POLS 7012: Problem Set 1 (Answer Key)

Joe Ornstein

Due September 2, 2020

In this problem set, you will create an `R` script that performs some basic analysis of the 2019 ANES (American National Election Studies) Pilot Study. More information about that study is here. (Basically, they test a bunch of questions on a non-random opt-in Internet panel, so don't draw far-reaching conclusions from this dataset.)

Make sure to comment your code so that a reader will know what each line is doing, like this:

```
# Compute the median age
median(data$age)
```

When you're done, upload the `.R` file to eLC. Feel free to work with others in the class, but you must submit your own work.

Create an R Project

R Projects are a great way to organize your workflow. In a nutshell, they keep all your files in one place so that `R` knows where to look. See R4DS Chapter 8 for more detail. I recommend that whenever you start a new data analysis project, your first step should be to create an R Project.

In RStudio, click the "Create a project" button. Put it in a New Directory (where you can easily find it), and title it whatever you want.

Create a subfolder in your project folder called `data/` and put the the data file `anes_pilot_2019.RData` into that subfolder.

Now you're all set up!

Load The Data

The `load()` function loads an `.RData` file. To load the ANES data, run `load("data/anes_pilot_2019.RData")`. (Don't forget the quotation marks around the path.)

```
load('data/anes_pilot_2019.RData')
```

You should now have an object called `data` in your environment.

Summarize The Data

- The `nrow()` function counts the number of rows (i.e. observations) in a dataframe. How many observations are in this dataset?
- The `ncol()` function counts the number of columns (i.e. variables) in a dataframe. How many variables are in this dataset?
- What are the `names()` of the variables?

```
# How many observations are there?
nrow(data)
```

```
## [1] 3165
```

```
# How many variables are there?
ncol(data)
```

```
## [1] 900
```

```
# What are the variables named?
names(data)[1:10] # just print the first 10 for the answer key
```

```
##  [1] "version"     "caseid"      "weight"      "weight_spss" "form"
##  [6] "follow"      "reg1a"       "reg1b"       "liveurban"   "youthurban"
```

Clean Up The Data

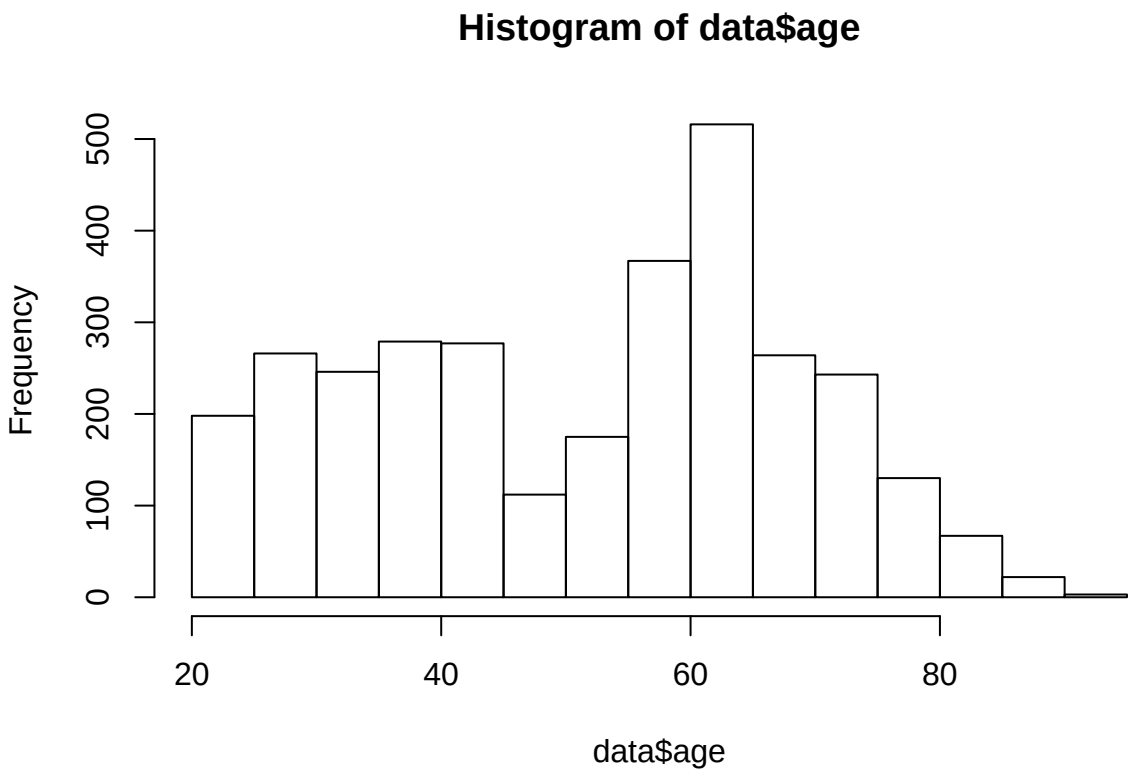
- There is a variable for birth year, called `birthyr`, but no variable for age. Let's fix that. Create a variable called `age`, and set it equal to the current year minus birth year.
- What is the `median()` age of our survey respondents?
- Create a histogram of age with the `hist()` function. (We'll learn how to make prettier ones later.)

```
# Create an 'age' variable
data$age <- 2020 - data$birthyr

# What is the median age?
median(data$age)
```

```
## [1] 56
```

```
# Plot a histogram of age
hist(data$age)
```



