Study 4

April 06, 2025

Items

Read Data

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

if(USE_API) {
  ## Pull directly from Qualtrics API
  d0 <- fetch_survey(surveyID='SV_1BJC5tq7iCTDLpk',
                     label = T,
                     convert = F,
                     start_date = "2024-03-27",
                     end_date ="2024-05-15",
                     force_request = T)
} else {
  # Read the processed data directly from CSV
  d0 <- read.csv('Study4.csv', check.names = F)
}


# Define the categories
women <- c('Anne Frank', 'Tina Fey', 'Jackie Kennedy', 'Helen Keller', 'Barbra
↪   Streisand')
pages <- c('Tina Fey', 'Keith Richards', 'Andre Agassi', 'Henry Winkler', 'Willie
↪   Nelson', 'Michael J. Fox', 'John Stamos')
year <- c('Steve Jobs', 'J. Robert Oppenheimer', 'Mark Twain', 'Anthony Bourdain',
↪   'Andrew Jackson', 'Tina Fey', 'Johnny Cash', 'Jackie Kennedy', 'Theodore
↪   Roosevelt', 'Che Guevara', 'Tennessee Williams', 'Keith Richards', 'Napoleon
↪   Bonaparte', 'Andre Agassi', 'Henry Winkler', 'Robin Williams', 'Willie Nelson',
↪   'Michael J. Fox', 'John Stamos')
entertainer <- c('Anthony Bourdain', 'Tina Fey', 'Johnny Cash', 'Jim Carroll', 'Keith
↪   Richards', 'Henry Winkler', 'Robin Williams', 'Willie Nelson', 'Michael J. Fox',
↪   'John Stamos', 'Barbra Streisand')

if(USE_API) {
  d0 <- d0 |>
    filter(!is.na(`choice-7`)) |>
    mutate(
      gender_feedback = as.numeric(grepl("feature a female protagonist", feedbackItem1) |
                  grepl("feature a female protagonist", feedbackItem2) |
                  grepl("feature a female protagonist", feedbackItem3)),
      pages_shown = as.numeric(grepl("are over 500 pages", feedbackItem1) |
                  grepl("are over 500 pages", feedbackItem2) |
                  grepl("are over 500 pages", feedbackItem3)),
      year_shown = as.numeric(grepl("were written in the past 25 years", feedbackItem1) |
                grepl("were written in the past 25 years", feedbackItem2) |
                grepl("were written in the past 25 years", feedbackItem3)),
      entertainer_shown = as.numeric(grepl("feature an entertainer", feedbackItem1) |
                  grepl("feature an entertainer", feedbackItem2) |
                  grepl("feature an entertainer", feedbackItem3)),
      female_pick = case_when(`choice-7` %in% women ~ 1,
                        TRUE ~ 0),
      pages_pick = case_when(`choice-7` %in% pages ~ 1,
                        TRUE ~ 0),
      year_pick = case_when(`choice-7` %in% year ~ 1,
```

```
                                 TRUE ~ 0),
    entertainer_pick = case_when(`choice-7` %in% entertainer ~ 1,
                                 TRUE ~ 0),
    base_gender = rowSums(across(`choice-1`:`choice-6`, ~ . %in% women)),
    gender_code = case_when(gender=="Man" ~ 1, TRUE ~ 0),
    race_code = case_when(race=="White / Caucasian" ~ 1, TRUE ~ 0),
  ) |>
    dplyr::select(gender_feedback:entertainer_pick, gender, base_gender,
↪   `choice-1`:`choice-7`, race,age, gender_code, race_code)

  # Write the API-pulled data into a CSV file
  write.csv(d0, 'Study4.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 biopic with a female protagonist for their seventh selection.
budget_shown	Binary indicator of whether a participant was randomly assigned to receive budget feedback.
budget_pick	Binary indicator of whether a participant selected a film with a big budget for their seventh selection.
year_shown	Binary indicator of whether a participant was randomly assigned to receive film year feedback.
year_pick	Binary indicator of whether a participant selected a film released after 2010 for their seventh selection.
poli_shown	Binary indicator of whether a participant was randomly assigned to receive political leader feedback
poli_pick	Binary indicator of whether a participant selected a film with a political protagonist for their seventh selection.
base_gender	Count of the number of films with female protagonists selected in the initial six films.
choice-1 to choice-7	The selected biographies
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

```
##                              Percentage gender
## 1 Another gender not listed here:   0.36
## 2                             Man  52.52
## 3                     Non-binary   2.16
## 4                          Woman  44.96


##                              Percentage  Race
## 1 American Indian or Alaskan Native   0.36
## 2              Asian / Pacific Islander 42.81
## 3          Black or African American 10.79
## 4                  Hispanic / Latinx   4.68
## 5                  White / Caucasian 41.37


## Mean (age):  25.11


## SD (age):  10.87


## Mean (num of initial women selected):  1.51


## SD (num of initial women selected):  1.15
```

```
## Percentage (initial women selected):  0.2516667


## SD (initial women selected):  0.1916667


## # A tibble: 2 x 2
##   gender_feedback  mean
##             <int> <dbl>
## 1               0 0.230
## 2               1 0.273


##
##  Welch Two Sample t-test
##
## data:  base_gender/6 by gender_feedback
## t = -1.9672, df = 288.06, p-value = 0.05012
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -8.611628e-02  2.356926e-05
## sample estimates:
## mean in group 0 mean in group 1
##       0.2295806       0.2726269
```

