# Study 3B

October 03, 2024

## Items

## Read Data

```
## Pull directly from Qualtrics API
qual_data <- fetch_survey(surveyID='SV_9KBd2ktMonQbXWS',
                  label = T,
                  convert = F,
                  start_date = "2022-10-14",
                  force_request = T)
```

```
books_list <- c("Agatha Christie", "Alice Walker", "Charles Dickens", "Herman Melville",
↪  "Isabel Allende", "Jack London", "John Steinbeck", "Joyce Carol Oates", "Jorge Luis
↪  Borges", "JRR Tolkien", "Louisa May Alcott", "Lucy Maud", "Michael Crichton", "Sandra
↪  Cisneros", "Toni Morrison")

oldies_list <- c("Agatha Christie", "Charles Dickens", "Emily Bronte", "Ernest
↪  Hemingway", "Herman Melville", "Jack London", "Jane Austen", "Jorge Luis Borges",
↪  "JRR Tolkien", "Louisa May Alcott", "Lucy Maud", "Nathaniel Hawthorne", "WEB Du
↪  Bois")

poets_list <- c("Alice Walker", "Charles Dickens", "Emily Bronte", "George Orwell", "Jack
↪  London", "James Baldwin", "Joyce Carol Oates", "Jorge Luis Borges", "Louisa May
↪  Alcott", "Lucy Maud", "Sandra Cisneros", "Sylvia Plath", "Toni Morrison")

race_list <- c("Alice Walker", "Gabriel Garcia Marquez", "Isabel Allende", "James
↪  Baldwin", "Sandra Cisneros", "Toni Morrison", "WEB Du Bois", "Jorge Luis Borges")


d0 <- qual_data |>
  mutate(ec_2 = tolower(ec_2)) |>
  filter(workerId!="", selection_6 != "", ec_2 %in% c("one one", "\"one one\""),
   ↪  bonus_ctrl2_7 != "" |
                bonus_ctrl1_7 != "" |
                bonus_trt_7 != "", Finished==1) |>
  dplyr::select(c(selection_1:last_col()))  |>
  # Replace NA with an empty string in relevant columns
  mutate(across(starts_with("race"), ~replace_na(., ""), .names = "new_{.col}")) %>%
  # Combine race columns into a single column
  unite("race_combined", starts_with("new_race"), sep = ", ", na.rm = TRUE, remove =
   ↪  TRUE) %>%
  # Clean up the race column
  mutate(race = sapply(strsplit(race_combined, ", "), function(x) {
    cleaned_entries = x[!str_detect(x, "^\\s*$")]
    paste(cleaned_entries, collapse = ", ")
  }))  |>
  mutate(race_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
                ~ case_when(. %in% race_list ~ 1,
                            TRUE ~ 0)),
         book_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
                ~ case_when(. %in% books_list  ~ 1,
                            TRUE ~ 0)),
                oldies_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
                ~ case_when(. %in% oldies_list ~ 1,
```

```
                    TRUE ~ 0)),
                    poets_choice = across(c(bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_7),
                ~ case_when(. %in% poets_list ~ 1,
                    TRUE ~ 0)),
        race_feedback = case_when(group %in% c("control1", "control2") ~ 0,
                                  TRUE ~ 1)) |>
 mutate(race_pick = case_when(race_choice$bonus_ctrl2_7==1 |
 ↪  race_choice$bonus_ctrl1_7==1 |  race_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
        poets_pick = case_when(poets_choice$bonus_ctrl2_7==1 |
        ↪  poets_choice$bonus_ctrl1_7==1 |  poets_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
        oldies_pick = case_when(oldies_choice$bonus_ctrl2_7==1 |
        ↪  oldies_choice$bonus_ctrl1_7==1 |  oldies_choice$bonus_trt_7==1 ~ 1, TRUE ~
        ↪  0),
        book_pick = case_when(book_choice$bonus_ctrl2_7==1 |
        ↪  book_choice$bonus_ctrl1_7==1 |  book_choice$bonus_trt_7==1 ~ 1, TRUE ~ 0),
        poets = case_when((group == "control1" & (`count_type-1` == "wrote poetry" |
        ↪  `count_type-2` == "wrote poetry")) | (group=="control2") |
        ↪  (group=="treatment" & (`minority_count_type-1` == "wrote poetry" |
        ↪  `minority_count_type-2` == "wrote poetry" | `minority_count_type-3` ==
        ↪  "wrote poetry")) ~ 1, TRUE ~ 0),
        oldies = case_when((group == "control1" & (`count_type-1` == "were born in the
        ↪  1800s" | `count_type-2` == "were born in the 1800s")) | (group=="control2")
        ↪  | (group=="treatment" & (`minority_count_type-1` == "were born in the 1800s"
        ↪  | `minority_count_type-2` == "were born in the 1800s" |
        ↪  `minority_count_type-3` == "were born in the 1800s")) ~ 1, TRUE ~ 0),
        books = case_when((group == "control1" & (`count_type-1` == "wrote more than 10
        ↪  books" | `count_type-2` == "wrote more than 10 books")) |
        ↪  (group=="control2") | (group=="treatment" & (`minority_count_type-1` ==
        ↪  "wrote more than 10 books" | `minority_count_type-2` == "wrote more than 10
        ↪  books" | `minority_count_type-3` == "wrote more than 10 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),
        age = as.numeric(age),
        list_two = case_when(group=="control1" ~ 1, group=="control2" ~ 0, TRUE ~
        ↪  NA_real_)) |>
 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,
```

```
    base_race = rowSums(across(selection_1:selection_6, ~ . %in% race_list))
  ) |>
  dplyr::select(list_two, race_feedback, race_pick:books, base_race,
↪   selection_1:selection_6,bonus_ctrl2_7, bonus_ctrl1_7, bonus_trt_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, 'Study3B.csv', row.names = FALSE, quote = TRUE)

##############################################################################################
# when reading the csv, use the following command: read.csv('Study3B.csv', check.names =
↪   F)
##############################################################################################
```

