Study 2A

January 28, 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_e4eMITeZ8WisKSa',
                    label = T,
                    convert = F,
                    start_date = "2024-01-03",
                    force_request = T)
} else {
  # Read the processed data directly from CSV
  d0 <- read.csv('Study2A.csv', check.names = F)
}

# Define the categories
women <- c('Sylvia Plath', 'Isabel Allende', 'Lucy Maud', 'Jane Austen', 'Joyce Carol
↪  Oates', 'Zadie Smith')
poets <- c('George Orwell','Sylvia Plath','Jack London', 'Charles Dickens', 'Neil
↪  Gaiman', 'Jorge Luis Borges')
oldies <- c('Charles Dickens', 'Ernest Hemingway', 'Herman Melville', 'Jack London',
↪  'Jane Austen', 'Jorge Luis Borges', 'J.R.R. Tolkien', 'Lucy Maud', 'Nathaniel
↪  Hawthorne', 'F. Scott Fitzgerald')
books <- c('Charles Dickens', 'Herman Melville', 'Isabel Allende', 'Jack London', 'John
↪  Steinbeck', 'Joyce Carol Oates', 'Jorge Luis Borges', 'J.R.R. Tolkien', 'Lucy Maud',
↪  'Michael Crichton', 'Ray Bradbury', 'Kurt Vonnegut', 'Ian McEwan')

if(USE_API) {
  # Process the API data
  d0 <- qual_data |>
    filter(!is.na(`choice-7`), !is.na(PROLIFIC_PID), Finished==1) |>
    mutate(
      gender_feedback = as.numeric(grepl("were women", feedbackItem1) |
                    grepl("were women", feedbackItem2) |
                    grepl("were women", feedbackItem3)),
      poets_shown = as.numeric(grepl("wrote poetry", feedbackItem1) |
                    grepl("wrote poetry", feedbackItem2) |
                    grepl("wrote poetry", feedbackItem3)),
      oldies_shown = as.numeric(grepl("were born in the 1800s", feedbackItem1) |
                  grepl("were born in the 1800s", feedbackItem2) |
                  grepl("were born in the 1800s", feedbackItem3)),
      books_shown = as.numeric(grepl("wrote more than 10 books", feedbackItem1) |
                   grepl("wrote more than 10 books", feedbackItem2) |
                   grepl("wrote more than 10 books", feedbackItem3)),
      female_pick = case_when(`choice-7` %in% women ~ 1,
                              TRUE ~ 0),
      poets_pick = case_when(`choice-7` %in% poets ~ 1,
                              TRUE ~ 0),
      oldies_pick = case_when(`choice-7` %in% oldies ~ 1,
                              TRUE ~ 0),
      books_pick = case_when(`choice-7` %in% books ~ 1,
```

```
                              TRUE ~ 0),
      gender_code = case_when(gender=="Man" ~ 1, TRUE ~ 0),
      race_code = case_when(str_detect(race, "White / Caucasian") ~ 1, TRUE ~ 0),
      base_gender = rowSums(across(`choice-1`:`choice-6`, ~ . %in% women))
    ) |>
    mutate(
      across(c(I1:E3),
             ~ case_when(
               . == "Strongly disagree" ~ 1, . == "Disagree" ~ 2, . == "Somewhat
↪  disagree" ~ 3, . == "Neither agree nor disagree" ~ 4,
               . == "Somewhat agree" ~ 5, . == "Agree" ~ 6, . == "Strongly agree" ~ 7,
↪  TRUE ~ NA_integer_))) |>
    mutate(
      internal1Z = (I1 - mean(I1, na.rm = TRUE)) / sd(I1, na.rm = TRUE),
      internal2Z = (I2 - mean(I2, na.rm = TRUE)) / sd(I2, na.rm = TRUE),
      internal3Z = (I3 - mean(I3, na.rm = TRUE)) / sd(I3, na.rm = TRUE),
      internal4Z = (I4 - mean(I4, na.rm = TRUE)) / sd(I4, na.rm = TRUE),
      internal = (internal1Z + internal2Z + internal3Z + internal4Z) / 4,
      external1Z = (E1 - mean(E1, na.rm = TRUE)) / sd(E1, na.rm = TRUE),
      external2Z = (E2 - mean(E2, na.rm = TRUE)) / sd(E2, na.rm = TRUE),
      external3Z = (E3 - mean(E3, na.rm = TRUE)) / sd(E3, na.rm = TRUE),
      external = (external1Z + external2Z + external3Z) / 3
    ) |>
    dplyr::select(gender_feedback:books_pick, base_gender,`choice-1`:`choice-7`, race,
↪  gender, age, gender_code, race_code, internal1Z:external) |>
    slice(1:1000) # pre-registered sample size

