robust standard errors
robust_summary(r0)
```

```
##
## Call:
## lm(formula = female ~ list_two, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2857 -0.2857 -0.2691  0.7143  0.7309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.28571    0.02857  10.000   <2e-16 ***
## list_two    -0.01664    0.04016  -0.414    0.679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4486 on 499 degrees of freedom
##   (499 observations deleted due to missingness)
## Multiple R-squared:  0.0003452,  Adjusted R-squared:  -0.001658
## F-statistic: 0.1723 on 1 and 499 DF,  p-value: 0.6782
```

Primary Analysis

```
# primary model
r1 <- lm(female ~ gender_feedback, data=d0)


# Display the summary with robust standard errors
robust_summary(r1)
```

```
##
## Call:
## lm(formula = female ~ gender_feedback, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5010 -0.2774 -0.2774  0.4990  0.7226
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.27745    0.02004  13.842  < 2e-16 ***
## gender_feedback  0.22356    0.03008   7.432 2.29e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.475 on 998 degrees of freedom
## Multiple R-squared:  0.05257,    Adjusted R-squared:  0.05162
## F-statistic: 55.37 on 1 and 998 DF,  p-value: 2.147e-13
```

Robustness

```
## which feedback was shown with gender
r2 <- lm(female ~ gender_feedback + ceo + tech + founder, data=d0)

# Display the summary with robust standard errors
robust_summary(r2)
```

```
##
## Call:
## lm(formula = female ~ gender_feedback + ceo + tech + founder,
##     data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5445 -0.3153 -0.2776  0.4932  0.7851
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.236400   0.102156   2.314   0.0209 *
## gender_feedback  0.229217   0.036083   6.353 3.22e-10 ***
## ceo              0.070785   0.046672   1.517   0.1297
## tech             0.008146   0.047623   0.171   0.8642
## founder         -0.029617   0.047594  -0.622   0.5339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4743 on 995 degrees of freedom
## Multiple R-squared:  0.05818,    Adjusted R-squared:  0.05439
## F-statistic: 15.37 on 4 and 995 DF,  p-value: 3.357e-12
```

```
robust_confint(r2)
```

```
##                        2.5 %     97.5 %
## (Intercept)      0.03593469 0.43686486
## gender_feedback  0.15841068 0.30002398
## ceo             -0.02080162 0.16237168
## tech            -0.08530769 0.10159988
## founder         -0.12301310 0.06377988
```

```
## robust to demographic controls

r3 <- lm(female ~ gender_feedback + gender_code + race_code + age, data=d0)

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

```
##
## Call:
## lm(formula = female ~ gender_feedback + gender_code + race_code +
##     age, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5586 -0.3394 -0.2652  0.5005  0.7889
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.367078   0.063217   5.807 8.57e-09 ***
## gender_feedback  0.223391   0.030199   7.397 2.95e-13 ***
## gender_code     -0.021694   0.030438  -0.713    0.476
## race_code       -0.033666   0.036171  -0.931    0.352
## age             -0.001274   0.001292  -0.986    0.324
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4751 on 995 degrees of freedom
## Multiple R-squared:  0.05504,    Adjusted R-squared:  0.05124
## F-statistic: 14.49 on 4 and 995 DF,  p-value: 1.662e-11
```

```
robust_confint(r3)
```

```
##                         2.5 %     97.5 %
## (Intercept)      0.243025270 0.49113170
## gender_feedback  0.164128762 0.28265227
## gender_code     -0.081423582 0.03803557
## race_code       -0.104645350 0.03731367
## age             -0.003809364 0.00126140
```

```
## logistic regression
# Fit the logistic regression model
r4 <- glm(female ~ gender_feedback, family = binomial, data=d0)

# 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)        0.384    0.0998     -9.59 8.60e-22    0.315     0.466
## 2 gender_feedback    2.61     0.134       7.17 7.52e-13    2.01      3.41
```

Secondary Analysis

```
## ceo feedback
r_ceo <- lm(ceo_pick ~ ceo, data=d0)

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

```
##
## Call:
## lm(formula = ceo_pick ~ ceo, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4802 -0.4802 -0.4752  0.5198  0.5248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.475207   0.032235  14.742   <2e-16 ***
## ceo         0.005004   0.037003   0.135    0.892
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5001 on 998 degrees of freedom
## Multiple R-squared:  1.841e-05,  Adjusted R-squared:  -0.0009836
## F-statistic: 0.01837 on 1 and 998 DF,  p-value: 0.8922
```

```
#robust_confint(r_ceo)

