Study 5 - AI Expert Selection

October 17, 2025

Items

Read Data

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

if(USE_API) {
  ## Pull directly from Qualtrics API
  qual_data <- fetch_survey(surveyID='SV_diYpq4P39svNYou',
                            start_date = "2025-10-01",
                     force_request = T)
} else {
  # Read the processed data directly from CSV
  d0 <- read.csv('Study5.csv', check.names = F)
  num_excluded <- unique(d0$num_excluded_total)
}

# Define the categories based on the experts
women <- c('Emily Kwong', 'Moira Gunn', 'Brittany Luse',
           'Zoe Kleinman', 'Davar Ardalan', 'Jane Barrett')

under_50 <- c('Bobby Allyn', 'Emily Kwong', 'Paris Marx',
              'Kevin Roose', 'Ezra Eeman')

west_coast <- c('Erik Brynjolfsson', 'Moira Gunn', 'Ethan Mollick',
                'Ed Zitron', 'Kevin Roose', 'Andrew Ng')

university <- c('Erik Brynjolfsson', 'Austan Goolsbee', 'Ethan Mollick',
                'Anton Korinek', 'Andrew Ng')

if(USE_API) {
  # Process the API data
  d0 <- qual_data |>
    filter(!is.na(`choice-3`), !is.na(PROLIFIC_PID)) |>
    mutate(
      # Treatment assignment
      treatment = case_when(cond == "treatment" ~ 1, TRUE ~ 0),

      # Set assignment (1 or 2)
      set_num = as.numeric(set),

      # Feedback detection
      women_feedback = as.numeric(feedback_women),
      age_feedback = as.numeric(feedback_age),
      location_feedback = as.numeric(feedback_location),
      university_feedback = as.numeric(feedback_university),

      # Count women selected across all 3 choices
      women_choice1 = case_when(`choice-1` %in% women ~ 1, TRUE ~ 0),
      women_choice2 = case_when(`choice-2` %in% women ~ 1, TRUE ~ 0),
      women_choice3 = case_when(`choice-3` %in% women ~ 1, TRUE ~ 0),

      # Total women count and proportion (primary DV)
      women_count = women_choice1 + women_choice2 + women_choice3,
      women_proportion = women_count / 3,
```

```
    # Other attribute picks
    age_count = case_when(`choice-1` %in% under_50 ~ 1, TRUE ~ 0) +
                case_when(`choice-2` %in% under_50 ~ 1, TRUE ~ 0) +
                case_when(`choice-3` %in% under_50 ~ 1, TRUE ~ 0),
    age_proportion = age_count / 3,

    location_count = case_when(`choice-1` %in% west_coast ~ 1, TRUE ~ 0) +
                     case_when(`choice-2` %in% west_coast ~ 1, TRUE ~ 0) +
                     case_when(`choice-3` %in% west_coast ~ 1, TRUE ~ 0),
    location_proportion = location_count / 3,

    university_count = case_when(`choice-1` %in% university ~ 1, TRUE ~ 0) +
                       case_when(`choice-2` %in% university ~ 1, TRUE ~ 0) +
                       case_when(`choice-3` %in% university ~ 1, TRUE ~ 0),
    university_proportion = university_count / 3,

    # Demographics
    gender_code = case_when(gender=="Man" ~ 1, TRUE ~ 0),
    race_code = case_when(str_detect(race, "White / Caucasian") ~ 1, TRUE ~ 0)
  ) |>
  dplyr::select(treatment, set_num, women_feedback, age_feedback, location_feedback,
↪  university_feedback,
              women_count, women_proportion, age_count, age_proportion,
              location_count, location_proportion, university_count,
↪  university_proportion,
              `choice-1`:`choice-3`, race, gender, age, gender_code, race_code, cond,
              ↪  set)