  # Write the API-pulled data into a CSV file
  write.csv(d0, 'Study2A.csv', row.names = FALSE, quote = TRUE)
}
```

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 author for their seventh selection
poets_pick	Binary indicator of whether a participant selected an author that wrote poetry for their seventh selection.
oldies_pick	Binary indicator of whether a participant selected an author that was born in the 1800s for their seventh selection.
book_pick	Binary indicator of whether a participant selected an author that wrote more than 10 books for their seventh selection.
base_gender	Count of the number of female authors selected in the initial six authors.
choice-1 to choice-7	The selected authors
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).
internal1Z-4Z	Individual standardized scale items for Internal Motivation to Respond Without Prejudice.
external1Z-3Z	Individual standardized scale items for External Motivation to Respond Without Prejudice.
internal	Aggregated scale items for Internal Motivation to Respond Without Prejudice.
external	Aggregated scale items for External Motivation to Respond Without Prejudice.

Demographics

```
## Excluded Participants: 123
```

```
##                          Percentage gender
## 1 Another gender not listed here:    0.1
## 2                               Man   45.4
## 3                        Non-binary    2.0
## 4                             Woman   52.5
```

```
##                            Percentage Race
## 1 American Indian or Alaskan Native  0.7
## 2           Asian / Pacific Islander  9.8
## 3          Black or African American 13.1
## 4                  Hispanic / Latinx  6.2
## 5                  White / Caucasian 70.2
```

```
## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     41.0   13.7
```

Cronbach's Alpha

```
# Calculating Cronbach's Alpha for the Internal subscale
internal_items <- d0[, c("internal1Z", "internal2Z", "internal3Z", "internal4Z")]
alpha_internal <- alpha(internal_items)

cat("Cronbach's Alpha for Internal Subscale: ", alpha_internal$total$raw_alpha, "\n")
```

```
## Cronbach's Alpha for Internal Subscale:  0.9161604
```

```
# Calculating Cronbach's Alpha for the External subscale
external_items <- d0[, c("external1Z", "external2Z", "external3Z")]
alpha_external <- alpha(external_items)
cat("Cronbach's Alpha for External Subscale: ", alpha_external$total$raw_alpha, "\n")
```

```
## Cronbach's Alpha for External Subscale:  0.8665694
```

Primary Analysis

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

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

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3508 -0.3508 -0.2421  0.6492  0.7579
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.24206    0.01912  12.662  < 2e-16 ***
## gender_feedback  0.10874    0.02875   3.783 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4537 on 998 degrees of freedom
## Multiple R-squared:  0.01419,    Adjusted R-squared:  0.0132
## F-statistic: 14.36 on 1 and 998 DF,  p-value: 0.0001599
```

```r
robust_confint(r1)
```

```
##                      2.5 %    97.5 %
## (Intercept)     0.20454862 0.2795784
## gender_feedback 0.05232815 0.1651578
```

Robustness

```
## which feedback was shown with gender, remove constant due to ollinearity
r2 <- lm(female_pick ~ gender_feedback + oldies_pick + poets_pick + books_pick - 1,
↪  data=d0)

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

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + oldies_pick + poets_pick +
##     books_pick - 1, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4782 -0.2551 -0.1503  0.5970  1.0000
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## gender_feedback 0.252651   0.025239  10.010  < 2e-16 ***
## oldies_pick     0.002419   0.028872   0.084   0.9333
## poets_pick      0.072811   0.033664   2.163   0.0308 *
## books_pick      0.150322   0.024801   6.061 1.92e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4737 on 996 degrees of freedom
## Multiple R-squared:  0.245,  Adjusted R-squared:  0.242
## F-statistic: 80.81 on 4 and 996 DF,  p-value: < 2.2e-16
```

```
robust_confint(r2)
```

```
##                       2.5 %     97.5 %
## gender_feedback  0.203123249 0.30217888
## oldies_pick     -0.054238744 0.05907642
## poets_pick       0.006749703 0.13887260
## books_pick       0.101653669 0.19899098
```

```
## robust to demographic controls

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

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

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + gender_code + race_code +
##     age, data = d0)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4414 -0.3027 -0.2630  0.5836  0.8709
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.329787   0.050668   6.509 1.20e-10 ***
## gender_feedback  0.117171   0.028543   4.105 4.37e-05 ***
## gender_code     -0.121673   0.028313  -4.297 1.90e-05 ***
## race_code        0.018696   0.031895   0.586    0.558
## age             -0.001215   0.001087  -1.118    0.264
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.45 on 995 degrees of freedom
## Multiple R-squared:  0.03324,    Adjusted R-squared:  0.02935
## F-statistic: 8.552 on 4 and 995 DF,  p-value: 8.726e-07
```

```
robust_confint(r3)
```

```
##                       2.5 %        97.5 %
## (Intercept)      0.230358267  0.4292165391
## gender_feedback  0.061158906  0.1731835632
## gender_code     -0.177233248 -0.0661121580
## race_code       -0.043893471  0.0812852028
## age             -0.003347433  0.0009171391
```

