I Don't Know - W241 Field Experiments Final Project

Christopher Walker

August 22, 2015

Introduction

This experiment seeks to learn whether people, when faced with questions that they do not know the answers to, will be less likely to admit they do not know when information about previous responses is visible to them. Participants were shown 10 pictures of fake or extremely obscure animals and asked to identify them. The outcome variable in this experiment is the total number of responses that were equivalent to "I don't know."

All data, code, and analysis materials related to this experiment can be found at:

https://github.com/cw25/i_dont_know_experiment

Data Collection

The survey was administered through Qualtrics. Many participants recruited into the study via Amazon Mechanical Turk, and the rest via personal appeals. We begin by ingesting and joining those two data sets.

```r
# Load data files downloaded from Qualtrics and Amazon
setwd("~/Documents/Berkeley/W241 - Field Experiments/Final Project/final")
turk_data <- read.csv("final_survey_results_mturk.csv", header=TRUE, sep=",")
qualtrics_data <- read.csv("final_survey_data_qualtrics.csv", header=TRUE, sep=",")

# Make sure we have a like-named column for joining the two data sets together
qualtrics_data$surveycode = qualtrics_data$random
turk_data$surveycode = turk_data$Answer.surveycode

# Join (note: respondents recruited outside of Turk won't have data in the Amazon file)
final_data_joined <- merge(turk_data, qualtrics_data, all.y = TRUE, by=c("surveycode") )
```

Data Formatting

First, we take the subset of columns in the data that is necessary for the analysis, and simplify the names since the names provided by Qualtrics are quite long.

```r
# Limit to the subset of data necessary for the analysis
final_clean = final_data_joined[, c(
  "group",
  "What.is.your.gender.",
  "What.is.your.age.",
  "Do.you.hold.a.Bachelors.or.higher.college.degree.",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know..",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...1",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...2",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...3",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...4",
```

```
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...5",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...6",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...7",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...8",
  "What.is.the.name.of.this.animal..If.you.don.t.know.the.animal.s.name..just.type..I.don.t.know...9",
  "Were.the.statistics.that.you.were.shown.during.this.study.helpful.in.identifying.the.various.anim..."
)]


# Simplify the column names
colnames(final_clean) = c(
  "group", "gender", "age", "bachelors", "animal1", "animal2", "animal3", "animal4",
  "animal5", "animal6", "animal7", "animal8", "animal9", "animal10", "helpful"
)
```

Next, we view all of the responses to the animal identification questions so we can build the list of responses that will be treated as equivalent to "I don't know." We use those values to create a scoring function that we can use to calculate the outcome variable.

The scoring function also has to be defined in a second way, where we treat blanks as IDK responses. In cases where the user did not answer every animal identification question, we can't know what their responses would have been. As a result, we will use these functions to generate two separate models and produce upper and lower bounds on the estimated CACE.

```
# Inspect the IDK values so we can catch and score them properly
idk_responses = c(as.character(final_clean$animal1), as.character(final_clean$animal2),
                  as.character(final_clean$animal3), as.character(final_clean$animal4),
                  as.character(final_clean$animal5), as.character(final_clean$animal6),
                  as.character(final_clean$animal7), as.character(final_clean$animal8),
                  as.character(final_clean$animal9))


# View responses... This is commented out here because of the length of the output
# sort(unique(idk_responses))


# Convert text responses to scores. All variants of "I don't know" are counted as 1,
# everything else as 0
idk <- function(text) {
  return(as.integer(text %in% c(
    " \"I don't know.\"", " I DON'T KNOW", "\"I don't know\"",
    "80's hair metal Andre Braugher (I don't know)", "I  don't know", "I do not know",
    "I don;t know", "I don;t know.", "I don't k ow", "I don't kmow", "I don't knkw",
    "I don't knoe", "i don't know", "I don't know", "I don't Know", "I don't KNOW",
    "I Don't know", "I DON'T KNOW", "i don't know ", "I don't know ", "I don't Know ",
    "I don't know - looks like 'nessie' / cgi", "i don't know :(", "I don't know :(",
    "I don't know, but it's cute", "I don't know!", "I don't know.", "I don't know. ",
    "I don't know. Looks like a wingless bat!", "I don't know. None of there are real. ",
    "i don't know...oh come on!", "i don't know...this shit is hard", "I don't konw",
    "I don't kow", "I don't ky", "I don't lnow", "I don't[ know", "i don'tknow",
    "I don'tknow", "I don'tknow.", "i dont know", "I dont know", "I dont Know",
    "i dont know ", "I dont know ", "i dont' know", "I dont' know.", "I dont't know",
    "I dontknow", "I odn't know", "I really don't know ", "I. Don't know ", "i.dont know",
    "I've seen this one before... but don't know its name",
```

```
    "albino medusa newt (I don't know)", "Albinodontknow", "don't know", "Don't know",
    "Don't know ", "dont know", "Dont know", "elephagoat (I don't know)",
    "glamour pig (I don't know)", "idk", "Idk", "IDK", "Idon't know",
    "kangaroo mouse? (I don't know)", "medusa salamander (I don't know)",
    "opossum with super sonic hearing (I don't know)",
    "Same as the other one I don't know ", "teddy bear with fangs (I don't know)",
    "What! I don't know "
  )))
}


idk_including_blank <- function(text) {
  return(as.integer(idk(text) | text == ""))
}
```

