

Study 4

January 28, 2025

Items

Read Data	2
Variable Names	4
Demographics	4
Primary Analysis	5
Figure 6 Code	6
System of Equations	9

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
  d0 <- fetch_survey(surveyID='SV_1BJC5tq7iCTDLpk',
                    label = T,
                    convert = F,
                    start_date = "2024-03-27",
                    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
```

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
## -0.3487 -0.3487 -0.2237  0.6513  0.7763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.22368    0.03402   6.574 2.15e-10 ***
## gender_feedback 0.12500    0.05169   2.418  0.0162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4491 on 302 degrees of freedom
## Multiple R-squared:  0.01912,    Adjusted R-squared:  0.01587
## F-statistic: 5.887 on 1 and 302 DF,  p-value: 0.01584

robust_confint(r1)

##              2.5 %    97.5 %
## (Intercept)  0.15673064 0.2906378
## gender_feedback 0.02328726 0.2267127
```

Figure 6 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\"")) |>
  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) |>
```

```

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(
  denterainer_plot %>% dplyr::mutate(Category = "\"Featured anEntertainer\""),
  dyear_plot %>% dplyr::mutate(Category = "\"Were Written inthe Past 25 Years\""),
  dpages_plot %>% dplyr::mutate(Category = "\"Were Over n500 Pages\""),
  dfemale_plot %>% dplyr::mutate(Category = "\"Featured aWoman\"")
, .id = "id") %>%
mutate(Category = factor(Category, levels = c('\"Featured anEntertainer', 'nWere
  Written inthe Past 25 Years', 'nWere Over n500 Pages', 'nFeatured aWoman')))

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('\"Featured anEntertainer', 'nWere Written inthe
    ↪ Past 25 Years', 'nWere Over n500 Pages', 'nFeatured aWoman')), 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 anEntertainer', 'nWere Written inthe 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 aWoman') &
    ↪ 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") +

```

```

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		26.014870	4.550477e-07
## 2	Female Feedback - Year		2.416112	1.206200e-01
## 3	Female Feedback - Entertainer		1.202289	2.733045e-01