## founder feedback
r_founder <- lm(founder_pick ~ founder, data=d0)

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

```
##
## Call:
## lm(formula = founder_pick ~ founder, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2955 -0.2955 -0.2955  0.7045  0.7807
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21933    0.02532   8.661   <2e-16 ***
## founder      0.07615    0.03044   2.501   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4457 on 998 degrees of freedom
## Multiple R-squared:  0.00572,    Adjusted R-squared:  0.004724 
## F-statistic: 5.741 on 1 and 998 DF,  p-value: 0.01675
```

```
#robust_confint(r_founder)


## tech feedback
r_tech <- lm(tech_pick ~ tech, data=d0)

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

```
## 
## Call:
## lm(formula = tech_pick ~ tech, data = d0)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.3132 -0.3132 -0.3132  0.6868  0.7004 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.29958    0.02988  10.026   <2e-16 ***
## tech         0.01366    0.03429   0.398     0.69    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4629 on 998 degrees of freedom
## Multiple R-squared:  0.0001577,  Adjusted R-squared:  -0.0008441 
## F-statistic: 0.1574 on 1 and 998 DF,  p-value: 0.6916
```

```
#robust_confint(r_tech)

## interaction of base gender
# primary model
r_interaction <- lm(female ~ gender_feedback*base_gender, data=d0)


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

```
## 
## Call:
## lm(formula = female ~ gender_feedback * base_gender, data = d0)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.5655 -0.3335 -0.2391  0.4889  0.8081 
## 
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  0.333519   0.030146  11.063  < 2e-16 ***
## gender_feedback              0.232015   0.045619   5.086 4.37e-07 ***
## base_gender                 -0.047215   0.016476  -2.866  0.00425 **
## gender_feedback:base_gender -0.007179   0.026642  -0.269  0.78763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4725 on 996 degrees of freedom
## Multiple R-squared:  0.06429,    Adjusted R-squared:  0.06147
## F-statistic: 22.81 on 3 and 996 DF,  p-value: 2.75e-14
```