## 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). |
| race_feedback | Binary indicator of whether a participant was randomly assigned to race feedback condition. |
| race_pick | Binary indicator of whether a participant selected a racial minority protagonist for their seventh author 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 1800*s* 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_race | Count of the number of racial minority authors selected in the initial six authors. |
| selection_1 to selection_6 | The selected authors |
| bonus_ctrl1_7, bonus_ctrl2_7 | The final selected author for control |
| bonus_trt_7 | The final selected author for treatment |
| 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: 138
```

```
##                          Percentage gender
## 1 Another gender not listed here:    0.2
## 2                               Man   47.6
## 3                        Non-binary    0.7
## 4                             Woman   51.5
```

```
##                                Race Percentage
## 1 American Indian or Alaskan Native       1.45
## 2          Asian / Pacific Islander       8.83
## 3         Black or African American       9.21
## 4                 Hispanic / Latinx       4.75
## 5                 White / Caucasian      75.75
```

```
## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     42.5   12.7
```

### Cronbach's Alpha

```r
# 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:  0.9349041
```

```r
# 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:  0.8995114
```

### Pooled Analysis

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

# Display the summary with robust standard errors
robust_summary(r0)
```

```
##
## Call:
## lm(formula = race_pick ~ list_two, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2910 -0.2910 -0.2734  0.7090  0.7266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.27344    0.02797   9.777   <2e-16 ***
## list_two     0.01755    0.04043   0.434    0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4508 on 498 degrees of freedom
##   (500 observations deleted due to missingness)
## Multiple R-squared:  0.0003799,  Adjusted R-squared:  -0.001627
## F-statistic: 0.1893 on 1 and 498 DF,  p-value: 0.6637
```

### Primary Analysis

```
# primary model
r1 <- lm(race_pick ~ race_feedback, data=d0)
summary(r1)
```

```
##
## Call:
## lm(formula = race_pick ~ race_feedback, data = d0)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.416 -0.416 -0.282  0.584  0.718
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.28200    0.02113  13.348  < 2e-16 ***
## race_feedback  0.13400    0.02988   4.485 8.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4724 on 998 degrees of freedom
```

```
## Multiple R-squared:  0.01976,    Adjusted R-squared:  0.01878
## F-statistic: 20.12 on 1 and 998 DF,  p-value: 8.135e-06
```