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

```
##
## Call:
## glm(formula = female_pick ~ gender_feedback, family = binomial,
##     data = d0)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.1414     0.1040  -10.98  < 2e-16 ***
## gender_feedback   0.5259     0.1402    3.75 0.000177 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1214.9  on 999  degrees of freedom
## Residual deviance: 1200.6  on 998  degrees of freedom
## AIC: 1204.6
##
## Number of Fisher Scoring iterations: 4
```

```r
# 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)        0.319     0.104    -11.0  5.00e-28    0.260     0.390
## 2 gender_feedback    1.69      0.140      3.75 1.77e- 4    1.29      2.23
```

Interaction Analysis

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


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

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback * base_gender, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4145 -0.3373 -0.2216  0.6439  0.8169
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.18307    0.02800   6.538 9.94e-11 ***
## gender_feedback               0.14695    0.04402   3.339 0.000873 ***
## base_gender                   0.03856    0.01547   2.493 0.012823 *
## gender_feedback:base_gender -0.02551    0.02295  -1.111 0.266631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4528 on 996 degrees of freedom
## Multiple R-squared:  0.02024,    Adjusted R-squared:  0.01729
## F-statistic: 6.858 on 3 and 996 DF,  p-value: 0.0001419
```

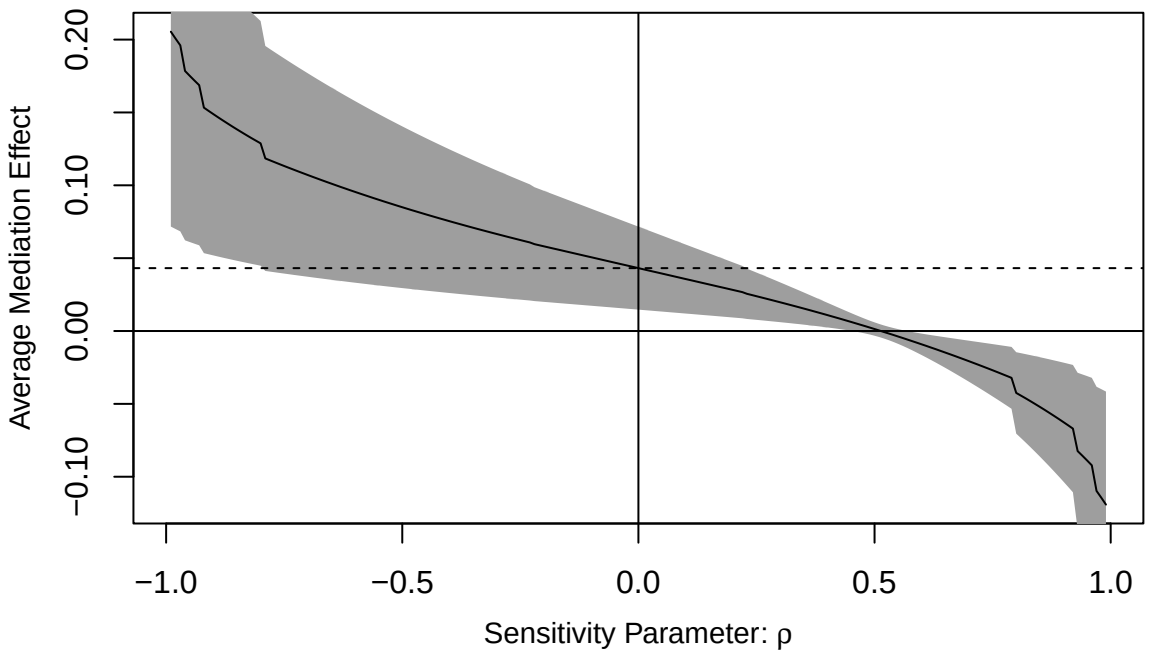
Secondary Analysis

Mediation

