Problem set 2

Your name here

Due 10/11/2022 at 5pm

NOTE: Start with the file `ps2_2022.Rmd` *(available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F22/tree/main/assignments). Modify that file to include your answers. Make sure you can "knit" the file (e.g. in RStudio by clicking on the* `Knit` *button). Submit both the Rmd file and the knitted PDF via Canvas*

Question 1: Probability

Consider the random process of flipping a weighted coin three times, where the probability of getting heads on any single flip is 0.8.

(1a) Describe the sample space, Ω.

$$\Omega = \{TTT, TTH, THT, HTT, THH, HTH, HHT, HHH\}$$

(1b) The random variable X that we're interested is the number of heads that we get from our random process. Write out the probability mass function for this random variable. $f_X(x)$.

$$f(x) = \begin{cases} 1/125 & x = 0 \\ 12/125 & x = 1 \\ 48/125 & x = 2 \\ 64/125 & x = 3 \\ 0 & \text{otherwise} \end{cases}$$

OR

x	$P(X = x)$
0	1/125
1	12/125
2	48/125
3	64/125

(1c) Write out code to simulate this random process, where the output is a single realization of the random variable (i.e., a number that represents the number of heads in your coin flips).

NOTE3: I set a random seed here, so that every time you recompile your assignment, you'll get the same number. For analyses that involve sampling or random processes, it is really important to set a random seed so that you can get reproducible results. Feel free to change the seed number to anything you want. In general you only need to set your random seed ONCE per script.

```r
set.seed(60637)

X <- c(0, 1, 2, 3)
probs <- c(1/125, 12/125, 48/125, 64/125)

sample(x = X,
       size = 1,
       prob = probs)
```

```
## [1] 2
```

(1d) Now run your random process so you sample from it 10,000 times [PLEASE DON'T OUTPUT ALL 10,000 OBSERVATIONS IN YOUR HOMEWORK, just save it to an R object]. What is the average number of heads across these 10k observations? This is the sample mean for a given sample.

```r
result_n <- sample(x = X,
                   size = 10000,
                   prob = probs,
                   replace = TRUE)

mean(result_n)
```

```
## [1] 2.4005
```

(1e): Write your own function called `my_summary()` that outputs a vector which is the minimum, maximum, and mean of a vector. Apply your function to your size 10k sample that you saved in the last problem.

```r
my_summary <- function(x){
  return(c(min = min(x), max = max(x), mean = mean(x)))
}

my_summary(result_n)
```

```
##    min    max   mean
## 0.0000 3.0000 2.4005
```

Question 2:

Using the same random process of flipping three biased coins, code the random variable Y as 1 if we get three heads, and 0 otherwise.

(2a) Write out the probability mass function for this random variable Y.

y	$P(Y = y)$
0	61/125
1	64/125

(2a) Write out the joint probability mass function for the joint distribution of X and Y.

$$f(x,y) = \begin{cases} 1/125 & x=0, y=0 \\ 12/125 & x=1, y=0 \\ 48/125 & x=2, y=0 \\ 64/125 & x=3, y=1 \\ 0 & \text{otherwise} \end{cases}$$

OR

x	y	$P(X=x, Y=y)$
0	0	1/125
1	0	12/125
2	0	48/125
3	1	64/125

(2b) Write out the probability mass function for this random variable X *conditional* on Y.

$$f(x|y) = \begin{cases} 1/61 & x=0|y=0 \\ 12/61 & x=1|y=0 \\ 48/61 & x=2|y=0 \\ 1 & x=3|y=1 \\ 0 & \text{otherwise} \end{cases}$$

OR

x	y	$P(X=x\|Y=y)$
0	0	1/7
1	0	3/7
2	0	3/7
3	1	1

(2c) Are X and Y *independent* random variables? Show why or why not. *Hint: See definition 1.3.16 in Aronow & Miller.*

No. $f(x,y) \neq f_X(x)f_Y(y)$.

Question 3: Programming (US presidential election results, again)

Download the file "tidy_county_pres_results.csv.zip" from the repository (https://github.com/UChicago-pol-methods/IntroQSS-F22/tree/main/data), unzip it, and put the CSV file in the same directory as your Rmd file.

Then load the data:

```
library(tidyverse)
df <- read_csv("tidy_county_pres_results.csv")
```

For each US county (uniquely identified by `FIPS` and labeled with `county` and `state`) in each presidential election `year`, we have the total number of votes cast (`total_vote`), number of votes for the Democratic candidate (`dem_vote`), and number of votes for the Republican candidate (`rep_vote`).

