Data Exploration: Emotional Arousal

Yanxi Fang

November 4, 2021

In this Data Exploration assignment we will explore Clifford and Jerit's (2018) findings about the effects of disgust and anxiety on political learning.

If you have a question about any part of this assignment, please ask! Note that the actionable part of each question is **bolded**.

Emotional Arousal: Disgust and Anxiety

```r
# read in data for Study 1
Study1_preprocess <- read_dta("Study1ReplicationData.dta")
```

Question 1 DO NOT SKIP

Cleaning and organizing data is an important part of any research process. Note: Your blog posts should not address the data cleaning portion of the assignment but rather the content of the material

Part a

Below, the variables Q13, Q14, and Q15 are renamed to `disease_transm`, `cure`, and `addl_info`, respectively, based on the variable descriptions.

```r
#Method 1: Renaming by name
Study1_processing1 <- Study1_preprocess %>%
  rename_with(.cols = c(Q13, Q14, Q15), ~c("disease_transm", "cure", "addl_info"))
```

Part b

Below, the last six columns are renamed to `gender`, `race`, `education`, `partyid`, `voter_reg`, and `ideo`, respectively, based on the variable descriptions.

```r
#Method 2: Renaming by index/position
#last_col() is just a function that returns the index number of the last variable in the dataset (33 in
Study1_processing2 <- Study1_processing1 %>%
  rename_with(.cols = last_col(offset = 5):last_col(), ~c("gender", "race", "education",
                                                          "partyid", "voter_reg", "ideo"))
```

Part c

The individual variables relating to emotional reaction of the respondent from Q11 are renamed below, based on the variable descriptions, to `disgust`, `grossed_out`, `repulsed`, `afraid`, `anxious`, and `worried`.

```r
#Method 3: Renaming by logical condition
Study1_processing3 <- Study1_processing2 %>%
  rename_with(.cols = contains("Q11"), ~c("disgust", "grossed_out", "repulsed",
                                          "afraid", "anxious", "worried"))
```

Part d

Below, all of the Question 16 variables are removed from the dataset.

```r
Study1_processing4 <- Study1_processing3 %>%
  select(-contains("Q16"))
```

Part e

Below, the values of `8` in all relevant data columns are changed into `NA`s, turning the variables into true binary ones.

```r
Study1_processing5 <- Study1_processing4 %>%
  mutate(across(.cols = c("disgust", "grossed_out", "repulsed", "afraid",
                          "anxious", "worried", "disease_transm", "cure",
                          "addl_info", "Q17_7"), ~na_if(., 8)))
```

Part f

Below, the `2`s in the relevant data columns are changed to `0`s (since they represent `No`, according to the variable description), while the `1`s, which represent `Yes`, remain as `1`s.

```r
Study1_processing6 <- Study1_processing5 %>%
  mutate(across(.cols = c("Q12_1", "Q12_2", "Q12_3", "Q12_4", "Q12_5", "Q12_6",
                          "Q12_7", "cure", "addl_info"),
                ~ifelse(. == 1, 1, ifelse(. == 2, 0, NA))))
```

Part g

Due to the complexity behind the fact that different sets of symptoms are correct according to the different treatments, I skipped this and moved onto the next question.

Part h

Below, I renamed the Q12 variables according to the symptoms listed in the variable descriptions. After this was done, I finalized the dataset for use in the rest of this assignment.

```r
# rename Q12 variables
Study1_processing7 <- Study1_processing6 %>%
  rename_with(.cols = contains("Q12"), ~c("fatigue", "headaches", "diarrhea",
                                          "joint", "boils", "warts", "fever"))

# finalize dataset
study1 <- Study1_processing7
```