A lot of the variables have **missing values**, and it will trip up your data analysis if you don't know where those missing values are.

- Create a `table()` of the variable `vote16`. How many respondents skipped this question (code = -1)?
- In `R`, we typically represent missing values with `NA`. We can recode those values with the power of **indexing**. Try this: `data$vote16[data$vote16 == -1] <- NA`. (Read that line of code as "get the vote16 variable, but only the entries where it equals -1, and assign those entries the value `NA`).
- Create the table again. What happened?

```
# Table of reported voting in 2016 (-1 = missing, 1 = Donald Trump, 2 = Hillary Clinton, 3 = someone el
table(data$vote16)
```

```
##
##   -1    1    2    3
##  603 1172 1110  280
```

```
# Replace the missing values with NA
data$vote16[data$vote16 == -1] <- NA

# Show that table again. The NA values are omitted!
table(data$vote16)
```

```
##
##    1    2    3
## 1172 1110  280
```

Explore The Data

- Create a `table()` of the variable `liveurban`. Where are our respondents most likely to live? See the ANES codebook to learn what the labels mean.
- Create a two-way table (just the `table` function, but with two inputs) with `liveurban` and `vote20jb`. Who are the rural respondents in our sample most likely to vote for? The urban respondents?

```
# Where do our respondents live?
table(data$liveurban)
```

```
##
##    1    2    3    4
##  643  589 1117  816
```

```
# Two-way table: place of residence vs. vote intention
table(data$liveurban, data$vote20jb)
```

```
##
##       1   2   3   4
##   1 314 204  64  61
##   2 259 221  58  51
##   3 465 462 107  83
##   4 235 401  92  88
```

- Skim the codebook and find three variables that you think are interesting. Summarize each one, either with a `table()` for categorical variables or the `mean()` for continuous variables.

```
# Two-way table: current place of residence and place of residence growing up
table(data$liveurban, data$youthurban)
```

```
##
##       1   2   3   4
##   1 321 126  92 104
##   2 105 281  97 106
##   3 128 199 587 203
##   4  94 122 124 476
```

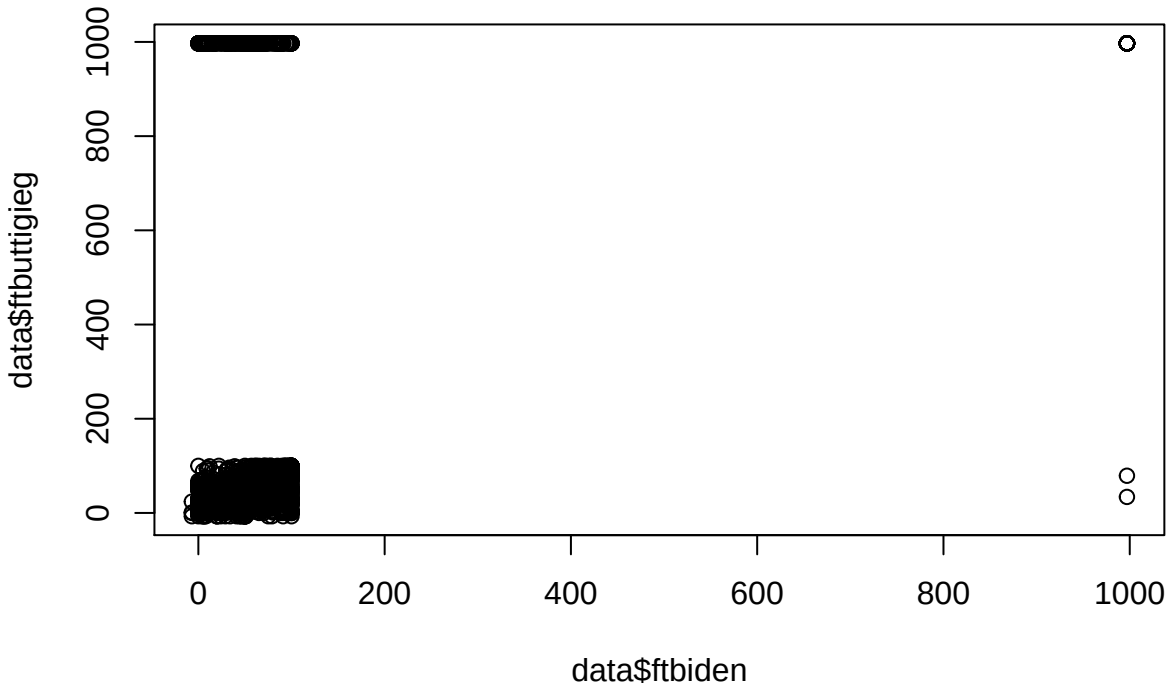
```
# Feeling thermometer: Pete Buttigieg vs. Joe Biden
median(data$ftbiden)
```

```
## [1] 44
```