```
## Sobel test for Internal Motivation

## $statistic
## internal
## 19.65463
##
## $p_value
## internal
##        0
##
## $se
##   internal
## 0.01295411

##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.0432       0.0151         0.07  0.0016 **
## ADE              0.0656       0.0164         0.11  0.0098 **
## Total Effect     0.1087       0.0530         0.17  <2e-16 ***
## Prop. Mediated   0.3969       0.1648         0.75  0.0016 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

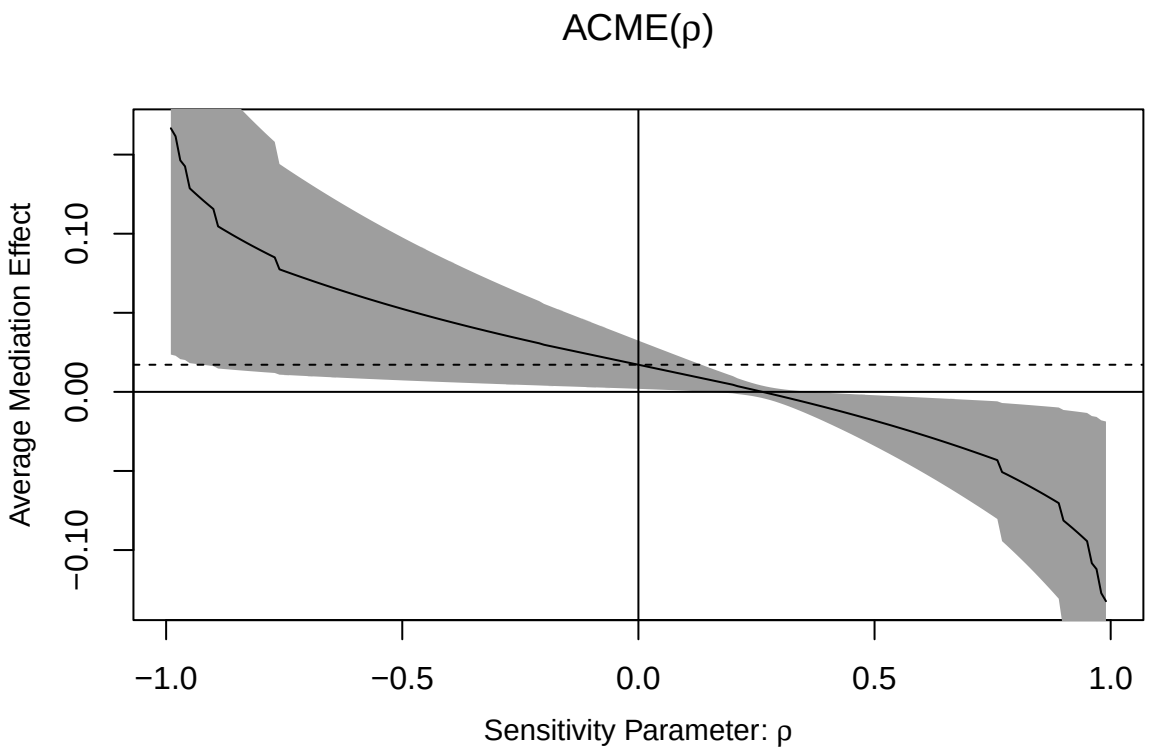



```
## Sobel test for External Motivation

## $statistic
## external
## 8.512787
##
## $p_value
## external
##        0
##
## $se
##  external
## 0.0157396

##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME            0.01715      0.00208         0.03  0.0248 *
## ADE             0.09159      0.03863         0.15  0.0012 **
## Total Effect    0.10874      0.05363         0.16  0.0002 ***
## Prop. Mediated  0.15774      0.02205         0.36  0.0246 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
```

```
##
## Simulations: 10000
```




```
## Correlation Between Internal and External:  0.7463554

## P-value:  1.152678e-178

## Combined Multiple Mediation Model Results

##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.0591       0.0200         0.10  0.0034 **
## ADE              0.0659       0.0174         0.11  0.0084 **
## Total Effect     0.1250       0.0632         0.19  0.0002 ***
## Prop. Mediated   0.4728       0.2118         0.79  0.0032 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000

##
```

```
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME           -0.01622     -0.03234         0.00  0.0204 *
## ADE             0.06588      0.01841         0.11  0.0076 **
## Total Effect    0.04965     -0.00102         0.10  0.0560 .
## Prop. Mediated -0.32672     -2.59288         0.92  0.0764 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

Figure S1 Code

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


##### poets


r_poets <- lm(poets_pick ~ poets_shown, data=d0)


dpoets_plot <- d0 |>
  dplyr::select(poets_shown, poets_pick) |>
  dplyr::group_by(poets_shown) |>
  dplyr::summarise(
    n = n(),
    freq = mean(poets_pick),
    sd = sd(poets_pick) * 100,
    se = (sd(poets_pick) / sqrt(n())) * 100
  )|>
```

```
  mutate(poets_shown = case_when(poets_shown==1 ~ "\"Treatment\"",
                           TRUE ~ "\"Control\"")) |>
  rename(Condition = poets_shown)


##### books

r_books <- lm(books_pick ~ books_shown, data=d0)

dbooks_plot <- d0 |>
  dplyr::select(books_shown, books_pick) |>
  dplyr::group_by(books_shown) |>
  dplyr::summarise(
    n = n(),
    freq = mean(books_pick),
    sd = sd(books_pick) * 100,
    se = (sd(books_pick) / sqrt(n())) * 100
