Study 5 - AI Expert Selection

October 20, 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_3TY7uEVc8wzS8N8',
                             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,
```

```r
      # 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: 361

##                        Percentage gender
## 1                           Woman  55.38
## 2                             Man  43.72
## 3                      Non-binary   0.90
## 4 Another gender not listed here:   0.00

##                          Percentage  Race
## 1 American Indian or Alaskan Native  0.80
## 2          Asian / Pacific Islander  7.24
## 3         Black or African American 13.17
## 4                 Hispanic / Latinx  6.53
## 5                 White / Caucasian 72.26

## # A tibble: 1 x 2
##   mean_age sd_age
##      <dbl>  <dbl>
## 1     43.8   13.3

## Treatment condition:  50.05 %

## Control condition:  49.95 %

## Set 1:  51.96 %

## Set 2:  48.04 %

## Mean proportion of women selected:  0.399

## SD proportion of women selected:  0.268

## # A tibble: 2 x 4
##   treatment  mean    sd     n
##       <dbl> <dbl> <dbl> <int>
## 1         0 0.337 0.251   497
## 2         1 0.462 0.270   498

##
##  Welch Two Sample t-test
##
## data:  women_proportion by treatment
## t = -7.5755, df = 987.99, p-value = 8.21e-14
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.15758254 -0.09273866
## sample estimates:
## mean in group 0 mean in group 1
##       0.3366868       0.4618474
```

Primary Analysis

```r
# Primary model: Effect of treatment on proportion of women selected
# As preregistered: includes treatment (gender feedback) and Set1 indicator
r1 <- lm(women_proportion ~ treatment + set_num, data=d0)

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

```
##
## Call:
## lm(formula = women_proportion ~ treatment + set_num, data = d0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47378 -0.14045 -0.01473  0.19288  0.67437
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30320    0.02645  11.463  < 2e-16 ***
## treatment    0.12572    0.01652   7.609 6.41e-14 ***
## set_num      0.02243    0.01656   1.354    0.176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2605 on 992 degrees of freedom
## Multiple R-squared:  0.05638,    Adjusted R-squared:  0.05448
## F-statistic: 29.63 on 2 and 992 DF,  p-value: 3.161e-13
```

```r
robust_confint(r1)
```

```
##                   2.5 %     97.5 %
## (Intercept)  0.25128993 0.35510255
## treatment    0.09329955 0.15814715
## set_num     -0.01007038 0.05493507
```

```r
# Treatment effect only (for comparison)
cat("\n\nTreatment effect without Set indicator (for comparison):\n")
```

```
##
##
## Treatment effect without Set indicator (for comparison):
```

```r
r1_simple <- lm(women_proportion ~ treatment, data=d0)
robust_summary(r1_simple)
```

```
##
## Call:
## lm(formula = women_proportion ~ treatment, data = d0)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46185 -0.12851 -0.00335  0.20482  0.66331
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33669    0.01126  29.889  < 2e-16 ***
## treatment    0.12516    0.01654   7.568 8.64e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2606 on 993 degrees of freedom
## Multiple R-squared:  0.05463,    Adjusted R-squared:  0.05368
## F-statistic: 57.38 on 1 and 993 DF,  p-value: 8.21e-14
```

```r
robust_confint(r1_simple)
```

```
##                  2.5 %    97.5 %
## (Intercept) 0.31458186 0.3587917
## treatment   0.09270623 0.1576150
```

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
## -0.46888 -0.13555 -0.00335  0.19779  0.66331
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## women_feedback       0.23526    0.01477  15.933  < 2e-16 ***
## age_feedback         0.12033    0.01892   6.361 3.05e-10 ***
## location_feedback    0.11330    0.01944   5.827 7.64e-09 ***
## university_feedback  0.10306    0.01662   6.202 8.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2608 on 991 degrees of freedom
## Multiple R-squared:  0.7069, Adjusted R-squared:  0.7057
## F-statistic: 597.6 on 4 and 991 DF,  p-value: < 2.2e-16
```

```r
robust_confint(r2)
```

```
##                          2.5 %    97.5 %
## women_feedback      0.20627992 0.2642305
## age_feedback        0.08320829 0.1574499
## location_feedback   0.07513812 0.1514526
## university_feedback 0.07045351 0.1356711
```

