Data Exploration: Voting and Social Pressure

Yao Yu

September 2, 2021

Welcome to this initial Data Exploration Assignment. This assignment is ungraded and is simply intended to give you a sense of what we will be doing in class and to be sure you have all the tools in place to successfully do these assignments. *You should be sure that you can use R to answer at least one of the questions below. If not, seek assistance from your Teaching Fellow before Thursday, September 9.*

Note that the actionable part of each question is **bolded**.

To get started, save this `.rmd` file and the data `ggl_2008_data_data.csv` to the same directory (a folder) on your computer. We suggest creating a drectory specifically for this class, say "Gov1372" and then a directory for each week, say "Week1". Whatever you call it, save your `.rmd` and the data to the same location.

Data Overview: We will work with the data collected by Gerber, Green, and Larimer as part of their experiment on Michigan voters in the context of the 2006 primary election, as described by Sasha Issenberg in *The Victory Lab*. In this study, the authors contacted 344,084 voters in Michigan, whose addresses they obtained off the official lists of voters. These lists include the names, addresses, and histories of voter participation of every voter in the state. The authors wanted to understand if they could stimulate voter participation in 2006 by applying social pressure. To do so, they randomly assigned each voter to one of five conditions, four of which were treatment conditions and one of which was a control condition. Those in the four treatment conditions were sent one of four mailers encouraging them to vote. They differed in their messages, which were as follows:

- **Civic Duty**: This was the baseline condition - all other conditions included the language from this one and added to it. Voters were told, "Remember your rights and responsibilities as a citizen. Remember to vote."
- **Hawthorne**: Voters were told "YOU ARE BEING STUDIED!"[1]
- **Self**: Voters were sent a mailer that included the past voting history of all individuals in the household. They were told that they would be sent another mailer after the election with whether or not they voted, allowing members of the household to discover their voting history.
- **Neighbors**: Voters were sent a mailer that included the past voting history of the individual's neighbors (including the individual themselves). They were told that they and their neighbors would be send another mailer with updated information after the election, thereby publicizing their decision on whether to vote.

Data Details:

- File Name: `ggl_2008_data_data.csv`
- Source: These data are adapted from the replication data for Gerber, Green, and Larimer (2008).

[1] The treatment is called "Hawthorne" because of the Hawthorne Effect, which is the phenomenon of individuals modifying their behavior when they know that they are being observed. This came from the famous reanalysis of a flawed study of worker productivity at the Hawthorne Works, an electrical plant in Cicero, Illinois.

Variable Name	Variable Description
`sex`	Either "male" or "female"
`age`	Age in 2006
`treatment`	The treatment condition the individual was assigned to: "Control", "Civic Duty", "Hawthorne", "Self", or "Neighbors"
`voted`	Whether or not the individual voted in 2006 (this is the outcome of the experiment): 1 if voted, 0 otherwise
`g2004_mean`	The proportion of eligible individuals in the household who voted in the 2004 general election (e.g. 0.25 if 1 of 4 voters in the household voted in 2004).

Initial setup

Loading packages: Let's get started. For those unfamiliar with R, the very first thing to do when setting up your code is to load the packages you will be using.[2]

You probably won't know before you start coding which packages you will need, but in any case it is good practice to load packages before other code. In our case for this problem set, we will need `readr`, `ggplot2`, and `dplyr`.

Before loading packages, they must be installed on your computer. You can find more details about this process on Modern Dive Section 1.3.

The instructions at that link are primarily for the point-and-click method of installing packages, but it's also important to know how to do it via the command line. Some may find it easier as well. To install packages via the command line, simply run `install.packages("your_package")` in R, making sure the package name is in quotes. Then you can run `library(your_package)` to load it into R. Note that the package doesn't need to be in quotes inside the `library()` function, but it can be if you like.