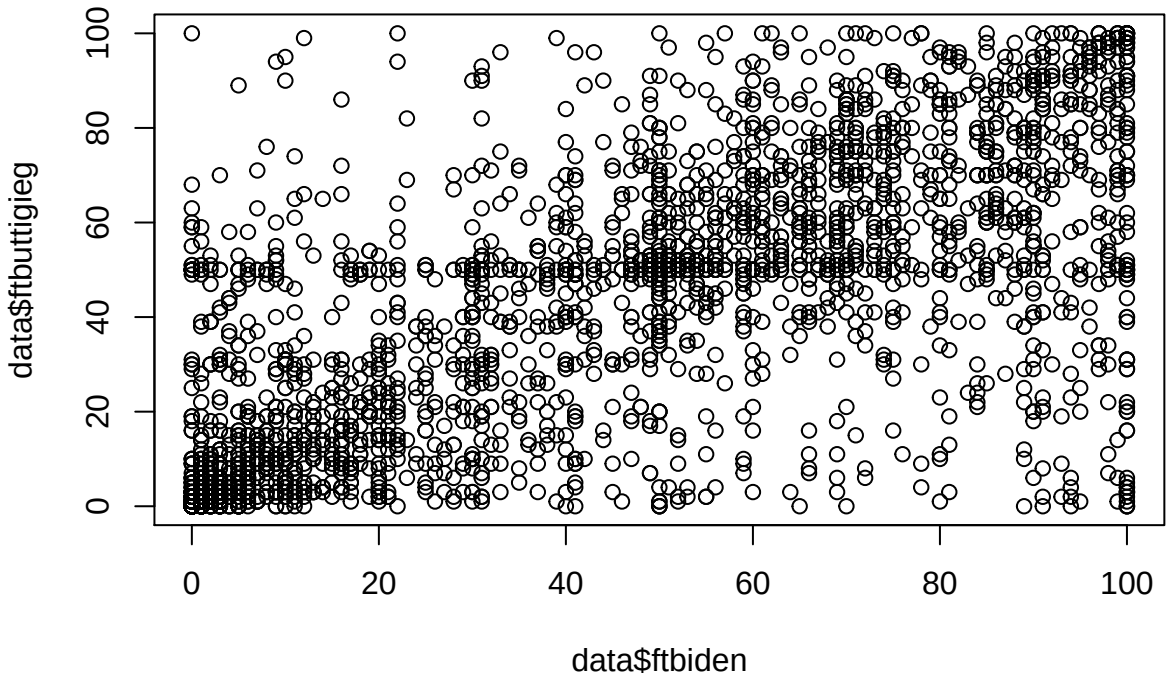
```
median(data$ftbuttigieg)
```

```
## [1] 48
```

```
plot(data$ftbiden, data$ftbuttigieg)
```




```
# Look at that: missing data is coded as 997! Boy if you didn't know that, you could come to some weird
data <- subset(data, ftbiden < 101 & ftbuttigieg < 101 &
                 ftbiden >= 0 & ftbuttigieg >= 0)
plot(data$ftbiden, data$ftbuttigieg)
```




```
# Conspiracy theories vs. political knowledge.
# Are you more likely to believe that business and politics are secretly controlled by a single group i
table(data$pk_germ_correct, data$conspire1)
```

```
##
##      -7   1   2   3   4   5
##   0   1  60 139 392 225 159
##   1   0 139 453 635 361 283
```

```
data %>%
  group_by(pk_germ_correct) %>%
  summarise(mean_conspiracy = mean(conspire1))
```

```
## # A tibble: 2 x 2
##   pk_germ_correct mean_conspiracy
##             <dbl>           <dbl>
## 1               0            3.28
## 2               1            3.10
```

```
# Maybe slightly? We'll learn how to do hypothesis tests later :-)
```