```r
## Robust to demographic controls (PREREGISTERED)
# Create missing indicator variables for demographics
d0_robust <- d0 |>
  mutate(
    gender_missing = case_when(is.na(gender) ~ 1, TRUE ~ 0),
    race_missing = case_when(is.na(race) ~ 1, TRUE ~ 0),
    age_missing = case_when(is.na(age) ~ 1, TRUE ~ 0),
    # Replace NA with 0 for the actual demographic variables
    gender_code = case_when(is.na(gender_code) ~ 0, TRUE ~ gender_code),
    race_code = case_when(is.na(race_code) ~ 0, TRUE ~ race_code),
    age = case_when(is.na(age) ~ 0, TRUE ~ age)
  )
```

```r
r3 <- lm(women_proportion ~ treatment + set_num + gender_code + race_code + age +
         gender_missing + race_missing + age_missing, data=d0_robust)

# Display the summary with robust standard errors
cat("Demographic controls with missing indicators (PREREGISTERED):\n")
```

```
## Demographic controls with missing indicators (PREREGISTERED):
```

```r
robust_summary(r3)
```

```
##
## Call:
## lm(formula = women_proportion ~ treatment + set_num + gender_code +
##     race_code + age + gender_missing + race_missing + age_missing,
##     data = d0_robust)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52506 -0.16940 -0.04917  0.17184  0.73213
##
## Coefficients: (3 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.606e-01  3.990e-02   9.038  < 2e-16 ***
## treatment       1.206e-01  1.635e-02   7.377 3.43e-13 ***
## set_num         2.283e-02  1.635e-02   1.397    0.163
## gender_code    -9.741e-02  1.650e-02  -5.904 4.87e-09 ***
## race_code      -1.611e-02  1.853e-02  -0.869    0.385
## age            -2.747e-05  6.471e-04  -0.042    0.966
## gender_missing         NA         NA      NA       NA
## race_missing           NA         NA      NA       NA
## age_missing            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2563 on 989 degrees of freedom
## Multiple R-squared:  0.08892,    Adjusted R-squared:  0.08431
## F-statistic:  19.3 on 5 and 989 DF,  p-value: < 2.2e-16
```

```r
robust_confint(r3)
```

```
##                        2.5 %       97.5 %
## (Intercept)      0.282290580  0.438869483
## treatment        0.088539129  0.152721255
## set_num         -0.009248816  0.054913539
## gender_code     -0.129779882 -0.065031046
## race_code       -0.052474086  0.020262629
## age             -0.001297322  0.001242389
## gender_missing            NA           NA
## race_missing              NA           NA
## age_missing               NA           NA
```

```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)     2.94     0.103     10.5  1.15e-25     2.41      3.62
## 2 treatment       2.85     0.178      5.88 4.07e- 9     2.02      4.07
```

```r
summary(r4)
```

```
##
## Call:
## glm(formula = any_woman ~ treatment, family = binomial, data = d0_logit)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0799     0.1031  10.473  < 2e-16 ***
## treatment     1.0479     0.1782   5.881 4.07e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 937.77  on 994  degrees of freedom
## Residual deviance: 900.39  on 993  degrees of freedom
## AIC: 904.39
##
## Number of Fisher Scoring iterations: 4
```

```r
## Dropout robustness check (PREREGISTERED)
# For any participants who drop out after assignment to conditions but before selecting
↪  experts,
# impute their women_proportion as the mean from the no gender feedback condition

cat("\n\nDropout Robustness Check (PREREGISTERED):\n")
```

```
##
##
## Dropout Robustness Check (PREREGISTERED):
```

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

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

```r
# Check if there are any missing values in women_proportion among those assigned to
↪  conditions
# (This would indicate dropouts after assignment)
if(any(is.na(d0$women_proportion))) {
  cat("Dropouts detected after condition assignment\n")

  # Calculate mean women_proportion in control condition
  control_mean <- mean(d0$women_proportion[d0$treatment == 0], na.rm = TRUE)

  cat("Control condition mean women proportion:", control_mean, "\n")
  cat("Number of dropouts to impute:", sum(is.na(d0$women_proportion)), "\n\n")

  # Create dataset with imputed values
  d0_imputed <- d0 |>
    mutate(women_proportion_imputed = case_when(
      is.na(women_proportion) ~ control_mean,
      TRUE ~ women_proportion
    ))