Now we score the responses using both styles.

```
# Score the responses (assuming blank != IDK)
final_clean$animal1_score_excl = idk(final_clean$animal1)
final_clean$animal2_score_excl = idk(final_clean$animal2)
final_clean$animal3_score_excl = idk(final_clean$animal3)
final_clean$animal4_score_excl = idk(final_clean$animal4)
final_clean$animal5_score_excl = idk(final_clean$animal5)
final_clean$animal6_score_excl = idk(final_clean$animal6)
final_clean$animal7_score_excl = idk(final_clean$animal7)
final_clean$animal8_score_excl = idk(final_clean$animal8)
final_clean$animal9_score_excl = idk(final_clean$animal9)
final_clean$animal10_score_excl = idk(final_clean$animal10)


# Score the responses (assuming blank == IDK)
final_clean$animal1_score_incl = idk_including_blank(final_clean$animal1)
final_clean$animal2_score_incl = idk_including_blank(final_clean$animal2)
final_clean$animal3_score_incl = idk_including_blank(final_clean$animal3)
final_clean$animal4_score_incl = idk_including_blank(final_clean$animal4)
final_clean$animal5_score_incl = idk_including_blank(final_clean$animal5)
final_clean$animal6_score_incl = idk_including_blank(final_clean$animal6)
final_clean$animal7_score_incl = idk_including_blank(final_clean$animal7)
final_clean$animal8_score_incl = idk_including_blank(final_clean$animal8)
final_clean$animal9_score_incl = idk_including_blank(final_clean$animal9)
final_clean$animal10_score_incl = idk_including_blank(final_clean$animal10)


# Generate total scores. These are our outcome variables!
final_clean$total_score_excl = (
                    final_clean$animal1_score_excl + final_clean$animal2_score_excl
                  + final_clean$animal3_score_excl + final_clean$animal4_score_excl
                  + final_clean$animal5_score_excl + final_clean$animal6_score_excl
                  + final_clean$animal7_score_excl + final_clean$animal8_score_excl
                  + final_clean$animal9_score_excl + final_clean$animal10_score_excl)

final_clean$total_score_incl = (
                    final_clean$animal1_score_incl + final_clean$animal2_score_incl
```

```
                  + final_clean$animal3_score_incl + final_clean$animal4_score_incl
                  + final_clean$animal5_score_incl + final_clean$animal6_score_incl
                  + final_clean$animal7_score_incl + final_clean$animal8_score_incl
                  + final_clean$animal9_score_incl + final_clean$animal10_score_incl)

# Uncomment this to see how the two scores differ across all records
# final_clean[
#   final_clean$total_score_incl != final_clean$total_score_excl,
#   c("total_score_incl", "total_score_excl")
# ]
```

We create a few dummy variables that will be helpful during our regression analysis, particularly for the covariates of interest in this study: gender (is the participant male?), age (is the participant over 30?), and education (does the participant hold a Bachelor's or higher degree?).

```
# Helpful dummy variables for identifying the different assignments
final_clean$control = as.integer(final_clean$group == 'control')
final_clean$treatment = as.integer(final_clean$group != 'control')

final_clean$moderate = as.integer(final_clean$group == 'moderate')
final_clean$strong = as.integer(final_clean$group == 'strong')


# Define never-takers as those who did not answer any animal questions
# Those answering any of the animal identification questions will be
# handled as though they were treated
final_clean$complied = as.integer(final_clean$treatment &
                    (final_clean$animal1 != '' | final_clean$animal2 != ''
                    | final_clean$animal3 != '' | final_clean$animal4 != ''
                    | final_clean$animal5 != '' | final_clean$animal6 != ''
                    | final_clean$animal7 != '' | final_clean$animal8 != ''
                    | final_clean$animal9 != '' | final_clean$animal10 != ''))


final_clean$moderate_complied = as.integer(final_clean$moderate & final_clean$complied)
final_clean$strong_complied = as.integer(final_clean$strong & final_clean$complied)


# Convert gender to dummy variable
final_clean$male = as.integer(final_clean$gender == 'Male')

# Convert Bachelors or higher degree to a dummy variable
final_clean$degree = as.integer(final_clean$bachelors == 'Yes')

# Convert age over 30 into a dummy variable
final_clean$over_thirty = as.integer(final_clean$age > 30)

# Convert the manipulation check variable into a dummy
final_clean$stats_helpful = as.integer(final_clean$helpful != 'I did not see any statistics')
```