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

```
##
## Call:
## lm(formula = race_pick ~ race_feedback, data = d0)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.416 -0.416 -0.282  0.584  0.718
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.28200    0.02016  13.985  < 2e-16 ***
## race_feedback  0.13400    0.02991   4.481  8.3e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4724 on 998 degrees of freedom
## Multiple R-squared:  0.01976,    Adjusted R-squared:  0.01878
## F-statistic: 20.12 on 1 and 998 DF,  p-value: 8.135e-06
```

```
robust_confint(r1)
```

```
##                    2.5 %    97.5 %
## (Intercept)   0.24243180 0.3215682
## race_feedback 0.07531266 0.1926873
```

## Robustness

```
## which feedback was shown with gender, remove constant due to collinearity
r2 <- lm(race_pick ~ race_feedback + oldies_pick + poets_pick + book_pick - 1, data=d0)


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

```
##
## Call:
## lm(formula = race_pick ~ race_feedback + oldies_pick + poets_pick +
##     book_pick - 1, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7306 -0.2526  0.0000  0.2694  1.2283
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## race_feedback  0.24974    0.02608   9.577   <2e-16 ***
## oldies_pick   -0.22825    0.02344  -9.738   <2e-16 ***
## poets_pick     0.32392    0.02425  13.356   <2e-16 ***
## book_pick      0.15694    0.02395   6.554    9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4393 on 996 degrees of freedom
## Multiple R-squared:  0.4493, Adjusted R-squared:  0.4471
## F-statistic: 203.1 on 4 and 996 DF,  p-value: < 2.2e-16
```

```
## robust to demographic controls

r3 <- lm(race_pick~ race_feedback + gender_code + race_code + age, data=d0)

# robust standard errors
robust_summary(r3)
```

```
##
## Call:
## lm(formula = race_pick ~ race_feedback + gender_code + race_code +
##     age, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5289 -0.3624 -0.2584  0.5597  0.8435
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.448277   0.059960   7.476 1.67e-13 ***
## race_feedback  0.134679   0.029791   4.521 6.90e-06 ***
```

```
## gender_code   -0.103115   0.029754  -3.466 0.000552 ***
## race_code     -0.016756   0.037118  -0.451 0.651787
## age           -0.002456   0.001148  -2.139 0.032674 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4694 on 995 degrees of freedom
## Multiple R-squared:  0.03512,    Adjusted R-squared:  0.03124
## F-statistic: 9.053 on 4 and 995 DF,  p-value: 3.491e-07
```

```
robust_confint(r3)
```

```
##                      2.5 %        97.5 %
## (Intercept)    0.330615178  0.5659397546
## race_feedback  0.076218684  0.1931394470
## gender_code   -0.161502152 -0.0447274505
## race_code     -0.089594893  0.0560831820
## age           -0.004708967 -0.0002028828
```

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

```
##
## Call:
## glm(formula = race_pick ~ race_feedback, family = binomial, data = d0)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.93456    0.09939  -9.403  < 2e-16 ***
## race_feedback  0.59535    0.13457   4.424 9.69e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1293.7  on 999  degrees of freedom
## Residual deviance: 1273.8  on 998  degrees of freedom
## AIC: 1277.8
##
## Number of Fisher Scoring iterations: 4
```

```
# 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)      0.393    0.0994     -9.40 5.29e-21    0.322     0.476
## 2 race_feedback    1.81     0.135       4.42 9.69e- 6    1.39      2.36
```

### Interaction Analysis

```r
## interaction of base gender
# primary model
r_interaction <- lm(race_pick ~ race_feedback*base_race, data=d0)


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

```
##
## Call:
## lm(formula = race_pick ~ race_feedback * base_race, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4883 -0.4153 -0.2716  0.5838  0.7718
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.22825    0.02640   8.645  < 2e-16 ***
## race_feedback             0.18830    0.04091   4.603 4.71e-06 ***
## base_race                 0.04334    0.01603   2.703  0.00698 **
## race_feedback:base_race -0.04375    0.02274  -1.924  0.05468 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4712 on 996 degrees of freedom
## Multiple R-squared:  0.02647,    Adjusted R-squared:  0.02353
## F-statistic: 9.026 on 3 and 996 DF,  p-value: 6.711e-06
```