  # Calculate the number of excluded participants
  num_excluded <- nrow(qual_data) - nrow(d0)

  # Save num_excluded in d0
  d0$num_excluded_total <- num_excluded

  # Write the API-pulled data into a CSV file
  write.csv(d0, 'Study5.csv', row.names = FALSE, quote = TRUE)
}
```

Variable Names

Variable	Description
treatment	Binary indicator of whether a participant was randomly assigned to treatment condition (shown women feedback).
set_num	Indicator of which feedback set was shown (1 or 2, with different percentage values).
women_feedback	Binary indicator of whether women feedback was shown to participant.
women_count	Count of women selected across the three choices (0-3).
women_proportion	Proportion of women selected (DV: ranges from 0 to 1).
age_feedback	Binary indicator of whether age feedback was shown.
age_proportion	Proportion of experts under 50 years old selected.
location_feedback	Binary indicator of whether location feedback was shown.
location_proportion	Proportion of experts based on West Coast selected.
university_feedback	Binary indicator of whether university feedback was shown.
university_proportion	Proportion of experts working at a university selected.
choice-1 to choice-3	The selected AI experts
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: 66

##                        Percentage gender
## 1                           Woman  57.51
## 2                             Man  41.97
## 3                      Non-binary   0.00
## 4 Another gender not listed here:   0.52

##                          Percentage  Race
## 1 American Indian or Alaskan Native  0.52
## 2          Asian / Pacific Islander  6.22
## 3         Black or African American 13.47
## 4                 Hispanic / Latinx  7.77
## 5                 White / Caucasian 72.02

## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     44.4   12.6

## Treatment condition:  49.22 %

## Control condition:  50.78 %

## Set 1:  54.4 %

## Set 2:  45.6 %

## Mean proportion of women selected:  0.399

## SD proportion of women selected:  0.275

## # A tibble: 2 x 4
##   treatment  mean    sd     n
##       <dbl> <dbl> <dbl> <int>
## 1         0 0.337 0.242    98
## 2         1 0.463 0.293    95

##
##  Welch Two Sample t-test
##
## data:  women_proportion by treatment
## t = -3.2651, df = 182.13, p-value = 0.001307
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.20281932 -0.05002708
## sample estimates:
## mean in group 0 mean in group 1
##       0.3367347       0.4631579
```

Primary Analysis

```r
# Primary model: Effect of treatment on proportion of women selected
r1 <- lm(women_proportion ~ treatment, data=d0)

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

```
##
## Call:
## lm(formula = women_proportion ~ treatment, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4632 -0.1298 -0.0034  0.2035  0.6633
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33673    0.02454  13.723  < 2e-16 ***
## treatment    0.12642    0.03892   3.248  0.00137 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2681 on 191 degrees of freedom
## Multiple R-squared:  0.05316,    Adjusted R-squared:  0.04821
## F-statistic: 10.72 on 1 and 191 DF,  p-value: 0.001255
```

```r
robust_confint(r1)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2883334 0.3851360
## treatment   0.0496506 0.2031958
```

Robustness

```r
## Which feedback was shown with gender, remove constant due to collinearity
r2 <- lm(women_proportion ~ women_feedback + age_feedback + location_feedback +
↪  university_feedback - 1, data=d0)

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

```
##
## Call:
## lm(formula = women_proportion ~ women_feedback + age_feedback +
##     location_feedback + university_feedback - 1, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5000 -0.1667 -0.0034  0.1860  0.6633
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## women_feedback       0.23201    0.03562   6.514 6.46e-10 ***
## age_feedback         0.06875    0.04509   1.525 0.129033
## location_feedback    0.17986    0.04921   3.655 0.000333 ***
## university_feedback  0.08813    0.04186   2.105 0.036606 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2677 on 189 degrees of freedom
## Multiple R-squared:  0.7004, Adjusted R-squared:  0.6941
## F-statistic: 110.5 on 4 and 189 DF,  p-value: < 2.2e-16
```

```r
robust_confint(r2)
```

```
##                            2.5 %    97.5 %
## women_feedback       0.161749758 0.3022767
## age_feedback        -0.020201857 0.1576977
## location_feedback    0.082787139 0.2769309
## university_feedback  0.005546282 0.1707092
```

```r
## Robust to demographic controls

r3 <- lm(women_proportion ~ treatment + gender_code + race_code + age, data=d0)

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

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

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52484 -0.14158 -0.01432  0.18274  0.61939
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.502985   0.076531   6.572 4.74e-10 ***
## treatment    0.124738   0.038858   3.210  0.00156 **
## gender_code -0.051408   0.039277  -1.309  0.19219
## race_code    0.016002   0.046595   0.343  0.73166
## age         -0.003497   0.001616  -2.164  0.03173 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2655 on 188 degrees of freedom
## Multiple R-squared:  0.08584,    Adjusted R-squared:  0.06639
## F-statistic: 4.414 on 4 and 188 DF,  p-value: 0.001966
```

```r
robust_confint(r3)
```

```
##                    2.5 %        97.5 %
## (Intercept)  0.352015235  0.6539550786
## treatment    0.048083833  0.2013929935
## gender_code -0.128888631  0.0260735541
## race_code   -0.075914978  0.1079193278
## age         -0.006683993 -0.0003091447
```

```r
## Logistic regression on any woman selected (0/1)
d0_logit <- d0 |> mutate(any_woman = case_when(women_count > 0 ~ 1, TRUE ~ 0))

r4 <- glm(any_woman ~ treatment, family = binomial, data=d0_logit)

# 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)     3.08     0.235      4.79 0.00000164    1.98       4.99
## 2 treatment       1.88     0.373      1.69 0.0913        0.914      3.98
```

```r
summary(r4)
```

```
##
## Call:
## glm(formula = any_woman ~ treatment, family = binomial, data = d0_logit)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)   1.1260     0.2349   4.793 1.64e-06 ***
## treatment     0.6294     0.3728   1.688   0.0913 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.48  on 192  degrees of freedom
## Residual deviance: 188.55  on 191  degrees of freedom
## AIC: 192.55
##
## Number of Fisher Scoring iterations: 4
```