The Primary Hypothesis

Respondents will give fewer "I don't know" (or equivalent) responses when summary statistics about previous responses are visible.

In order to compute an estimate of the effect, we generate two models, using the two styles of outcome scores that we computed earlier. The first is the "exclusive" or "upper" model where blanks are treated as non-IDK responses, and the magnitude of the estimated effect is greater. The second is the "inclusive" or "lower" model that treats all blanks as IDK responses, and the magnitude of the estimated effect is smaller.

Since we had some non-compliance in the study, and we can't discount the possibility that the treatment itself caused it, we treat those cases as Never-Takers and estimate the effect by calculating the complier average causal effect (CACE). To do this, we use 2-stage regression models, including our covariates of interest.

In order to bracket the true effect size, we use an extreme value bounds strategy. We take the 2.5% quantile for the low extreme and the 97.5% quantile for the high extreme, giving us a very conservative estimate of the range where we can expect the true effect size to live.

```
# Include libraries for IV regression and robust standard errors
library(AER)
library(sandwich)

# Use IV/2-stage regression to derive the estimated CACE, including our covariates
# of interest. We start with the model that excludes blank responses from being
# counted as IDKs. Having fewer IDKs in treatment will cause the CACE to be
# overestimated, so we refer to it as the "upper" model since it will become the
# upper bound in our extreme value bounds.
model_excl = ivreg(total_score_excl ~ complied + male + degree + over_thirty,
                                   ~ treatment + male + degree + over_thirty,
                                   data=final_clean)
upper_model = summary(model_excl)
upper_bound_cace = upper_model$coefficients[2,1]
upper_model
```

```
##
## Call:
## ivreg(formula = total_score_excl ~ complied + male + degree +
##     over_thirty | treatment + male + degree + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3969 -2.1072  0.5978  2.5978  6.2701
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.014351   0.357849  19.601   <2e-16 ***
## complied    -2.994753   0.277282 -10.800   <2e-16 ***
## male        -0.289685   0.271345  -1.068    0.286
## degree       0.373324   0.276803   1.349    0.178
## over_thirty  0.009239   0.292260   0.032    0.975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.183 on 584 degrees of freedom
## Multiple R-Squared: 0.1095,  Adjusted R-squared: 0.1034
```

```
## Wald test: 30.61 on 4 and 584 DF,  p-value: < 2.2e-16
```

```
# Compute robust standard errors for the model
upper_rse = coeftest(model_excl, vcovHC(model_excl))
upper_robust_se = upper_rse[[7]]
upper_rse
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  7.0143506  0.3533700  19.8499    <2e-16 ***
## complied    -2.9947526  0.2767714 -10.8203    <2e-16 ***
## male        -0.2896845  0.2774244  -1.0442    0.2968
## degree       0.3733237  0.2774345   1.3456    0.1789
## over_thirty  0.0092392  0.2903825   0.0318    0.9746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Compute the 97.5% quantile for the more extreme effect
# Signs look a little funny because the effect we are measuring
# is negative. A greater effect means fewer IDK responses.
excl_ci_upper_bound = upper_bound_cace + (1.96 * upper_robust_se)
excl_ci_lower_bound = upper_bound_cace - (1.96 * upper_robust_se)
```

```
# Now run the same model for the "lower" estimate
model_incl = ivreg(total_score_incl ~ complied + male + degree + over_thirty,
                                    ~ treatment + male + degree + over_thirty,
                                    data=final_clean)
lower_model = summary(model_incl)
lower_bound_cace = lower_model$coefficients[2,1]
lower_model
```

```
##
## Call:
## ivreg(formula = total_score_incl ~ complied + male + degree +
##     over_thirty | treatment + male + degree + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0126 -2.0126  0.3767  2.1721  5.1539
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.5912     0.3151  24.090    <2e-16 ***
## complied     -2.3893     0.2442  -9.785    <2e-16 ***
## male         -0.3558     0.2389  -1.489    0.137
## degree        0.1847     0.2437   0.758    0.449
## over_thirty   0.2368     0.2574   0.920    0.358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.803 on 584 degrees of freedom
## Multiple R-Squared: 0.1882,  Adjusted R-squared: 0.1826
## Wald test: 26.06 on 4 and 584 DF,  p-value: < 2.2e-16
```

```r
# Compute robust standard errors for the model
lower_rse = coeftest(model_incl, vcovHC(model_incl))
lower_robust_se = lower_rse[[7]]
lower_rse
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.59117    0.30430 24.9464   <2e-16 ***
## complied    -2.38932    0.24374 -9.8029   <2e-16 ***
## male        -0.35579    0.24757 -1.4371   0.1512
## degree       0.18466    0.24886  0.7420   0.4584
## over_thirty  0.23676    0.25954  0.9122   0.3620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Compute the 2.5% quantile for the less extreme effect
incl_ci_upper_bound = lower_bound_cace + (1.96 * lower_robust_se)
incl_ci_lower_bound = lower_bound_cace - (1.96 * lower_robust_se)
```