Question 4: DATA SCIENCE QUESTION

Part a

Below, I used the Cronbach's alpha formula to manually calculate the values for the disgust index and the anxiety index used in the paper. As shown, the alpha values are 0.932 and 0.917, respectively.

```
# disgust index: 3 items (disgusted, grossed out, repulsed)

# drop NAs to calculate variances and covariances
study1_d <- study1 %>%
  select(disgust:repulsed) %>%
  drop_na()

# calculate variances
dv1 <- var(study1_d$disgust)
dv2 <- var(study1_d$grossed_out)
dv3 <- var(study1_d$repulsed)
sum_var_d <- sum(dv1, dv2, dv3)

# calculate covariances
d1 <- cov(study1_d$disgust, study1_d$grossed_out)
d2 <- cov(study1_d$disgust, study1_d$repulsed)
d3 <- cov(study1_d$grossed_out, study1_d$repulsed)
sum_cov_d <- sum(d1, d2, d3)

# calculate Cronbach's alpha
(3*sum_cov_d)/(sum_var_d + 2*sum_cov_d)
```

```
## [1] 0.9323854
```

```
# anxiety index: 3 items (afraid, anxious, worried)

# drop NAs to calculate variances and covariances
study1_a <- study1 %>%
  select(afraid:worried) %>%
  drop_na()

# calculate variances
av1 <- var(study1_a$afraid)
av2 <- var(study1_a$anxious)
av3 <- var(study1_a$worried)
sum_var_a <- sum(av1, av2, av3)

# calculate covariances
a1 <- cov(study1_a$afraid, study1_a$anxious)
a2 <- cov(study1_a$afraid, study1_a$worried)
a3 <- cov(study1_a$anxious, study1_a$worried)
sum_cov_a <- sum(a1, a2, a3)

# calculate Cronbach's alpha
(3*sum_cov_a)/(sum_var_a + 2*sum_cov_a)
```

```
## [1] 0.9168325
```

Part b

Below, I used the `cronbach.alpha()` function from the ltm package to calculate Cronbach's alpha for the disgust index and the anxiety index. These results do match my previous calculations, and based on the rule of thumb that Cronbach's alpha values less than 0.7 show indices that are insufficiently internally consistent, the two scales, with alpha values above 0.9, do seem to each be sufficiently internally consistent.

```
library(ltm)

cronbach.alpha(study1_d)
```

```
##
## Cronbach's alpha for the 'study1_d' data-set
##
## Items: 3
## Sample units: 991
## alpha: 0.932
```

```
cronbach.alpha(study1_a)
```

```
##
## Cronbach's alpha for the 'study1_a' data-set
##
## Items: 3
## Sample units: 992
## alpha: 0.917
```

Part c

Below, I chose to create a multivariate regression model with `page_article_timing`, the amount of time that each respondent spent on the page containing the article about the disease, as the outcome variable. I used a wide range of explanatory variables: the treatment (low/high anxiety and low/high disgust), the respondent's answers to the disgust and anxiety scales (a total of 6 variables), and whether the respondent sought out more information (3 variables). I added in the respondent's gender, race, education, party identification, and voter registration status as control variables, since there may presumably be some difference in how, say, females perceive threats versus males (or Republicans versus Democrats, as seen during the worldview week). My hypothesis is that the amount of time that each respondent spends should be based upon the treatment, as well as their response to the anxiety and disgust scales, although I am concerned about potentially some degree of multicollinearity between treatment and the two scales.

Here, I am very aware of the fact that due to the survey design, the amount of time spent (my outcome variable) actually comes *before* most of the other collected variables, making causality far from clear. However, this model is meant to potentially measure whether people who are more disgusted are also the people who spent more time; in other words, I'm looking for a correlation only.

The results are printed below. As shown, only one variable – `grossed_out` – had a statistically significant estimated coefficient at the standard 5% significance level; all other variables had p-values that were quite far away from 0.05. The coefficient on that variable, `45.97`, suggests that when holding all other variables equal (including the treatment), a person that answered with a response that is numerically larger by 1 on the `grossed_out` scale is predicted to spend almost 46 seconds longer than a person who answers with a `grossed_out` scale response that is 1 level lower. This makes intuitive sense: numerically larger values on that scale indicate a higher level of disgust about the virus, so it would make sense that someone who finds the virus more disgusting would spend more time finding out about it. (This can come about due to

2 reasons: the person is naturally concerned about viruses – like germophobes – and spends more time on the article because they want to be more informed and prepared, or the person becomes more disgusted as a result of reading more about the virus in the article.)