(3a) Add a variable called `other_vote_share`, which is the proportion of votes cast for candidates other than the Democrat and the Republican.

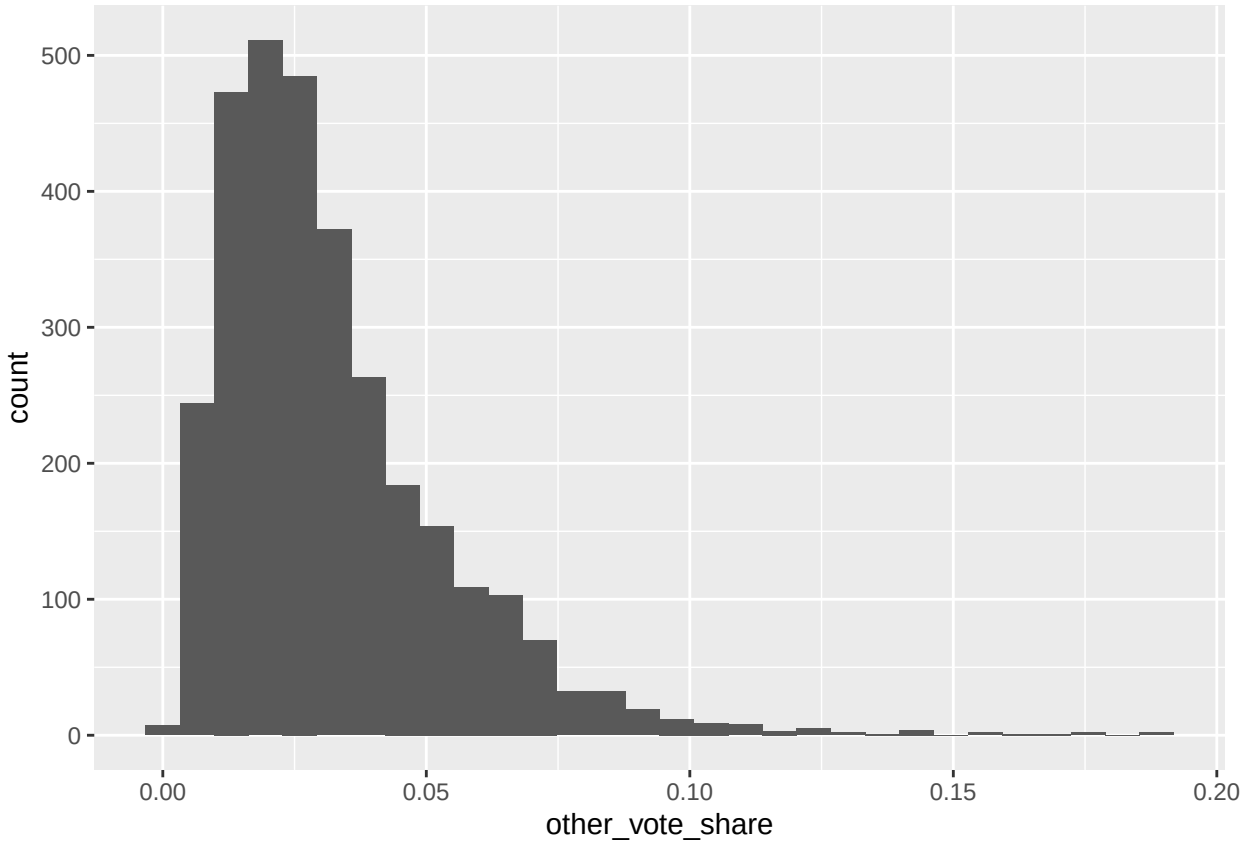
```
df %>%
  mutate(other_vote_share = 1 - (dem_vote + rep_vote)/total_vote) -> df2
```