  # Run primary model with imputed data
  r5_imputed <- lm(women_proportion_imputed ~ treatment + set_num, data=d0_imputed)

  cat("Primary analysis with dropout imputation:\n")
  robust_summary(r5_imputed)
  robust_confint(r5_imputed)

} else {
  cat("No dropouts detected after condition assignment.\n")
  cat("All participants who were assigned to conditions completed their expert
   ↪  selections.\n")
}
```

```
## No dropouts detected after condition assignment.
## All participants who were assigned to conditions completed their expert selections.
```

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
## -0.40934 -0.07601 -0.05674  0.25732  0.60993
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.37080    0.02617   14.17   <2e-16 ***
## set_num      0.01927    0.01705    1.13    0.259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2678 on 993 degrees of freedom
## Multiple R-squared:  0.001293,   Adjusted R-squared:  0.0002876
## F-statistic: 1.286 on 1 and 993 DF,  p-value: 0.2571
```

```r
robust_confint(r_set)
```

```
##                    2.5 %     97.5 %
## (Intercept)  0.31943331 0.42216139
## set_num     -0.01418618 0.05273332
```

```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
## -0.48498 -0.15165 -0.00408  0.18169  0.66402
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.334543   0.034472   9.705   <2e-16 ***
## treatment         0.063498   0.051066   1.243    0.214
## set_num           0.001436   0.022610   0.064    0.949
## treatment:set_num 0.042033   0.033135   1.269    0.205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2604 on 991 degrees of freedom
## Multiple R-squared:  0.05792,    Adjusted R-squared:  0.05506
## F-statistic: 20.31 on 3 and 991 DF,  p-value: 8.918e-13
```

```r
robust_confint(r_interaction)
```

```
##                         2.5 %     97.5 %
## (Intercept)        0.26689547 0.40218995
## treatment         -0.03671147 0.16370671
## set_num           -0.04293193 0.04580419
## treatment:set_num -0.02298963 0.10705558
```

```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
## -0.4850 -0.1517 -0.1082  0.2252  0.5585
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.39804    0.03767  10.565   <2e-16 ***
## set_num      0.04347    0.02422   1.795   0.0733 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2694 on 496 degrees of freedom
## Multiple R-squared:  0.006468,   Adjusted R-squared:  0.004465
## F-statistic: 3.229 on 1 and 496 DF,  p-value: 0.07295
```

```r
robust_confint(r_set_treatment)
```

```
##                    2.5 %     97.5 %
## (Intercept)  0.324019544 0.47206111
## set_num     -0.004122273 0.09106049
```

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

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

```r
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
## -0.33741 -0.33598 -0.00265 -0.00265  0.66402
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.334543   0.034472   9.705   <2e-16 ***
## set_num     0.001436   0.022610   0.064    0.949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2511 on 495 degrees of freedom
## Multiple R-squared:  8.208e-06,  Adjusted R-squared:  -0.002012
## F-statistic: 0.004063 on 1 and 495 DF,  p-value: 0.9492
```

```r
robust_confint(r_set_control)
```

```
##                    2.5 %     97.5 %
## (Intercept)  0.26681248 0.40227293
## set_num     -0.04298636 0.04585862
```

```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      0.336    0.238   252
## 2         0       2      0.337    0.264   245
## 3         1       1      0.442    0.271   265
## 4         1       2      0.485    0.268   233
```

```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
## -0.44151 -0.10818 -0.00265  0.22516  0.66402
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33598    0.01502  22.363  < 2e-16 ***
## treatment    0.10553    0.02243   4.704 3.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2553 on 515 degrees of freedom
## Multiple R-squared:  0.0411, Adjusted R-squared:  0.03924
## F-statistic: 22.07 on 1 and 515 DF,  p-value: 3.373e-06
```

```r
robust_confint(r_treat_set1)
```

```
##                   2.5 %    97.5 %
## (Intercept) 0.30646314 0.3654945
## treatment   0.06145836 0.1496028
```

```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
## -0.48498 -0.15165 -0.00408  0.18169  0.66259
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33741    0.01690  19.970  < 2e-16 ***
## treatment    0.14756    0.02439   6.051 2.91e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2658 on 476 degrees of freedom
## Multiple R-squared:  0.07176,    Adjusted R-squared:  0.06981
## F-statistic:  36.8 on 1 and 476 DF,  p-value: 2.673e-09
```

```r
robust_confint(r_treat_set2)
```

```
##                  2.5 %    97.5 %
## (Intercept) 0.3042151 0.3706148
## treatment   0.0996466 0.1954805
```