Set Assignment Analysis

```r
# Test whether Set 1 vs Set 2 matters
# In Set 1: women feedback shows 10%, in Set 2: women feedback shows 20%

cat("Set Assignment Analysis\n")
```

```
## Set Assignment Analysis
```

```r
cat("========================\n\n")
```

```
## ========================
```

```r
# Overall set effect
r_set <- lm(women_proportion ~ set_num, data=d0)
cat("Main effect of Set (1 vs 2):\n")
```

```
## Main effect of Set (1 vs 2):
```

```r
robust_summary(r_set)
```

```
##
## Call:
## lm(formula = women_proportion ~ set_num, data = d0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44318 -0.10985 -0.02857  0.22348  0.63810
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.28063    0.06019   4.662 5.87e-06 ***
## set_num      0.08128    0.03978   2.043   0.0424 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2725 on 191 degrees of freedom
## Multiple R-squared:  0.02181,    Adjusted R-squared:  0.01669
## F-statistic: 4.258 on 1 and 191 DF,  p-value: 0.04041
```

```r
robust_confint(r_set)
```

```
##                   2.5 %    97.5 %
## (Intercept) 0.161896059 0.3993594
## set_num     0.002815523 0.1597386
```

```r
# Interaction between treatment and set
r_interaction <- lm(women_proportion ~ treatment * set_num, data=d0)
cat("\nInteraction between Treatment and Set:\n")
```

```
##
## Interaction between Treatment and Set:
```

```r
robust_summary(r_interaction)
```

```
##
## Call:
## lm(formula = women_proportion ~ treatment * set_num, data = d0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50758 -0.17424 -0.04545  0.15909  0.62121
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.22615    0.07345   3.079  0.00239 **
## treatment          0.11595    0.11900   0.974  0.33112
## set_num            0.07632    0.04984   1.531  0.12736
## treatment:set_num  0.00642    0.07849   0.082  0.93490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2665 on 189 degrees of freedom
## Multiple R-squared:  0.07405,    Adjusted R-squared:  0.05935
## F-statistic: 5.038 on 3 and 189 DF,  p-value: 0.002214
```

```r
robust_confint(r_interaction)
```

```
##                         2.5 %    97.5 %
## (Intercept)        0.08125488 0.3710459
## treatment         -0.11878894 0.3506830
## set_num           -0.02199116 0.1746286
## treatment:set_num -0.14841712 0.1612580
```

```r
# Simple effects: Does set matter within treatment condition?
d0_treatment <- d0 |> filter(treatment == 1)
d0_control <- d0 |> filter(treatment == 0)

cat("\nSet effect within Treatment condition only:\n")
```

```
##
## Set effect within Treatment condition only:
```

```r
r_set_treatment <- lm(women_proportion ~ set_num, data=d0_treatment)
robust_summary(r_set_treatment)
```

```
##
## Call:
## lm(formula = women_proportion ~ set_num, data = d0_treatment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5076 -0.1742 -0.0915  0.1591  0.5752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.34210    0.09362   3.654 0.000427 ***
## set_num      0.08274    0.06064   1.364 0.175744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2915 on 93 degrees of freedom
## Multiple R-squared:  0.02005,    Adjusted R-squared:  0.009512
## F-statistic: 1.903 on 1 and 93 DF,  p-value: 0.1711
```

```
robust_confint(r_set_treatment)
```

```
##                   2.5 %    97.5 %
## (Intercept)  0.15618216 0.5280127
## set_num     -0.03768542 0.2031637
```

```
cat("\nSet effect within Control condition only:\n")
```

```
##
## Set effect within Control condition only:
```