## Mediation

```r
# Set seed for reproducibility
set.seed(123)

# Define function for Sobel Test
sobel_test <- function(med.fit, out.fit, mediator) {
  med.se <- sqrt(diag(vcovHC(med.fit)))[mediator]
  out.se <- sqrt(diag(vcovHC(out.fit)))[mediator]
  sobel_test_statistic <- coef(out.fit)[mediator] / sqrt(vcovHC(out.fit)[mediator,
↪  mediator])
  sobel_p_value <- 2 * (1 - pnorm(abs(sobel_test_statistic)))
  list(statistic = sobel_test_statistic, p_value = sobel_p_value, se = out.se)
}

# ------------------------------
# Internal Motivation Analysis
# ------------------------------

# Direct effect model
dir.fit.internal <- lm(race_pick ~ race_feedback, data=d0)

# Mediator model
med.fit.internal <- lm(internal ~ race_feedback, data = d0)

# Outcome model including mediator
out.fit.internal <- lm(race_pick ~ race_feedback + internal, data = d0)

# Mediation analysis
med.out.internal <- mediate(med.fit.internal, out.fit.internal, boot = TRUE,
                            treat = "race_feedback", boot.ci.type = "perc", mediator =
                            ↪  "internal", sims = 10000)

# Sensitivity analysis
sens.out.internal <- medsens(med.out.internal, rho.by = 0.01, eps=.01, effect.type =
↪  "indirect", sims = 10000)

# Sobel test for internal motivation
sobel.internal <- sobel_test(med.fit.internal, out.fit.internal, "internal")

# Print and visualize results for internal motivation
cat("Sobel test for Internal Motivation\n")
```

```
## Sobel test for Internal Motivation
```

```r
print(sobel.internal)
```

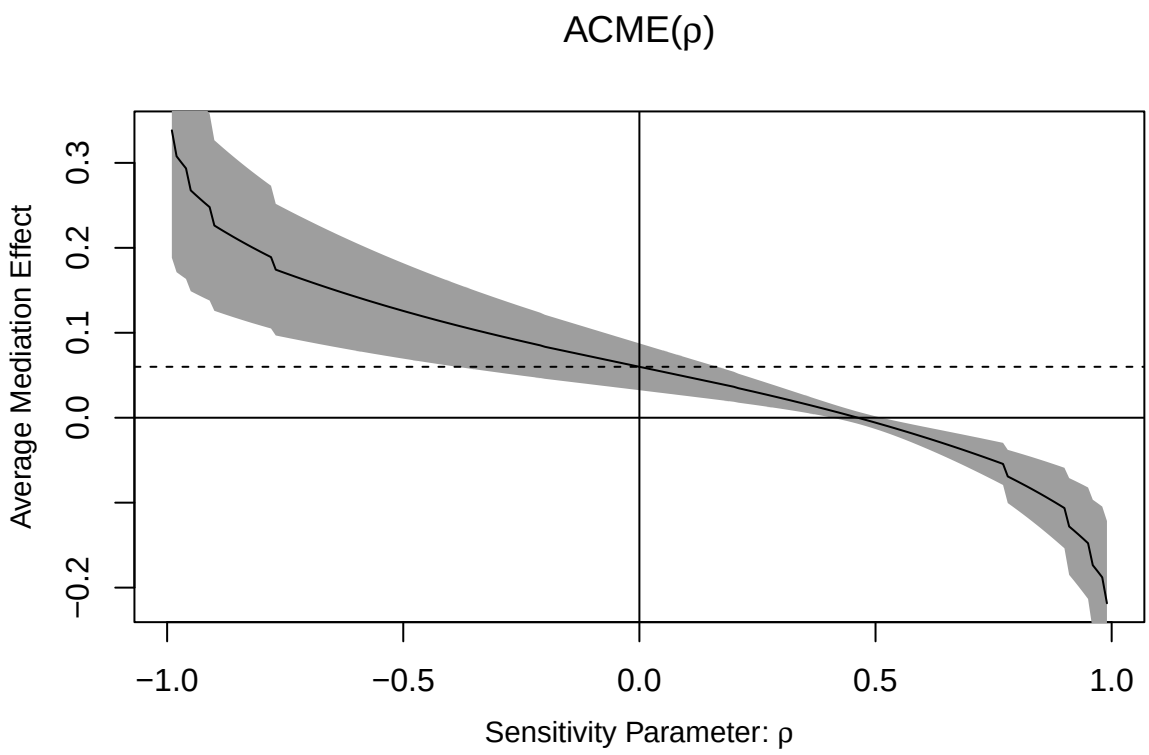
```
## $statistic
## internal
## 16.80819
##
```

```
## $p_value
## internal
##        0
##
## $se
##   internal
## 0.01408376
```

```
summary(med.out.internal)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.0600       0.0335         0.09  <2e-16 ***
## ADE              0.0740       0.0203         0.13   0.008 **
## Total Effect     0.1340       0.0756         0.19  <2e-16 ***
## Prop. Mediated   0.4474       0.2626         0.76  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