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
## -0.4850 -0.1517 -0.1082  0.2252  0.5585
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.44151    0.01666  26.502   <2e-16 ***
## set2         0.04347    0.02422   1.795   0.0733 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2694 on 496 degrees of freedom
## Multiple R-squared:  0.006468,   Adjusted R-squared:  0.004465
## F-statistic: 3.229 on 1 and 496 DF,  p-value: 0.07295
```

```r
robust_confint(r_dose_response)
```

```
##                     2.5 %     97.5 %
## (Intercept)  0.408777508 0.47424136
## set2        -0.004122273 0.09106049
```

```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  3.897  3.8967 57.4635 7.897e-14 ***
## set_num             1  0.125  0.1249  1.8419    0.1750
## treatment:set_num   1  0.110  0.1096  1.6167    0.2038
## Residuals         991 67.202  0.0678
## ---
## 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:  0.04203298
```

```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  4.2  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:  10.55 %
```

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

```
## Treatment effect in Set 2:  14.76 %
```

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

```
## Difference:  4.2  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:  9.075181e-06
```

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

```
## Lower test p-value:  0.04010875
```

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

```
## Equivalence p-value:  0.04010875
```

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

```
## Conclusion:  Effects are statistically equivalent
```

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
## -0.1787 -0.1787 -0.1532  0.1546  0.8213
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.15323    0.01854   8.265 4.44e-16 ***
## age_feedback  0.02550    0.01983   1.286    0.199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2071 on 993 degrees of freedom
## Multiple R-squared:  0.001654,   Adjusted R-squared:  0.0006482
## F-statistic: 1.645 on 1 and 993 DF,  p-value: 0.2
```

```r
robust_confint(r_age)
```

```
##                    2.5 %     97.5 %
## (Intercept)   0.11684546 0.18960615
## age_feedback -0.01341179 0.06440373
```

```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
## -0.25947 -0.25947  0.07386  0.07386  0.74053
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.21303    0.01901  11.206   <2e-16 ***
## location_feedback  0.04644    0.02075   2.238   0.0254 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2406 on 993 degrees of freedom
## Multiple R-squared:  0.004305,   Adjusted R-squared:  0.003302
## F-statistic: 4.293 on 1 and 993 DF,  p-value: 0.03853
```

```r
robust_confint(r_location)
```

```
##                         2.5 %     97.5 %
## (Intercept)       0.175726054 0.25033911
## location_feedback 0.005727032 0.08715599
```

```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
## -0.24713 -0.24713  0.08621  0.08621  0.81604
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.18396    0.01519  12.111  < 2e-16 ***
## university_feedback  0.06317    0.01775   3.559  0.00039 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2481 on 993 degrees of freedom
## Multiple R-squared:  0.01178,    Adjusted R-squared:  0.01079
## F-statistic: 11.84 on 1 and 993 DF,  p-value: 0.0006044
```

```r
robust_confint(r_university)
```