The code below loads the packages. Note that we will always include the functions we used from each package in a comment next to the code we use to load the library.

```
#library(readr) # read_csv()
#library(ggplot2) # ggplot()
#library(dplyr) # group_by(), summarize()
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gt) # gt()
```

Loading the data: After loading the packages we need, it's time to read the data into R. But there's one last step! Before you try to read data, it is a good idea to tell R where on your computer you're working. To do that, you need to set your working directory. Remember, "directory" is just a computer science term for a folder on your computer. By setting your working directory, you're telling R the folder in which to look for

[2] One of the reasons R is such a widely used language is that there is a whole community that develops packages, which add functionality to the language. You can think of a package as just a collection of useful functions that aren't available in base R.

files. Usually it's best practice to set your working directory to the directory that your code is in. To do that, just go to the toolbar at the top of your screen, select "Session", hover over "Set Working Directory", and select "To Source File Location".

You can check your current working directory by running `getwd()` with nothing in the parentheses. Try running `getwd()` in the console to make sure your current working directory is the one where you have this file saved. Make sure that you have downloaded the data for this assignment into that same directory for the code below to work. This works because by setting the working directory you told R the folder where it will find the data.

Question 1

Now let's read the data into R. When doing data science, this is often the first thing to do (after loading your packages!). If you read in the data correctly, you should see that you have 344,084 rows (one for each voter) and 5 columns (one for each variable) in the data. **Fill in the file name for the data in the code below.**

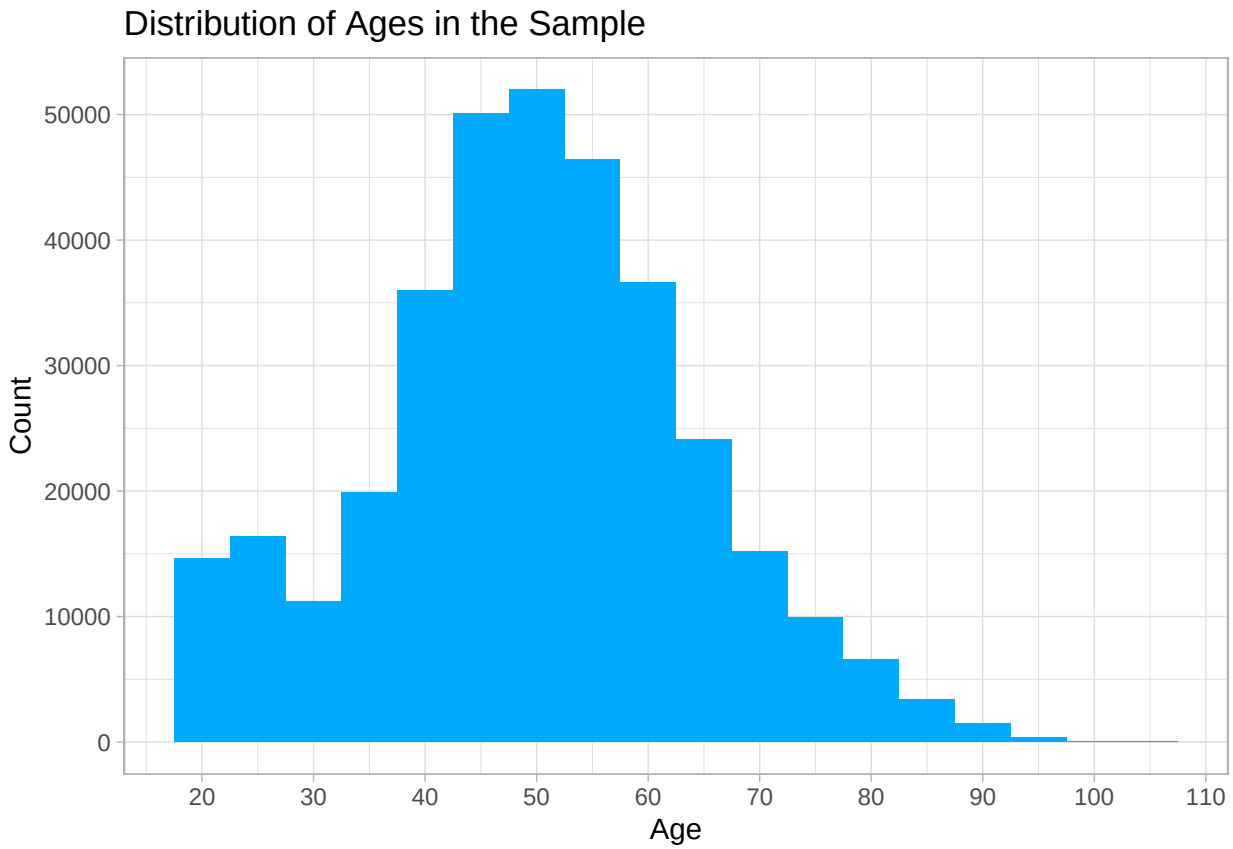
```
ggl <- read_csv("data/ggl_2008_data.csv", col_types = cols(
  sex = col_character(),
  age = col_double(),
  treatment = col_character(),
  voted = col_double(),
  g2004_mean = col_double()
))
```