```
r_set_control <- lm(women_proportion ~ set_num, data=d0_control)
robust_summary(r_set_control)
```

```
##
## Call:
## lm(formula = women_proportion ~ set_num, data = d0_control)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37879 -0.04545  0.03086  0.03086  0.62121
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.22615    0.07345   3.079  0.00271 **
## set_num      0.07632    0.04984   1.531  0.12897
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2399 on 96 degrees of freedom
## Multiple R-squared:  0.02493,    Adjusted R-squared:  0.01477
## F-statistic: 2.454 on 1 and 96 DF,  p-value: 0.1205
```

```r
robust_confint(r_set_control)
```

```
##                   2.5 %    97.5 %
## (Intercept)  0.08034484 0.3719559
## set_num     -0.02260862 0.1752461
```

```r
# Descriptive statistics by condition and set
cat("\nDescriptive statistics by Treatment and Set:\n")
```

```
##
## Descriptive statistics by Treatment and Set:
```

```r
d0 |>
  group_by(treatment, set_num) |>
  summarize(
    mean_women = mean(women_proportion),
    sd_women = sd(women_proportion),
    n = n()
  ) |>
  print()
```

```
## # A tibble: 4 x 5
## # Groups:   treatment [2]
##   treatment set_num mean_women sd_women     n
##       <dbl>   <dbl>      <dbl>    <dbl> <int>
## 1         0       1      0.302    0.227    54
## 2         0       2      0.379    0.255    44
## 3         1       1      0.425    0.291    51
## 4         1       2      0.508    0.292    44
```

```r
# Test treatment effect separately for each set
cat("\nTreatment effect within Set 1 (women feedback shows 10%):\n")
```

```
##
## Treatment effect within Set 1 (women feedback shows 10%):
```

```r
d0_set1 <- d0 |> filter(set_num == 1)
r_treat_set1 <- lm(women_proportion ~ treatment, data=d0_set1)
robust_summary(r_treat_set1)
```

```
##
## Call:
## lm(formula = women_proportion ~ treatment, data = d0_set1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42484 -0.09150  0.03086  0.03086  0.57516
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30247    0.03115   9.709 3.29e-16 ***
## treatment    0.12237    0.05164   2.370   0.0197 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2601 on 103 degrees of freedom
## Multiple R-squared:  0.05337,    Adjusted R-squared:  0.04418
## F-statistic: 5.807 on 1 and 103 DF,  p-value: 0.01774
```

```r
robust_confint(r_treat_set1)
```

```
##                   2.5 %    97.5 %
## (Intercept) 0.24068251 0.3642558
## treatment   0.01995664 0.2247783
```

```r
cat("\nTreatment effect within Set 2 (women feedback shows 20%):\n")
```

```
##
## Treatment effect within Set 2 (women feedback shows 20%):
```

```r
d0_set2 <- d0 |> filter(set_num == 2)
r_treat_set2 <- lm(women_proportion ~ treatment, data=d0_set2)
robust_summary(r_treat_set2)
```

```
##
## Call:
## lm(formula = women_proportion ~ treatment, data = d0_set2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50758 -0.17424 -0.04545  0.15909  0.62121
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.37879    0.03890   9.737 1.55e-15 ***
## treatment    0.12879    0.05912   2.178   0.0321 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2741 on 86 degrees of freedom
## Multiple R-squared:  0.05345,    Adjusted R-squared:  0.04244
## F-statistic: 4.856 on 1 and 86 DF,  p-value: 0.03022
```

```r
robust_confint(r_treat_set2)
```

```
##                   2.5 %    97.5 %
## (Intercept) 0.30145648 0.4561193
## treatment   0.01126536 0.2463104
```