```
plot(sens.out.internal)
```




```r
# ------------------------------
# External Motivation Analysis
# ------------------------------

# Direct effect model (same as internal, so no need to recompute)
# dir.fit.external <- dir.fit.internal

# Mediator model
med.fit.external <- lm(external ~ race_feedback, data = d0)

# Outcome model including mediator
out.fit.external <- lm(race_pick ~ race_feedback + external, data = d0)

# Mediation analysis
med.out.external <- mediate(med.fit.external, out.fit.external, boot = TRUE,
                            treat = "race_feedback", boot.ci.type = "perc", mediator =
                            ↪  "external", sims = 10000)

# Sensitivity analysis
sens.out.external <- medsens(med.out.external, rho.by = 0.01, eps=.01, effect.type =
↪  "indirect", sims = 10000)

# Sobel test for external motivation
sobel.external <- sobel_test(med.fit.external, out.fit.external, "external")

# Print and visualize results for external motivation
```

```r
cat("Sobel test for External Motivation\n")
```

```
## Sobel test for External Motivation
```

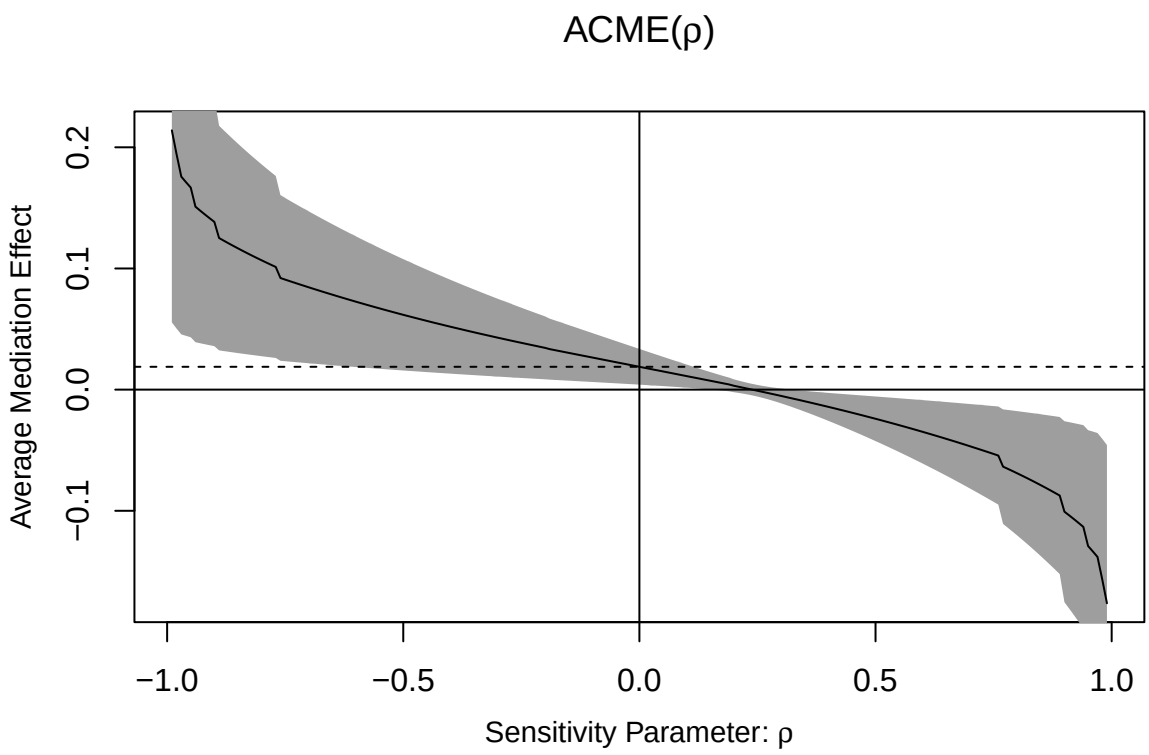
```r
print(sobel.external)
```