Question 2

Before we start analyzing the experiment, lets first look at the distribution of age in the data set. When analyzing data, it's always a good idea to check and see if the data look like you would expect and make sure there aren't any strange values. For example, we shouldn't have anyone younger than 18 (the minimum age to vote) or older than 122 (the age of the oldest person ever to live).

As an example of how this can be done, here is how to make a histogram using the package `ggplot2`. It is a histogram of `g2004_mean`. **Modify the code below to make a histogram of age in the sample, and make sure to edit the x-axis label. Comment on whether you think the distribution seems reasonable and why.**

```
# modify this code to make a histogram of age in the sample
ggplot(data = ggl, mapping = aes(x = age)) + # mapping = aes() controls which
                                               # variables go into your plots
  geom_histogram(binwidth = 5, fill = "#00abfd") + # geoms, like geom_histogram, control the type of pl
  labs(title = "Distribution of Ages in the Sample", x = "Age", y = "Count") +
  scale_x_continuous(breaks = c(20, 30, 40, 50, 60, 70, 80, 90, 100, 110)) +
  theme_light()
```



The distribution looks reasonable as I do not see any ages below 18 or older than 122. However, I question if this distribution accurately reflects the actual distribution of registered voters in Michigan. If hypothetically the actual distribution is different from this sample distribution of ages, then there could be potentially coverage error in this survey.

Question 3

Now let's do some analysis. We want to find the proportion of individuals in each condition who voted. As an example of how to do this, the code below calculates the proportion of males and females who voted below. **Modify the code to find the proportion of individuals who voted in each of the 5 experimental conditions. For an extension to this problem, answer this question using `tapply()` or the `dplyr` package. Did you get the same proportions as Gerber, Green, and Larimer found? Look back in chapter 7 of *The Victory Lab* or see Gerber et al.'s (2008) paper to find their results. Interpret what you have found.**

```
# mean(ggl$voted[ggl$sex == "male"]) # the proportion of men who voted
# mean(ggl$voted[ggl$sex == "female"]) # the proportion of women who voted

ggl %>%
  group_by(treatment) %>%
  summarize(prop_voted = sum(voted) / n(), .group = "drop") %>%
  arrange(prop_voted) %>%
  gt()
```

treatment	prop_voted	.group
Control	0.2966383	drop

Civic Duty	0.3145377	drop
Hawthorne	0.3223746	drop
Self	0.3451515	drop
Neighbors	0.3779482	drop