Statistical Approach to Set Differences

```r
# From a statistical perspective, we can test set differences in multiple ways:

cat("Statistical Testing Approaches for Set Differences\n")
```

```
## Statistical Testing Approaches for Set Differences
```

```r
cat("==================================================\n\n")
```

```
## ==================================================
```

```r
cat("Approach 1: Test if the percentage shown (10% vs 20%) affects response\n")
```

```
## Approach 1: Test if the percentage shown (10% vs 20%) affects response
```

```r
cat("----------------------------------------------------------------------\n")
```

```
## ----------------------------------------------------------------------
```

```r
cat("This tests whether the *magnitude* of the feedback percentage matters.\n")
```

```
## This tests whether the *magnitude* of the feedback percentage matters.
```

```r
cat("H0: Set 1 (10%) and Set 2 (20%) produce the same response\n")
```

```
## H0: Set 1 (10%) and Set 2 (20%) produce the same response
```

```r
cat("Ha: Different percentages produce different responses\n\n")
```

```
## Ha: Different percentages produce different responses
```

```r
# Create dummy variable for set 2
d0_treated <- d0 |> filter(treatment == 1) |> mutate(set2 = case_when(set_num == 2 ~ 1,
↪  TRUE ~ 0))

r_dose_response <- lm(women_proportion ~ set2, data=d0_treated)
cat("Test of Set 2 (20%) vs Set 1 (10%) within Treatment:\n")
```

```
## Test of Set 2 (20%) vs Set 1 (10%) within Treatment:
```

```r
robust_summary(r_dose_response)
```

```
##
## Call:
## lm(formula = women_proportion ~ set2, data = d0_treated)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5076 -0.1742 -0.0915  0.1591  0.5752
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.42484    0.04118  10.316   <2e-16 ***
## set2         0.08274    0.06064   1.364    0.176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2915 on 93 degrees of freedom
## Multiple R-squared:  0.02005,    Adjusted R-squared:  0.009512
## F-statistic: 1.903 on 1 and 93 DF,  p-value: 0.1711
```

```r
robust_confint(r_dose_response)
```

```
##                   2.5 %    97.5 %
## (Intercept)  0.34305964 0.5066136
## set2        -0.03768542 0.2031637
```

```r
cat("\n\nApproach 2: Test for interaction (does treatment effect vary by set?)\n")
```

```
##
##
## Approach 2: Test for interaction (does treatment effect vary by set?)
```

```r
cat("----------------------------------------------------------------------\n")
```

```
## ----------------------------------------------------------------------
```

```r
cat("This tests whether the treatment is MORE effective with one set vs another.\n")
```

```
## This tests whether the treatment is MORE effective with one set vs another.
```

```r
cat("H0: Treatment effect is the same for both sets\n")
```

```
## H0: Treatment effect is the same for both sets
```

```r
cat("Ha: Treatment effect differs between sets\n\n")
```

```
## Ha: Treatment effect differs between sets
```

```r
# F-test for interaction term
cat("F-test for interaction term from full model:\n")
```

```
## F-test for interaction term from full model:
```

```r
anova_result <- anova(r_interaction)
print(anova_result)
```

```
## Analysis of Variance Table
##
## Response: women_proportion
##                    Df  Sum Sq Mean Sq F value   Pr(>F)
## treatment           1  0.7710 0.77099 10.8516 0.001179 **
## set_num             1  0.3024 0.30243  4.2566 0.040464 *
## treatment:set_num   1  0.0005 0.00049  0.0069 0.933688
## Residuals         189 13.4281 0.07105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Extract interaction coefficient
interaction_coef <- coef(r_interaction)["treatment:set_num"]
cat("\nInteraction coefficient: ", interaction_coef, "\n")
```

```
##
## Interaction coefficient:  0.006420413
```

```r
cat("Interpretation: The treatment effect changes by ", round(interaction_coef * 100, 2),
    " percentage points for each unit increase in set number\n")
```

```
## Interpretation: The treatment effect changes by  0.64  percentage points for each unit increase in se
```

```r
cat("\n\nApproach 3: Equivalence testing\n")
```

```
##
##
## Approach 3: Equivalence testing
```

```r
cat("----------------------------------------------------------------------\n")
```

```
## ----------------------------------------------------------------------
```

```r
cat("Rather than testing if sets are different, test if they are equivalent.\n")
```

```
## Rather than testing if sets are different, test if they are equivalent.
```

```r
cat("This reverses the null hypothesis to show similarity.\n\n")
```

```
## This reverses the null hypothesis to show similarity.
```

```r
# Calculate treatment effects for each set
treat_effect_set1 <- coef(r_treat_set1)["treatment"]
treat_effect_set2 <- coef(r_treat_set2)["treatment"]

cat("Treatment effect in Set 1: ", round(treat_effect_set1 * 100, 2), "%\n")
```

```
## Treatment effect in Set 1:  12.24 %
```

```r
cat("Treatment effect in Set 2: ", round(treat_effect_set2 * 100, 2), "%\n")
```

```
## Treatment effect in Set 2:  12.88 %
```

```r
cat("Difference: ", round((treat_effect_set2 - treat_effect_set1) * 100, 2), " percentage
↪  points\n")
```

```
## Difference:  0.64  percentage points
```

```r
# Two one-sided tests (TOST) for equivalence
# Define equivalence bounds (e.g., +/- 10 percentage points)
equiv_bound <- 0.10

# Get SEs for each effect
se_set1 <- sqrt(diag(vcovHC(r_treat_set1, type = "HC3")))["treatment"]
se_set2 <- sqrt(diag(vcovHC(r_treat_set2, type = "HC3")))["treatment"]

# Calculate z-scores for equivalence tests
z_upper <- (treat_effect_set1 - treat_effect_set2 - equiv_bound) / sqrt(se_set1^2 +
↪  se_set2^2)
z_lower <- (treat_effect_set1 - treat_effect_set2 + equiv_bound) / sqrt(se_set1^2 +
↪  se_set2^2)

p_upper <- pnorm(z_upper)
p_lower <- pnorm(z_lower, lower.tail = FALSE)

cat("\nEquivalence test (bounds = +/- ", equiv_bound * 100, " percentage points):\n")
```

```
##
## Equivalence test (bounds = +/-  10  percentage points):
```

```r
cat("Upper test p-value: ", p_upper, "\n")
```

```
## Upper test p-value:  0.08758588
```

```r
cat("Lower test p-value: ", p_lower, "\n")
```

```
## Lower test p-value:  0.1165948
```

```r
cat("Equivalence p-value: ", max(p_upper, p_lower), "\n")
```

```
## Equivalence p-value:  0.1165948
```

```r
cat("Conclusion: ", ifelse(max(p_upper, p_lower) < 0.05,
                           "Effects are statistically equivalent",
                           "Cannot conclude equivalence"), "\n")
```

```
## Conclusion:  Cannot conclude equivalence
```