  )|>
  mutate(books_shown = case_when(books_shown==1 ~ "\"Treatment\"",
                           TRUE ~ "\"Control\"")) |>
  rename(Condition = books_shown)


#### oldies

r_oldies <- lm(oldies_pick ~ oldies_shown, data=d0)


doldies_plot <- d0 |>
  dplyr::select(oldies_shown, oldies_pick) |>
  dplyr::group_by(oldies_shown) |>
  dplyr::summarise(
    n = n(),
    freq = mean(oldies_pick),
    sd = sd(oldies_pick) * 100,
    se = (sd(oldies_pick) / sqrt(n())) * 100
  )|>
  mutate(oldies_shown = case_when(oldies_shown==1 ~ "\"Treatment\"",
                           TRUE ~ "\"Control\"")) |>
  rename(Condition = oldies_shown)

## Combine plots

df_combined <- bind_rows(
  dpoets_plot %>% mutate(Category = "\nWrote Poetry"),
  dbooks_plot %>% mutate(Category = "\nWrote > 10 books"),
  doldies_plot %>% mutate(Category = "\nWere Born\nin the 1800s"),
  dfemale_plot %>% mutate(Category = "\nWere Women")
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('\nWrote Poetry', '\nWrote > 10 books',
  ↪  '\nWere Born\nin the 1800s', '\nWere Women')))

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('\nWrote Poetry', '\nWrote > 10 books', '\nWere Born\nin
  ↪  the 1800s', '\nWere Women')), 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('\nWrote Poetry', '\nWrote >
   ↪  10 books', '\nWere Born\nin the 1800s') & 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 == '\nWere Women' & 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(0,90)) +
  scale_x_discrete(labels = c("\"Control\"" = "Not\nShown", "\"Treatment\"" = "Shown")) +
  labs(x = "Feedback on % of authors who...", y = "% of New Authors with the Target
  ↪  Attribute",title = "The Effect of Getting Feedback on Your Author Selections") +
  theme(plot.caption = element_text(face = "italic"),
        legend.position = c(0.5, 0.95),
        legend.title = element_blank(),
        legend.direction = "horizontal",
        legend.text = element_text(size = 20),
        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 = 21, vjust = 17),
        plot.title = element_blank(),
        axis.title.y = element_text(size = 20, color = "black"),
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        axis.text.y = element_text(size = 20, color = "black"),
        strip.text = element_text(size = 20, color = "black"),
        strip.background = element_rect(colour = "white", fill = "white"))

#p_combined
# Save the plot with Times New Roman font
ggsave("../Supplemental Studies/Supplemental_Figures/Figure-S1.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 - poets Feedback       45.46436853 2.029532e-11
## Gender Feedback - books Feedback        0.05131848 8.208079e-01
## Gender Feedback - oldies Feedback      18.51765959 1.764927e-05
```