Both models report a highly significant treatment effect (p < 0.0001), but with different CACEs (as expected).

The "exclusive" model reports a confidence interval of (-3.5372245, -2.4522806).

The "inclusive" model reports a confidence interval of (-2.867042, -1.9115962).

Taking the two extremes from the separate models, our conservative extreme value bounds estimate of the CACE is (-3.5372245, -1.9115962).

Additional Hypotheses

To investigate the remaining hypotheses, we use the "inclusive" model, which reports the more conservative effect size. (Had any of these additional models produced significant results, the "exclusive" model could have also been generated to produce extreme value bounds estimates of their effect sizes.)

Hypothesis 2

Seeing highly confident summary statistics will lead to a greater reduction in IDK responses than seeing moderately confident summary statistics.

```r
model_h2 = ivreg(total_score_incl
                 ~ strong_complied + moderate_complied + male + degree + over_thirty,
                 ~ strong + moderate + male + degree + over_thirty,
                 data=final_clean)
mh2 = summary(model_h2)
mh2
```

```
##
## Call:
## ivreg(formula = total_score_incl ~ strong_complied + moderate_complied +
##     male + degree + over_thirty | strong + moderate + male +
##     degree + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0074 -2.0074  0.4153  2.1825  5.3135
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.5847     0.3152  24.062  < 2e-16 ***
## strong_complied    -2.5579     0.3053  -8.379 4.01e-16 ***
## moderate_complied  -2.2415     0.2924  -7.665 7.52e-14 ***
## male               -0.3404     0.2395  -1.421    0.156
## degree              0.1898     0.2438   0.779    0.437
## over_thirty         0.2328     0.2574   0.904    0.366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.803 on 583 degrees of freedom
## Multiple R-Squared: 0.1895,  Adjusted R-squared: 0.1825
## Wald test: 21.02 on 5 and 583 DF,  p-value: < 2.2e-16
```

The two levels of treatment report very similar effects, with a great deal of overlap between their respective confidence intervals. The confidence interval for the moderate treatment effect is (-1.6683502, -2.8147449) and the confidence interval for the strong treatment effect is (-1.9595073, -3.1562112).

Given the sizable overlap in the two confidence intervals, we suspect that there is no significant difference between the treatments. We use a basic t-test to formally check our hypothesis.

```
t.test(
  final_clean$total_score_incl[final_clean$strong_complied == 1],
  final_clean$total_score_incl[final_clean$moderate_complied == 1]
)
```

```
##
##  Welch Two Sample t-test
##
## data:  final_clean$total_score_incl[final_clean$strong_complied == 1] and final_clean$total_score_in
## t = -0.90937, df = 265.12, p-value = 0.364
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1445549  0.4213375
## sample estimates:
## mean of x mean of y
##  5.007519  5.369128
```

The test does not yield a statistically significant result. We therefore fail to reject the null hypothesis that there is no difference between the effect of the moderate treatment and the strong treatment.

Hypothesis 3

Males receiving the treatment will be less likely than females to answer IDK.

```
h3_stage1 = lm(complied ~ treatment + male + degree + over_thirty, data=final_clean)
h3_stage2 = lm(
  total_score_incl~ h3_stage1$fitted.values * male + degree + over_thirty,
  data = final_clean
)
mh3 = summary(h3_stage2)
mh3
```

```
##
## Call:
## lm(formula = total_score_incl ~ h3_stage1$fitted.values * male +
##     degree + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0635 -2.0635  0.4082  2.1208  4.9832
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.6343     0.3355  22.758  < 2e-16 ***
## h3_stage1$fitted.values      -2.4932     0.3289  -7.580 1.37e-13 ***
## male                         -0.4763     0.3482  -1.368    0.172
## degree                        0.1824     0.2504   0.728    0.467
## over_thirty                   0.2433     0.2646   0.919    0.358
## h3_stage1$fitted.values:male  0.2481     0.5083   0.488    0.626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.878 on 583 degrees of freedom
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.1379
## F-statistic: 19.82 on 5 and 583 DF,  p-value: < 2.2e-16
```

The interaction term representing males under treatment has a p-value of 0.6257362, not a statistically significant result. Thus, we fail to reject the null hypothesis that males and females are equally likely to answer IDK when receiving the treatment.