Secondary Analysis: Other Attributes

```r
## Age feedback
r_age <- lm(age_proportion ~ age_feedback, data=d0)

# Display the summary with robust standard errors
cat("Effect of age feedback:\n")
```

```
## Effect of age feedback:
```

```r
robust_summary(r_age)
```

```
##
## Call:
## lm(formula = age_proportion ~ age_feedback, data = d0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.1798 -0.1798 -0.1429  0.1535  0.5238
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.14286    0.03674   3.888 0.000139 ***
## age_feedback  0.03694    0.04084   0.904 0.366876
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2235 on 191 degrees of freedom
## Multiple R-squared:  0.003413,   Adjusted R-squared:  -0.001805
## F-statistic: 0.6541 on 1 and 191 DF,  p-value: 0.4197
```

```r
robust_confint(r_age)
```

```
##                    2.5 %    97.5 %
## (Intercept)   0.07038511 0.2153292
## age_feedback -0.04361778 0.1174995
```

```r
## Location feedback
r_location <- lm(location_proportion ~ location_feedback, data=d0)

# Display the summary with robust standard errors
cat("\nEffect of location feedback:\n")
```

```
##
## Effect of location feedback:
```

```r
robust_summary(r_location)
```

```
##
## Call:
## lm(formula = location_proportion ~ location_feedback, data = d0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25049 -0.25049  0.08284  0.08284  0.74951
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.15278    0.04088   3.737 0.000246 ***
## location_feedback  0.09772    0.04473   2.185 0.030129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2307 on 191 degrees of freedom
## Multiple R-squared:  0.01935,    Adjusted R-squared:  0.01422
## F-statistic: 3.769 on 1 and 191 DF,  p-value: 0.05368
```

```r
robust_confint(r_location)
```

```
##                         2.5 %    97.5 %
## (Intercept)       0.072134046 0.2334215
## location_feedback 0.009492414 0.1859382
```

```r
## University feedback
r_university <- lm(university_proportion ~ university_feedback, data=d0)

# Display the summary with robust standard errors
cat("\nEffect of university feedback:\n")
```

```
##
## Effect of university feedback:
```

```r
robust_summary(r_university)
```

```
##
## Call:
## lm(formula = university_proportion ~ university_feedback, data = d0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27132 -0.22667  0.06202  0.10667  0.77333
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.27132    0.04525   5.996 9.95e-09 ***
## university_feedback -0.04465    0.04961  -0.900    0.369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2588 on 191 degrees of freedom
## Multiple R-squared:  0.005181,   Adjusted R-squared:  -2.782e-05
## F-statistic: 0.9947 on 1 and 191 DF,  p-value: 0.3199
```

```r
robust_confint(r_university)
```