Figure S3 Code

```
## dataframe for Gender information

dgender_plot <- d0 |>
  dplyr::select(gender_feedback, female) |>
  group_by(gender_feedback, female) |>
  summarise(n = n()) |>
  mutate(freq = n / sum(n)) |>
  filter(female == 1) |>
  mutate(sd = sqrt((freq*(1-freq))/n)*100,
         se = case_when(gender_feedback==0 ~ coef(summary(r1))[, "Std. Error"][1]*100,
                        TRUE ~ coef(robust_summary(r1))[, "Std. Error"][2]*100)) |>
  mutate(gender_feedback = case_when(gender_feedback==1 ~ "\"Treatment\"",
                          TRUE ~ "\"Control\"")) |>
  rename(Condition = gender_feedback)

## dataframe for CEO information

dceo_plot <- d0 |>
  dplyr::select(ceo, ceo_pick) |>
  group_by(ceo, ceo_pick) |>
  summarise(n = n()) |>
  mutate(freq = n / sum(n)) |>
  filter(ceo_pick == 1) |>
  mutate(sd = sqrt((freq*(1-freq))/n)*100,
         se = case_when(ceo==0 ~ coef(robust_summary(r_ceo))[, "Std. Error"][1]*100,
                        TRUE ~ coef(robust_summary(r_ceo))[, "Std. Error"][2]*100)) |>
  mutate(ceo = case_when(ceo==1 ~ "\"Treatment\"",
                          TRUE ~ "\"Control\"")) |>
  rename(Condition = ceo)

## dataframe for Founder information

dfounder_plot <- d0 |>
  dplyr::select(founder, founder_pick) |>
  group_by(founder, founder_pick) |>
  summarise(n = n()) |>
  mutate(freq = n / sum(n)) |>
  filter(founder_pick == 1) |>
  mutate(sd = sqrt((freq*(1-freq))/n)*100,
         se = case_when(founder==0 ~ coef(robust_summary(r_founder))[, "Std.
         ↪  Error"][1]*100,
                        TRUE ~ coef(robust_summary(r_founder))[, "Std. Error"][2]*100))
                        ↪   |>
  mutate(founder = case_when(founder==1 ~ "\"Treatment\"",
                          TRUE ~ "\"Control\"")) |>
  rename(Condition = founder)

## dataframe for Tech information

dtech_plot <- d0 |>
  dplyr::select(tech, tech_pick) |>
  group_by(tech, tech_pick) |>
  summarise(n = n()) |>
  mutate(freq = n / sum(n)) |>
  filter(tech_pick == 1) |>
  mutate(sd = sqrt((freq*(1-freq))/n)*100,
         se = case_when(tech==0 ~ coef(robust_summary(r_tech))[, "Std. Error"][1]*100,
                        TRUE ~ coef(robust_summary(r_tech))[, "Std. Error"][2]*100)) |>
  mutate(tech = case_when(tech==1 ~ "\"Treatment\"",
                          TRUE ~ "\"Control\"")) |>
  rename(Condition = tech)

df_combined <- bind_rows(
  dceo_plot %>% mutate(Category = "\nCEOs"),
  dtech_plot %>% mutate(Category = "\nTechnologists"),
  dfounder_plot %>% mutate(Category = "\nFounders"),
  dgender_plot %>% mutate(Category = "\nFemale")
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('\nCEOs', '\nTechnologists',
   ↪  '\nFounders', '\nFemale')))

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 = 5, color = "white") +
  geom_errorbar(aes(ymin=freq*100-se, ymax=freq*100+se), width = .1, position =
   ↪  position_dodge(width = 0.7)) +
  facet_wrap(~factor(Category, c('\nCEOs', '\nTechnologists', '\nFounders', '\nFemale')),
   ↪  nrow = 1, strip.position = "bottom") +
  geom_segment(data = df_combined %>% filter(Condition == "\"Treatment\""),
               aes(x = 1, xend = 2, y = freq*100 + se + 5, yend = freq*100 + se + 5),
               inherit.aes = FALSE) +
  geom_text(data = df_combined %>% filter(Category %in% c('\nCEOs', '\nTechnologists') &
   ↪  Condition == "\"Treatment\""),
            aes(x = 1.5, xend = 1.5, y = freq*100 + se + 7, yend = freq*100 + se + 7,
             ↪  label = "n.s."),
            inherit.aes = FALSE, vjust = 0, size = 5) +
    geom_text(data = df_combined %>% filter(Category %in% c('\nFounders') & Condition ==
     ↪  "\"Treatment\""),
            aes(x = 1.5, xend = 1.5, y = freq*100 + se + 5, yend = freq*100 + se + 5,
             ↪  label = "*"),
            inherit.aes = FALSE, vjust = 0, size = 5) +
  geom_text(data = df_combined %>% filter(Category == '\nFemale' & Condition ==
   ↪  "\"Treatment\""),
            aes(x = 1.5, xend = 1.5, y = freq*100 + se + 5, yend = freq*100 + se + 5,
            ↪  label = "***"),
            inherit.aes = FALSE, vjust = 0, size = 5) +
  theme_bw() +
  scale_fill_manual(values = c("#990000", "#011F5B"), labels = c("No feedback provided",
   ↪  "Feedback provided"), "Feedback") +
  scale_y_continuous(labels = function(x) paste0(x,"%"), limits = c(0,80)) +
  scale_x_discrete(labels = c("\"Control\"" = "Not\nShown", "\"Treatment\"" = "Shown")) +
  labs(x = "Feedback on % of panelists who were...", y = "% of New Panelists with the
   ↪  Target Identity",
       title = "The Effect of Getting Feedback on Your Panel's Composition") +
  theme(plot.caption = element_text(face = "italic"),
        legend.position = c(0.5, 0.85),
        legend.title = element_blank(),
        legend.direction = "horizontal",
        legend.text = element_text(size = 20, family = "Times New Roman"),
        legend.key.size = unit(7, '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(face="bold", size = 22, vjust = 13, family = "Times
         ↪  New Roman"),
        plot.title = element_blank(),
        axis.title.y = element_text(size = 20, color = "black", family = "Times New
         ↪  Roman"),
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        axis.text.y = element_text(size = 20, color = "black", family = "Times New
         ↪  Roman"),
        strip.text = element_text(size = 20, color = "black", family = "Times New
         ↪  Roman"),
        strip.background = element_rect(colour = "white", fill = "white"))

# Save the plot with Times New Roman font
# ggsave("../Supplemental_Figures/Figure-S3.pdf", plot = p_combined, width = 10, height =
 ↪  8, units = "in", device = cairo_pdf, family = "Times New Roman")
```

System of Simultaneous Equations

```
##                                    Wald.Coefficient      P_Value
## Gender Feedback - Founder Feedback         57.74091 4.574119e-14
## Gender Feedback - Tech Feedback            78.69334 0.000000e+00
## Gender Feedback - CEO Feedback             25.02350 6.160383e-07
```