In addition, the model comes with a very low R-squared value (in fact, the adjusted R-squared is negative, showing that the model is essentially worse than randomly guessing the amount of time that someone is spending on the article). This means that overall, the covariates that I picked are, together, not very good at explaining the variation in the amount of time spent. This is perhaps because there is a *lot* of variation in that outcome variable: the minimum is 0.975 and the maximum is $10,437.234$ seconds, making it difficult for any model to predict…

```r
model1 <- lm(page_article_timing ~ treat_rand1 + disgust + grossed_out + repulsed +
               afraid + anxious + worried + addl_info + Q17_6 + Q17_7 +
               gender + race + education + partyid + voter_reg, data = study1)
summary(model1)
```

```
##
## Call:
## lm(formula = page_article_timing ~ treat_rand1 + disgust + grossed_out +
##     repulsed + afraid + anxious + worried + addl_info + Q17_6 +
##     Q17_7 + gender + race + education + partyid + voter_reg,
##     data = study1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -277.2   -76.2   -44.0    -4.0 10184.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.5092    82.3231   1.245   0.2134
## treat_rand1  -4.6144    12.6357  -0.365   0.7151
## disgust     -35.0883    23.3270  -1.504   0.1329
## grossed_out  45.9680    21.0918   2.179   0.0295 *
## repulsed    -20.6125    22.6325  -0.911   0.3627
## afraid       -8.2145    23.4309  -0.351   0.7260
## anxious       7.9116    20.9111   0.378   0.7053
## worried      15.6100    21.4989   0.726   0.4680
## addl_info    -3.6644    35.2827  -0.104   0.9173
## Q17_6       -22.6330    15.6670  -1.445   0.1489
## Q17_7        15.7492    17.8641   0.882   0.3782
## gender       39.8703    27.6008   1.445   0.1489
## race          7.2078    11.4171   0.631   0.5280
## education    -1.7295     9.8601  -0.175   0.8608
## partyid      -0.4638     6.2244  -0.075   0.9406
## voter_reg   -10.9406    33.3589  -0.328   0.7430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 421.9 on 954 degrees of freedom
##   (30 observations deleted due to missingness)
## Multiple R-squared:  0.0114, Adjusted R-squared:  -0.004141
## F-statistic: 0.7336 on 15 and 954 DF,  p-value: 0.7515
```

```
summary(study1$page_article_timing)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.975    25.823    48.196    95.967    69.113 10437.234
```

```
var(study1$page_article_timing)
```