Primary Analysis

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

# robust standard errors
robust_summary(r1)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3510 -0.3510 -0.2185  0.6490  0.7815
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.21854    0.03385   6.455 4.34e-10 ***
## gender_feedback  0.13245    0.05172   2.561   0.0109 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4479 on 300 degrees of freedom
## Multiple R-squared:  0.02153,    Adjusted R-squared:  0.01827
## F-statistic: 6.602 on 1 and 300 DF,  p-value: 0.01067
```

```
robust_confint(r1)
```

```
##                      2.5 %     97.5 %
## (Intercept)     0.15192039 0.2851657
## gender_feedback 0.03067135 0.2342293
```

```
# robustness test

# primary model
r_robust <- lm(female_pick ~ gender_feedback + base_gender, data=d0)

# robust standard errors
robust_summary(r_robust)
```

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + base_gender, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3558 -0.3470 -0.2197  0.6442  0.7892
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.222608   0.043876   5.074 6.87e-07 ***
## gender_feedback 0.133213   0.052968   2.515   0.0124 *
## base_gender    -0.002951   0.023299  -0.127   0.8993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4486 on 299 degrees of freedom
## Multiple R-squared:  0.02159,    Adjusted R-squared:  0.01504
## F-statistic: 3.299 on 2 and 299 DF,  p-value: 0.03828
```

`robust_confint(r_robust)`

```
##                      2.5 %      97.5 %
## (Intercept)     0.13626369 0.30895301
## gender_feedback 0.02897536 0.23744980
## base_gender    -0.04880113 0.04289862
```

Robustness

```
## which feedback was shown with gender, remove constant due to ollinearity
r2 <- lm(female_pick ~ gender_feedback + pages_shown + year_shown + entertainer_shown -
→  1, data=d0)

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

```
##
## Call:
## lm(formula = female_pick ~ gender_feedback + pages_shown + year_shown +
##     entertainer_shown - 1, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3770 -0.3276 -0.2185  0.6230  0.7815
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## gender_feedback    0.20377    0.04715   4.322 2.11e-05 ***
## pages_shown        0.09472    0.05614   1.687   0.0926 .
## year_shown         0.04526    0.05628   0.804   0.4219
## entertainer_shown  0.07856    0.06592   1.192   0.2343
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4491 on 298 degrees of freedom
## Multiple R-squared:  0.301,  Adjusted R-squared:  0.2917
## F-statistic: 32.09 on 4 and 298 DF,  p-value: < 2.2e-16
```

```
robust_confint(r2)
```

```
##                        2.5 %     97.5 %
## gender_feedback    0.11097645 0.2965564
## pages_shown       -0.01576071 0.2052072
## year_shown        -0.06550212 0.1560227
## entertainer_shown -0.05116184 0.2082808
```

```
## robust to demographic controls

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

# 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)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5613 -0.3016 -0.1894  0.5678  0.9084
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.157807   0.090183   1.750   0.0815 .
## gender_feedback 0.135678   0.057372   2.365   0.0189 *
## gender_code    -0.085584   0.058297  -1.468   0.1434
## race_code      -0.079007   0.058126  -1.359   0.1754
## age             0.004687   0.003045   1.539   0.1251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.436 on 234 degrees of freedom
##   (63 observations deleted due to missingness)
## Multiple R-squared:  0.05085,    Adjusted R-squared:  0.03463
## F-statistic: 3.134 on 4 and 234 DF,  p-value: 0.01548
```

```
robust_confint(r3)
```

```
##                      2.5 %      97.5 %
## (Intercept)     -0.019868533 0.33548194
## gender_feedback  0.022645502 0.24871006
## gender_code     -0.200438565 0.02927120
## race_code       -0.193524563 0.03551136
## age             -0.001312344 0.01068566
```