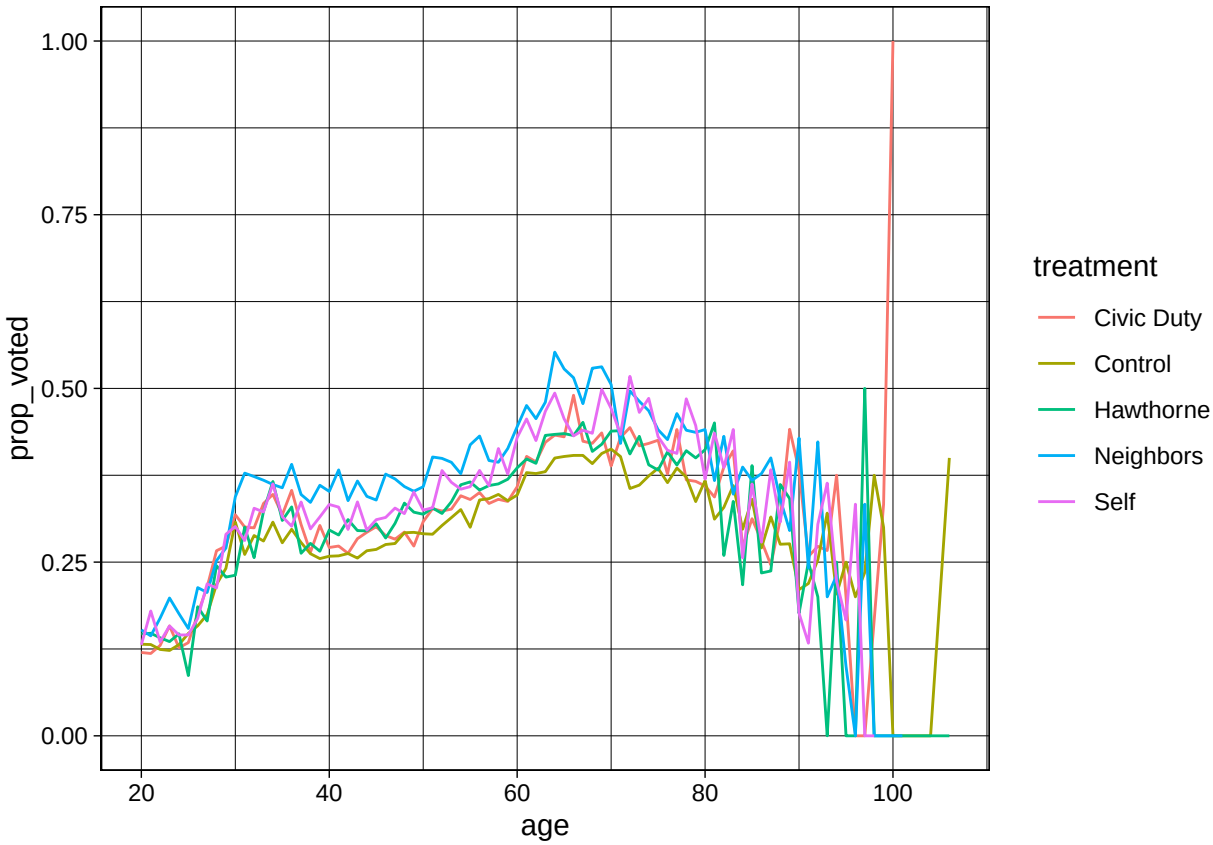
Question 4

Lastly, is there anything else you can investigate with this data? For example, was the experiment more effective for some people rather than others?