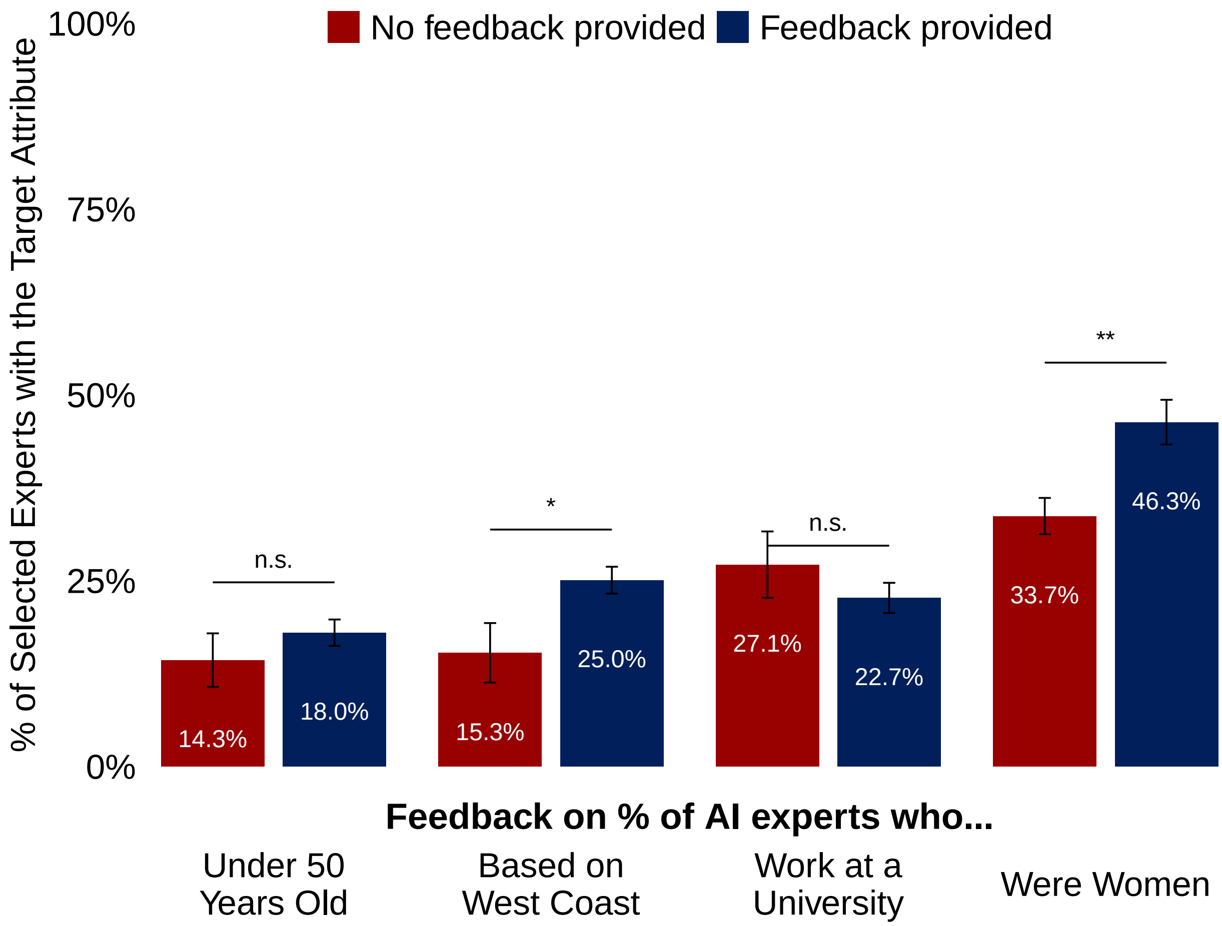
```
##                          2.5 %     97.5 %
## (Intercept)          0.1820577 0.36057795
## university_feedback -0.1425108 0.05320845
```