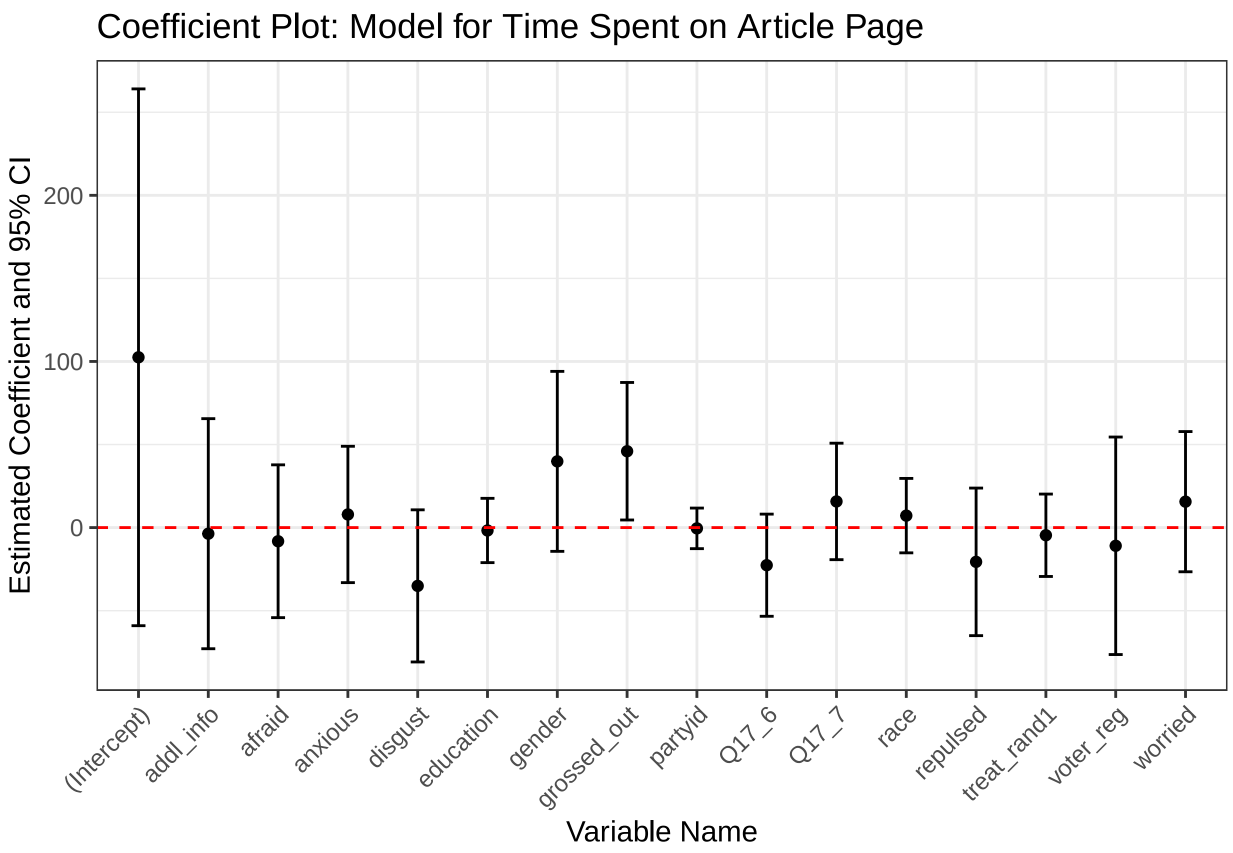
```
## [1] 187679.1
```

Part d

Below, I created a coefficient plot corresponding to the model from part (c). As shown, the confidence intervals are very wide for each of the coefficients, and as expected based on the p-values from the model summary, only the `grossed_out` variable has a CI that doesn't cross the $y = 0$ line.

```
# find confidence intervals and create dataframe for plotting
point <- summary(model1)$coefficients[,1]
ci <- confint(model1)
plotdata <- cbind(point, ci)
plotdata <- as.data.frame(plotdata)
plotdata <- tibble::rownames_to_column(plotdata, "variable")

# plot
plotdata %>%
  ggplot(aes(x = variable, y = point)) +
  geom_point() +
  geom_errorbar(aes(ymin = `2.5 %`, ymax = `97.5 %`), color = "black", width = 0.2) +
  geom_hline(yintercept = 0, linetype = 'dashed', color = "red") +
  xlab("Variable Name") + ylab("Estimated Coefficient and 95% CI") +
  ggtitle("Coefficient Plot: Model for Time Spent on Article Page") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggsave("emotion_timeonpage_model.png", height = 6, width = 8)
```