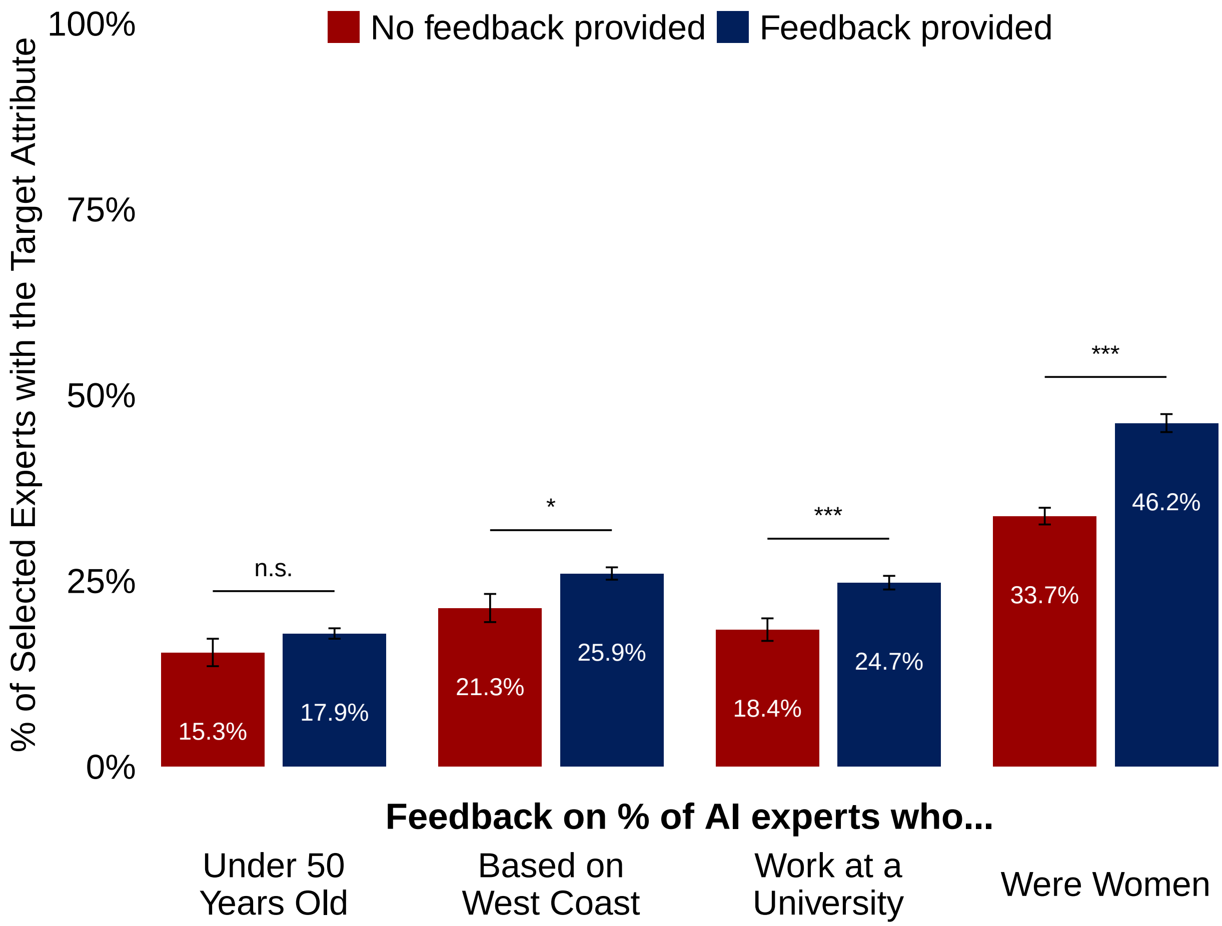
```
##                         2.5 %     97.5 %
## (Intercept)         0.1541490 0.21376244
## university_feedback 0.0283375 0.09800389
```

Visualization

```
# 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),
```

```r
             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 (PREREGISTERED)

```
## Wald Tests for Cross-Equation Comparisons:

## ==========================================

## Test 1: Women Feedback Effect vs. Age Feedback Effect

## ------------------------------------------------------

## Linear hypothesis test (Theil's F test)
##
## Hypothesis:
## ageeq_age_feedback - womeneq_women_feedback = 0
##
## Model 1: restricted model
## Model 2: unrestricted_age
##
##   Res.Df Df      F    Pr(>F)
## 1   1981
## 2   1980  1 785.45 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
##
## Test 2: Women Feedback Effect vs. Location Feedback Effect

## -----------------------------------------------------------

## Linear hypothesis test (Theil's F test)
##
## Hypothesis:
## locationeq_location_feedback - womeneq_women_feedback = 0
##
## Model 1: restricted model
## Model 2: unrestricted_location
##
##   Res.Df Df      F    Pr(>F)
## 1   1981
## 2   1980  1 280.65 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
##
## Test 3: Women Feedback Effect vs. University Feedback Effect

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

```
## Linear hypothesis test (Theil's F test)
##
## Hypothesis:
## universityeq_university_feedback - womeneq_women_feedback = 0
##
## Model 1: restricted model
## Model 2: unrestricted_university
##
##   Res.Df Df      F    Pr(>F)
## 1   1981
## 2   1980  1 337.17 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##
## Summary of Wald Tests:
```

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

```
##                           Test F_Statistic P_Value Significant
##         Women vs. Age Feedback      785.45  <2e-16         Yes
##    Women vs. Location Feedback      280.65  <2e-16         Yes
##  Women vs. University Feedback      337.17  <2e-16         Yes
```

```
##
## Interpretation: A significant result (p < 0.05) indicates that the feedback effect
```

```
## for women is statistically different from the feedback effect for that attribute.
```