```
## logistic regression
# Fit the logistic regression model
r4 <- glm(female_pick ~ 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)        0.280     0.197     -6.47 9.77e-11    0.187     0.406
## 2 gender_feedback    1.93      0.260      2.53 1.13e- 2    1.17      3.24
```

```
summary(r4)
```

```
##
## Call:
## glm(formula = female_pick ~ gender_feedback, family = binomial,
##     data = d0)
##
## Coefficients:
```

```
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.2742     0.1969  -6.471 9.77e-11 ***
## gender_feedback   0.6595     0.2605   2.532   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 360.83  on 301   degrees of freedom
## Residual deviance: 354.28  on 300   degrees of freedom
## AIC: 358.28
##
## Number of Fisher Scoring iterations: 4
```

Figure 6 Code

```r
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\"")) |>
  rename(Condition = gender_feedback)

##### entertainer


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


##### year

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


#### pages


dpages_plot <- d0 |>
  dplyr::select(pages_shown, pages_pick) |>
```

```r
  dplyr::group_by(pages_shown) |>
  dplyr::summarise(
    n = n(),
    freq = mean(pages_pick),
    sd = sd(pages_pick) * 100,
    se = (sd(pages_pick) / sqrt(n())) * 100
  ) |>
  dplyr::mutate(pages_shown = case_when(pages_shown==1 ~ "\"Treatment\"",
                            TRUE ~ "\"Control\"")) |>
  rename(Condition = pages_shown)

## Combine plots

df_combined <- dplyr::bind_rows(
  dentertainer_plot %>% dplyr::mutate(Category = "\nFeatured an\nEntertainer"),
  dyear_plot %>% dplyr::mutate(Category = "\nWere Written in\nthe Past 25 Years"),
  dpages_plot %>% dplyr::mutate(Category = "\nWere Over \n500 Pages"),
  dfemale_plot %>% dplyr::mutate(Category = "\nFeatured a\nWoman")
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('\nFeatured an\nEntertainer', '\nWere
  ↪  Written in\nthe Past 25 Years', '\nWere Over \n500 Pages', '\nFeatured a\nWoman')))

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(data = df_combined %>% filter(Category %in% c('\nWere Over \n500 Pages')),
  ↪  aes(label=paste0(sprintf("%.1f", freq*100),"%")),
  ↪  position=position_dodge(width=0.7), vjust=5, size = 5, color = "white", family =
  ↪  "Times New Roman") +
  geom_text(data = df_combined %>% filter(!Category %in% c('\nWere Over \n500 Pages')),
  ↪  aes(label=paste0(sprintf("%.1f", freq*100),"%")),
  ↪  position=position_dodge(width=0.7), vjust=5, size = 5, 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('\nFeatured an\nEntertainer', '\nWere Written in\nthe
  ↪  Past 25 Years', '\nWere Over \n500 Pages', '\nFeatured a\nWoman')), 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('\nWere Over \n500 Pages',
  ↪  '\nFeatured an\nEntertainer', '\nWere Written in\nthe Past 25 Years') & 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, family = "Times New Roman", size = 6) +
  geom_text(data = df_combined %>% filter(Category %in% c('\nFeatured a\nWoman') &
  ↪  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, family = "Times New Roman", size = 6) +
  theme_bw() +
  scale_fill_manual(values = c("#990000", "#011F5B"), labels = c("No feedback provided",
  ↪  "Feedback provided"), "Feedback") +
```

```r
  scale_y_continuous(labels = function(x) paste0(x,"%"), limits = c(0,100)) +
  scale_x_discrete(labels = c("\"Control\"" = "Not\nShown", "\"Treatment\"" = "Shown")) +
  labs(x = "Feedback on % of biographies that...", y = "% of Final Biographies with the
↪   Target Attribute",
      title = "The Effect of Getting Feedback on Your Biography Selections") +
  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.title = element_blank(),
        legend.direction = "horizontal",
        legend.text = element_text(size = 16, 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 = 16, vjust = 19, family = "Times
↪   New Roman"),
        plot.title = element_blank(),
        axis.title.y = element_text(size = 16, color = "black", family = "Times New
↪   Roman"),
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        axis.text.y = element_text(size = 16, color = "black", family = "Times New
↪   Roman"),
        strip.text = element_text(size = 16, 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-6.pdf", plot = p_combined, width = 10, height = 8,
↪   units = "in", device = cairo_pdf, family = "Times New Roman")
```

System of Equations

```
##                              Test Wald.Coefficient      P_Value
## 1        Female Feedback - Pages      30.233954 5.686358e-08
## 2         Female Feedback - Year       2.391022 1.225646e-01
## 3 Female Feedback - Entertainer        1.752781 1.860351e-01
```