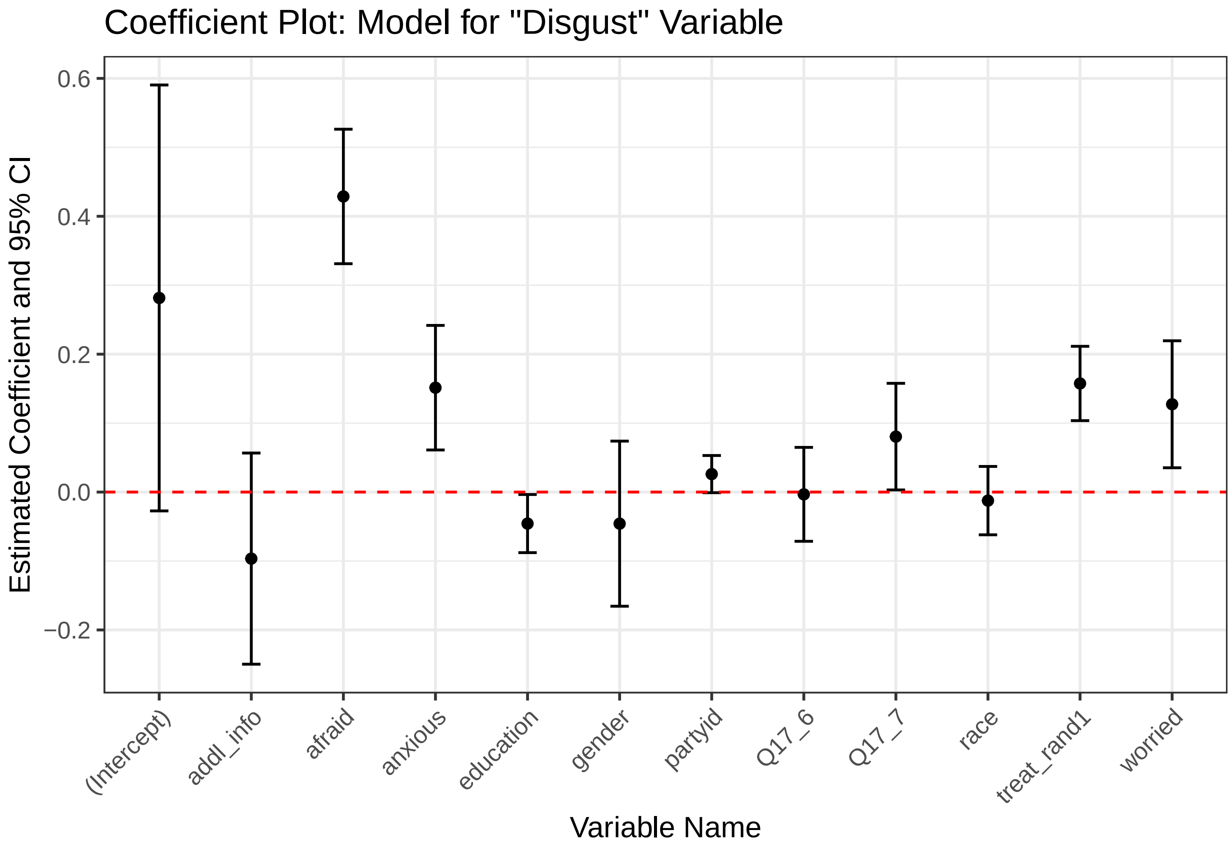
Blog Post

```r
# run regression model2
model2 <- lm(disgust ~ treat_rand1 + afraid + anxious + worried +
               addl_info + Q17_6 + Q17_7 + gender + race + education + partyid,
             data = study1)
summary(model2)
```

```
##
## Call:
## lm(formula = disgust ~ treat_rand1 + afraid + anxious + worried +
##     addl_info + Q17_6 + Q17_7 + gender + race + education + partyid,
##     data = study1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3252 -0.5521 -0.1343  0.5115  3.4131
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.281545   0.157400   1.789  0.07397 .
## treat_rand1  0.157498   0.027460   5.736  1.3e-08 ***
## afraid       0.428746   0.049690   8.628  < 2e-16 ***
## anxious      0.151379   0.046027   3.289  0.00104 **
## worried      0.127348   0.046921   2.714  0.00676 **
## addl_info   -0.096533   0.078043  -1.237  0.21642
## Q17_6       -0.003318   0.034707  -0.096  0.92387
## Q17_7        0.080396   0.039398   2.041  0.04156 *
## gender      -0.045848   0.061023  -0.751  0.45265
## race        -0.012409   0.025254  -0.491  0.62326
## education   -0.045705   0.021481  -2.128  0.03362 *
## partyid      0.026057   0.013741   1.896  0.05823 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.937 on 960 degrees of freedom
##   (28 observations deleted due to missingness)
## Multiple R-squared:  0.4468, Adjusted R-squared:  0.4404
## F-statistic: 70.48 on 11 and 960 DF,  p-value: < 2.2e-16
```

```r
# find confidence intervals and create dataframe for plotting
point2 <- summary(model2)$coefficients[,1]
ci2 <- confint(model2)
plotdata2 <- cbind(point2, ci2)
plotdata2 <- as.data.frame(plotdata2)
plotdata2 <- tibble::rownames_to_column(plotdata2, "variable")

# plot
plotdata2 %>%
  ggplot(aes(x = variable, y = point2)) +
  geom_point() +
  geom_errorbar(aes(ymin = `2.5 %`, ymax = `97.5 %`), color = "black", width = 0.2) +
  geom_hline(yintercept = 0, linetype = 'dashed', color = "red") +
  xlab("Variable Name") + ylab("Estimated Coefficient and 95% CI") +
  ggtitle("Coefficient Plot: Model for \"Disgust\" Variable") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggsave("emotion_disgust_model.png", height = 6, width = 8)
```