Visualization

```r
# Get p-values from regression models
p_women <- robust_summary(r1)$coefficients["treatment", "Pr(>|t|)"]
p_age <- robust_summary(r_age)$coefficients["age_feedback", "Pr(>|t|)"]
p_location <- robust_summary(r_location)$coefficients["location_feedback", "Pr(>|t|)"]
p_university <- robust_summary(r_university)$coefficients["university_feedback",
↪  "Pr(>|t|)"]

# Function to convert p-value to significance stars
get_sig_stars <- function(p) {
  if (p < 0.001) return("***")
  else if (p < 0.01) return("**")
  else if (p < 0.05) return("*")
  else return("n.s.")
}

# Get significance labels
sig_women <- get_sig_stars(p_women)
sig_age <- get_sig_stars(p_age)
sig_location <- get_sig_stars(p_location)
sig_university <- get_sig_stars(p_university)

# Women feedback plot
dwomen_plot <- d0 |>
  dplyr::select(women_feedback, women_proportion) |>
  dplyr::group_by(women_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(women_proportion),
    sd = sd(women_proportion) * 100,
    se = (sd(women_proportion) / sqrt(n())) * 100
  ) |>
  dplyr::mutate(
    women_feedback = case_when(
      women_feedback == 1 ~ "\"Treatment\"",
      TRUE ~ "\"Control\""
    )
  ) |>
  dplyr::rename(Condition = women_feedback)

# Age feedback plot
dage_plot <- d0 |>
  dplyr::select(age_feedback, age_proportion) |>
  dplyr::group_by(age_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(age_proportion),
    sd = sd(age_proportion) * 100,
    se = (sd(age_proportion) / sqrt(n())) * 100
  ) |>
  mutate(age_feedback = case_when(age_feedback==1 ~ "\"Treatment\"",
                        TRUE ~ "\"Control\"")) |>
  rename(Condition = age_feedback)
```

```r
# Location feedback plot
dlocation_plot <- d0 |>
  dplyr::select(location_feedback, location_proportion) |>
  dplyr::group_by(location_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(location_proportion),
    sd = sd(location_proportion) * 100,
    se = (sd(location_proportion) / sqrt(n())) * 100
  ) |>
  mutate(location_feedback = case_when(location_feedback==1 ~ "\"Treatment\"",
                             TRUE ~ "\"Control\"")) |>
  rename(Condition = location_feedback)

# University feedback plot
duniversity_plot <- d0 |>
  dplyr::select(university_feedback, university_proportion) |>
  dplyr::group_by(university_feedback) |>
  dplyr::summarise(
    n = n(),
    freq = mean(university_proportion),
    sd = sd(university_proportion) * 100,
    se = (sd(university_proportion) / sqrt(n())) * 100
  ) |>
  mutate(university_feedback = case_when(university_feedback==1 ~ "\"Treatment\"",
                             TRUE ~ "\"Control\"")) |>
  rename(Condition = university_feedback)

## Combine plots

df_combined <- bind_rows(
  dage_plot %>% mutate(Category = "\nUnder 50\nYears Old", sig_label = sig_age),
  dlocation_plot %>% mutate(Category = "\nBased on\nWest Coast", sig_label =
↪  sig_location),
  duniversity_plot %>% mutate(Category = "\nWork at a\nUniversity", sig_label =
↪  sig_university),
  dwomen_plot %>% mutate(Category = "\nWere Women", sig_label = sig_women)
, .id = "id") %>%
  mutate(Category = factor(Category, levels = c('\nUnder 50\nYears Old', '\nBased
  ↪  on\nWest Coast', '\nWork at a\nUniversity', '\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(~Category, 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(Condition == "\"Treatment\""),
             aes(x = 1.5, y = freq*100 + se + 7, label = sig_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(0,100)) +
  scale_x_discrete(labels = c("\"Control\"" = "Not\nShown", "\"Treatment\"" = "Shown")) +
  labs(x = "Feedback on % of AI experts who...", y = "% of Selected Experts with the
   ↪  Target Attribute",
       title = "The Effect of Getting Feedback on Your AI Expert 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"))

print(p_combined)
```




```r
# Save the plot
ggsave("Figure-Study5.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
## Women Feedback - Age Feedback                  172.84238 0.000000e+00
## Women Feedback - Location Feedback              58.39583 1.783018e-13
## Women Feedback - University Feedback           246.02215 0.000000e+00
```