I Don't Know - W241 Field Experiments Final Project

Christopher Walker

August 20, 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 rare 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 assignment can be obtained at: https://github.com/cw25/i_dont_know_experiment

Data Collection

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

```
# 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.

```
# 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$animal2_score_excl + final_
                        + final_clean$animal4_score_excl + final_clean$animal5_score_excl + final_clean$
                        + final_clean$animal7_score_excl + final_clean$animal8_score_excl + final_clean$
                        + final_clean$animal10_score_excl)

final_clean$total_score_incl = (final_clean$animal1_score_incl + final_clean$animal2_score_incl + final_
                           + final_clean$animal4_score_incl + final_clean$animal5_score_incl + final_cl
                           + final_clean$animal7_score_incl + final_clean$animal8_score_incl + final_cl
                           + final_clean$animal10_score_incl)
```

```r
# Uncomment to look at the data to get a sense of how scores differ
# final_clean[final_clean$total_score_incl != final_clean$total_score_excl, c("total_score_incl", "tota
```

```r
# 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 up to 9 of the questions will be handled as though
# they were treatedx
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 the manipulation check variable into a dummy
final_clean$stats_helpful = as.integer(final_clean$helpful != 'I did not see any statistics')

# Using age over 24 (the lowest age observed in the study). Without the adjustment,
# when the age is included in regressions, the intercept comes out around 13,
# which seems like an impossible value and can be a bit confusing.
final_clean$age_over_24 = final_clean$age - min(final_clean$age)

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




# Data Analysis


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

```
## Loading required package: car
```

```
## Loading required package: lmtest
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: sandwich
## Loading required package: survival
```

```
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_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
```

```
upper_bound_cace = upper_model$coefficients[2,1]
upper_rse = coeftest(model_excl, vcovHC(model_excl))
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
```

```
upper_robust_se = upper_rse[[7]]

# Check the outputs for the CACE and robust SE
upper_bound_cace
```

```
## [1] -2.994753
```

```
upper_robust_se
```

```
## [1] 0.2767714
```

```
# 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.
ci_upper = upper_bound_cace - (1.96 * upper_robust_se)
ci_upper
```

```
## [1] -3.537225
```

```
# 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_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
lower_bound_cace = lower_model$coefficients[2,1]
lower_rse = coeftest(model_incl, vcovHC(model_incl))
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
lower_robust_se = lower_rse[[7]]

# Check outputs
lower_bound_cace
```

```
## [1] -2.389319
```

```r
lower_robust_se
```

```
## [1] 0.2437362
```

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

```
## [1] -1.911596
```

```r
# Use the inclusive model since it represents the lesser effect (stay conservative)
# and investigate the additional hypotheses


# Hypothesis 2: Seeing highly confident summary statistics will lead to a greater reduction in IDK
# responses than seeing moderately confident summary statistics
model_h2 = ivreg(total_score_incl ~ strong_complied + moderate_complied + male + degree + over_thirty,
                                    ~ strong + moderate + male + degree + over_thirty,
                                    data=final_clean)
summary(model_h2)
```

```
##
## 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
```

```r
# Hypothesis 3: Males under treatment will be even less likely than females to answer IDK
model_h3 = ivreg(total_score_incl ~ complied * male + degree + over_thirty,
                 ~ treatment * male + degree + over_thirty,
                 data=final_clean)
summary(model_h3)
```

```
##
## 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
## -8.0586 -2.0586  0.4145  2.1228  5.0890
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     7.6327     0.3267  23.365  < 2e-16 ***
## complied       -2.4903     0.3202  -7.777 3.38e-14 ***
## male           -0.4731     0.3393  -1.394    0.164
## degree          0.1813     0.2440   0.743    0.458
## over_thirty     0.2445     0.2581   0.947    0.344
## complied:male   0.2417     0.4959   0.487    0.626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.805 on 583 degrees of freedom
## Multiple R-Squared: 0.1883,  Adjusted R-squared: 0.1813
## Wald test: 20.87 on 5 and 583 DF,  p-value: < 2.2e-16
```

```r
# Hypothesis 4: Holders of 4-year college degrees will be even more likely to answer IDK
model_h4 = ivreg(total_score_incl ~ complied * degree + male + over_thirty,
                 ~ treatment * degree + male + over_thirty,
                 data=final_clean)
summary(model_h4)
```

```
##
## Call:
## ivreg(formula = total_score_incl ~ complied * degree + male +
##     over_thirty | treatment * degree + male + over_thirty, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9246 -2.0056  0.4357  2.0754  5.3505
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.76092    0.35185  22.058  < 2e-16 ***
## complied         -2.75119    0.41369  -6.650 6.74e-11 ***
## degree           -0.08098    0.34583  -0.234    0.815
## male             -0.36028    0.23892  -1.508    0.132
## over_thirty       0.24469    0.25740   0.951    0.342
## complied:degree   0.55377    0.51155   1.083    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.802 on 583 degrees of freedom
## Multiple R-Squared:  0.19,   Adjusted R-squared: 0.183
## Wald test:  21.1 on 5 and 583 DF,  p-value: < 2.2e-16
```

```r
# Hypothesis 5: Participants over the age of 30 receiving the treatment are more likely to answer IDK
model_h5 = ivreg(total_score_incl ~ complied * over_thirty + male + degree,
                 ~ treatment * over_thirty + male + degree,
                 data=final_clean)
summary(model_h5)
```

```
##
## Call:
## ivreg(formula = total_score_incl ~ complied * over_thirty + male +
##     degree | treatment * over_thirty + male + degree, data = final_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9909 -1.9909  0.3611  2.1696  5.2053
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            7.6431     0.3627  21.071  < 2e-16 ***
## complied              -2.4964     0.4432  -5.632 2.77e-08 ***
## over_thirty            0.1605     0.3691   0.435    0.664
## male                  -0.3520     0.2395  -1.470    0.142
## degree                 0.1873     0.2441   0.767    0.443
## complied:over_thirty   0.1530     0.5306   0.288    0.773
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.805 on 583 degrees of freedom
## Multiple R-Squared: 0.1883,  Adjusted R-squared: 0.1813
## Wald test: 20.84 on 5 and 583 DF,  p-value: < 2.2e-16
```

```r
# 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 fi
## 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
```