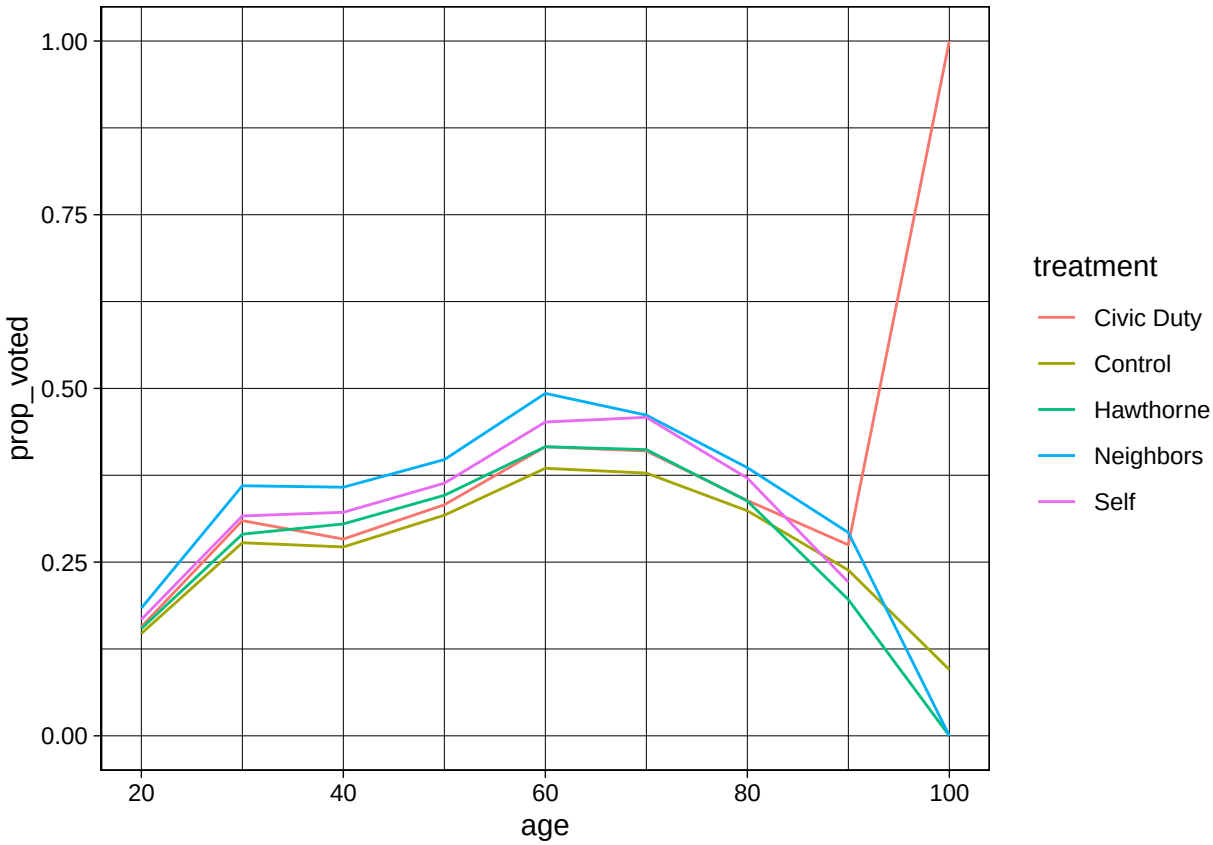
```
# Break-down by sex
ggl %>%
  group_by(sex, treatment) %>%
  summarize(prop_voted = sum(voted) / n(), .groups = "drop") %>%
  arrange(prop_voted) %>%
  pivot_wider(names_from = sex, values_from = prop_voted) %>%
  gt()
```

treatment	female	male
Control	0.2904558	0.3027947
Civic Duty	0.3063402	0.3227411
Hawthorne	0.3172472	0.3274817
Self	0.3417483	0.3485490
Neighbors	0.3713553	0.3845429

```
# Ages
ggl %>%
  group_by(age, treatment) %>%
  summarize(prop_voted = sum(voted) / n(), .groups = "drop") %>%
  ggplot(aes(x = age, y = prop_voted, color = treatment)) +
  geom_line() +
  theme_linedraw()
```




```
# Ages binned
ggl %>%
  mutate(age = case_when(
    age < 30 ~ 20,
    age < 40 ~ 30,
    age < 50 ~ 40,
    age < 60 ~ 50,
    age < 70 ~ 60,
    age < 80 ~ 70,
    age < 90 ~ 80,
    age < 100 ~ 90,
    age < 110 ~ 100,
  )) %>%
  group_by(age, treatment) %>%
  summarize(prop_voted = sum(voted) / n(), .groups = "drop") %>%
  ggplot(aes(x = age, y = prop_voted, color = treatment)) +
  geom_line() +
  theme_linedraw()
```


1.00
0.75
0.50
0.25
0.00
prop_voted
20
40
60
80
100
age
treatment
Civic Duty
Control
Hawthorne
Neighbors
Self