```
## $statistic
## external
## 7.576647
##
## $p_value
##     external
## 3.552714e-14
##
## $se
##   external
## 0.01636691
```

```r
summary(med.out.external)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME            0.01884      0.00457         0.03  0.0076 **
## ADE             0.11516      0.05774         0.17  <2e-16 ***
## Total Effect    0.13400      0.07521         0.19  <2e-16 ***
## Prop. Mediated  0.14062      0.03630         0.29  0.0076 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

```r
plot(sens.out.external)
```




```
# ---------------------------------
# Combined Multiple Mediation Model
# ---------------------------------

# Compute the correlation coefficient and p-value
correlation_result <- cor.test(d0$internal, d0$external)
correlation_coefficient <- correlation_result$estimate
p_value <- correlation_result$p.value

# Print the results
cat("Correlation Between Internal and External: ", correlation_coefficient, "\n")
```

```
## Correlation Between Internal and External:  0.7246567
```

```
cat("P-value: ", p_value, "\n")
```

```
## P-value:  1.424694e-163
```

```
# Building combined outcome model with both mediators
out.fit.combined <- lm(race_pick ~ race_feedback + internal + external, data = d0)

# Run combined mediation analyses
med.out.combined.internal <- mediate(med.fit.internal, out.fit.combined, boot = TRUE,
```

```
                                    treat = "race_feedback", boot.ci.type = "perc",
                                    ↪  mediator = "internal", sims = 10000)
med.out.combined.external <- mediate(med.fit.external, out.fit.combined, boot = TRUE,
                                    treat = "race_feedback", boot.ci.type = "perc",
                                    ↪  mediator = "external", sims = 10000)

# Summarize and print the results for combined analysis
cat("Combined Multiple Mediation Model Results\n")
```

```
## Combined Multiple Mediation Model Results
```

```
summary(med.out.combined.internal)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.0779       0.0424         0.11  <2e-16 ***
## ADE              0.0710       0.0181         0.12  0.0084 **
## Total Effect     0.1488       0.0870         0.21  0.0002 ***
## Prop. Mediated   0.5231       0.3240         0.82  0.0002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

```
summary(med.out.combined.external)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME           -0.01482     -0.02886         0.00  0.0060 **
## ADE             0.07097      0.01840         0.12  0.0062 **
## Total Effect    0.05615      0.00238         0.11  0.0424 *
## Prop. Mediated -0.26397     -1.78186         0.00  0.0484 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 1000
##
##
## Simulations: 10000
```

### Figure S3

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

##### poets


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

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


##### books

r_books <- lm(book_pick ~ books, data=d0)

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


#### oldies
```

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

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

## 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"),
  drace_plot %>% mutate(Category = "\nWere Racial\nMinorities")
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('\nWrote Poetry', '\nWrote > 10 books',
  ↪  '\nWere Born\nin the 1800s', '\nWere Racial\nMinorities')))

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 = 4, 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 Racial\nMinorities')), 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 > 10 books') &
   ↪  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) +
    geom_text(data = df_combined %>% filter(Category %in% c('\nWrote Poetry', '\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 = "+"),
             inherit.aes = FALSE, vjust = 0) +
   geom_text(data = df_combined %>% filter(Category == '\nWere Racial\nMinorities' &
   ↪  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) +
```

```
  theme_bw() +
  scale_fill_manual(values = c("#535350", "#c18354"), labels = c("No feedback provided",
  ↪  "Feedback provided"), "Feedback") +
  scale_y_continuous(labels = function(x) paste0(x,"%"), limits = c(0,95)) +
  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",
       caption = 'Note: Error Bars are SEs', 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 = 12),
        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 = 13, vjust = 21),
        plot.title = element_blank(),
        axis.title.y = element_text(size = 12, color = "black"),
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        axis.text.y = element_text(size = 12, color = "black"),
        strip.text = element_text(size = 12, color = "black"),
        strip.background = element_rect(colour = "white", fill = "white"))

#p_combined
# Save the plot with Times New Roman font
ggsave("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
## Race Feedback - poets Feedback          6.192915 0.01290740
## Race Feedback - books Feedback          5.504652 0.01906350
## Race Feedback - oldies Feedback         5.264682 0.02186598
```