Hypothesis 4

Holders of 4-year college degrees receiving the treatment will be even more likely to answer IDK than those who do not hold degrees.

```
h4_stage1 = lm(complied ~ treatment + degree + male + over_thirty, data=final_clean)
h4_stage2 = lm(
  total_score_incl ~ h4_stage1$fitted.values * degree + male + over_thirty,
  data = final_clean
)
mh4 = summary(h4_stage2)
mh4
```

```
##
## Call:
## lm(formula = total_score_incl ~ h4_stage1$fitted.values * degree +
##     male + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9246 -2.0075  0.4298  2.0754  5.2211
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    7.76748    0.36301  21.397  < 2e-16 ***
## h4_stage1$fitted.values       -2.75692    0.42553  -6.479 1.97e-10 ***
## degree                        -0.08416    0.35473  -0.237    0.813
## male                          -0.35927    0.24522  -1.465    0.143
## over_thirty                    0.23817    0.26410   0.902    0.368
## h4_stage1$fitted.values:degree 0.56146    0.52532   1.069    0.286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.876 on 583 degrees of freedom
## Multiple R-squared:  0.1466, Adjusted R-squared:  0.1393
## F-statistic: 20.03 on 5 and 583 DF,  p-value: < 2.2e-16
```

The coefficient for the term interacting 4-year college degree holders with the treatment has a p-value of 0.2856054, and is not a statistically significant result. We fail to reject the null hypothesis that there is no difference in the effect between participants with or without 4-year degrees.

Hypothesis 5

Participants over 30 years old who receive the treatment will be more likely to answer IDK than those aged 30 years or younger.

```
h5_stage1 = lm(complied ~ treatment + degree + male + over_thirty, data=final_clean)
h5_stage2 = lm(
  total_score_incl ~ h5_stage1$fitted.values * over_thirty + male + degree,
  data = final_clean
)
mh5 = summary(h5_stage2)
mh5
```

```
##
## Call:
## lm(formula = total_score_incl ~ h5_stage1$fitted.values * over_thirty +
##     male + degree, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9892 -1.9892  0.3576  2.1983  5.0993
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                              7.6567      0.3761  20.357  < 2e-16
## h5_stage1$fitted.values                  -2.5207      0.4589  -5.493 5.91e-08
## over_thirty                               0.1435      0.3799   0.378    0.706
## male                                     -0.3520      0.2457  -1.433    0.152
## degree                                    0.1858      0.2504   0.742    0.458
## h5_stage1$fitted.values:over_thirty       0.1871      0.5472   0.342    0.733
##
## (Intercept)                          ***
## h5_stage1$fitted.values              ***
## over_thirty
## male
## degree
## h5_stage1$fitted.values:over_thirty
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.879 on 583 degrees of freedom
## Multiple R-squared:  0.1451, Adjusted R-squared:  0.1378
## F-statistic: 19.79 on 5 and 583 DF,  p-value: < 2.2e-16
```

The coefficient for the term interacting participants aged over 30 with the treatment has a p-value of 0.7325932, and is not a significant result. We fail to reject the null hypothesis that there is no difference in effect between those whose age is over 30 and those who are 30 and under.

Manipulation Check

And, as a sanity check on delivery of the treatment, we also test the assignment of participants to treatment and control groups to make sure we do not have significant differences. Assignment to the two levels of treatment is also checked to ensure there were no differences.

```
# Manipulation check
t.test(
  final_clean$stats_helpful[final_clean$control == 1 & !is.na(final_clean$stats_helpful)],
  final_clean$stats_helpful[final_clean$control == 0 & !is.na(final_clean$stats_helpful)]
)
```

```
##
##  Welch Two Sample t-test
##
## data:  final_clean$stats_helpful[final_clean$control == 1 & !is.na(final_clean$stats_helpful)] and f:
## t = -48.476, df = 311.22, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9266123 -0.8543249
## sample estimates:
## mean of x mean of y
## 0.1061644 0.9966330
```

As expected, the test for difference in means yields a significant result, indicating the treatment was successfully delivered.