(3b) Show a histogram of `other_vote_share` in 2000.

```
df2 %>%
  filter(year == 2000) %>%
  ggplot(aes(x = other_vote_share)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



(3c) Identify the counties with the highest `other_vote_share` in 2000. Output a table showing the county name, state, and `other_vote_share` for the seven counties with the highest `other_vote_share` in 2000. (Don't worry about making the table look nice; just produce the raw R output.)

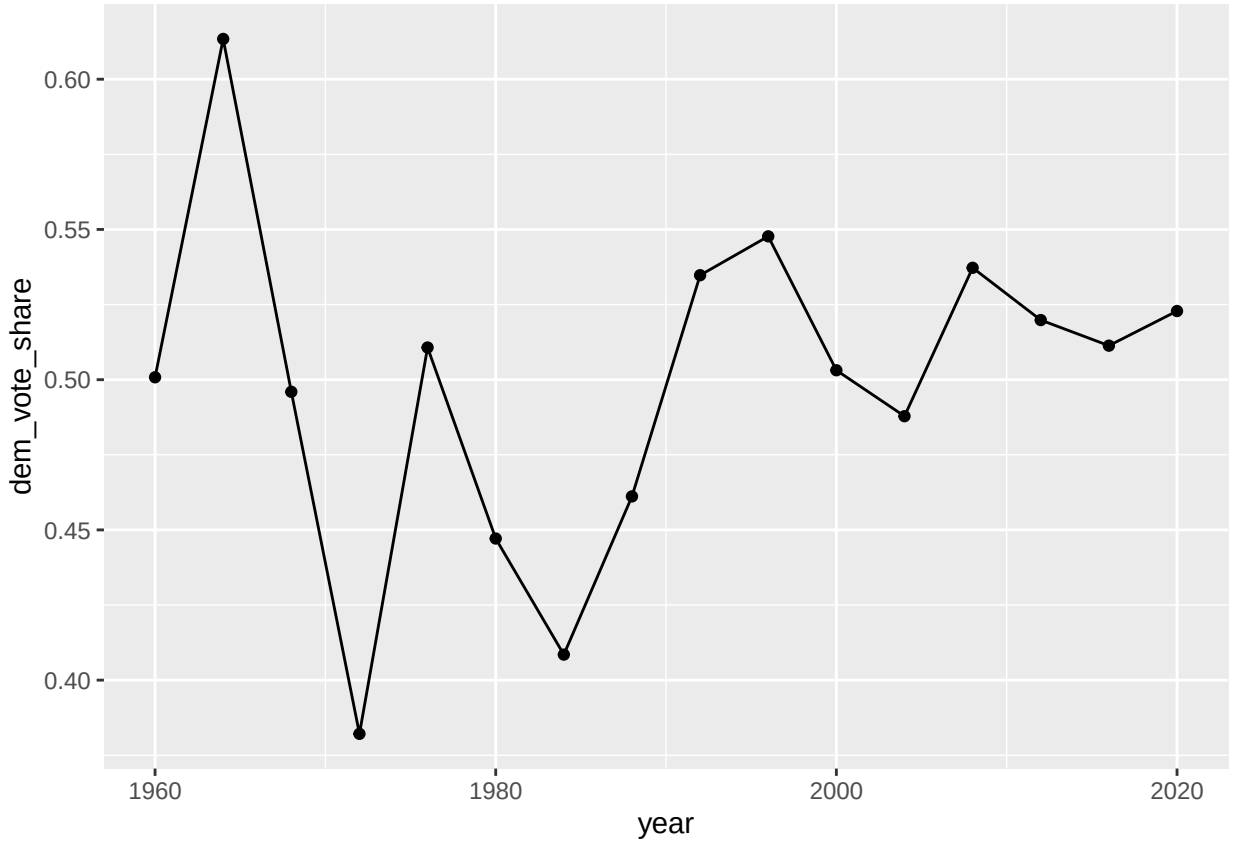
```
df2 %>%
  filter(year == 2000) %>%
  arrange(desc(other_vote_share)) %>%
  select(county, state, other_vote_share) %>%
  head()
```

```
## # A tibble: 6 x 3
##   county     state other_vote_share
```

```
##   <chr>      <chr>           <dbl>
## 1 Jefferson  IA              0.190
## 2 San Miguel CO              0.189
## 3 San Juan   CO              0.177
## 4 Grand      UT              0.175
## 5 Missoula   MT              0.169
## 6 Mendocino  CA              0.160
```

(3d) Using `group_by()` and `summarize()`, produce and store a new `tibble` showing the two-party vote share for the Democrat in each election year. ("Two-party vote share for the Democrat" is the votes for the Democrat divided by the votes for either the Democrat or the Republican.) Use it to make a plot showing the Democrats' two-party vote share (vertical axis) across years (horizontal axis).

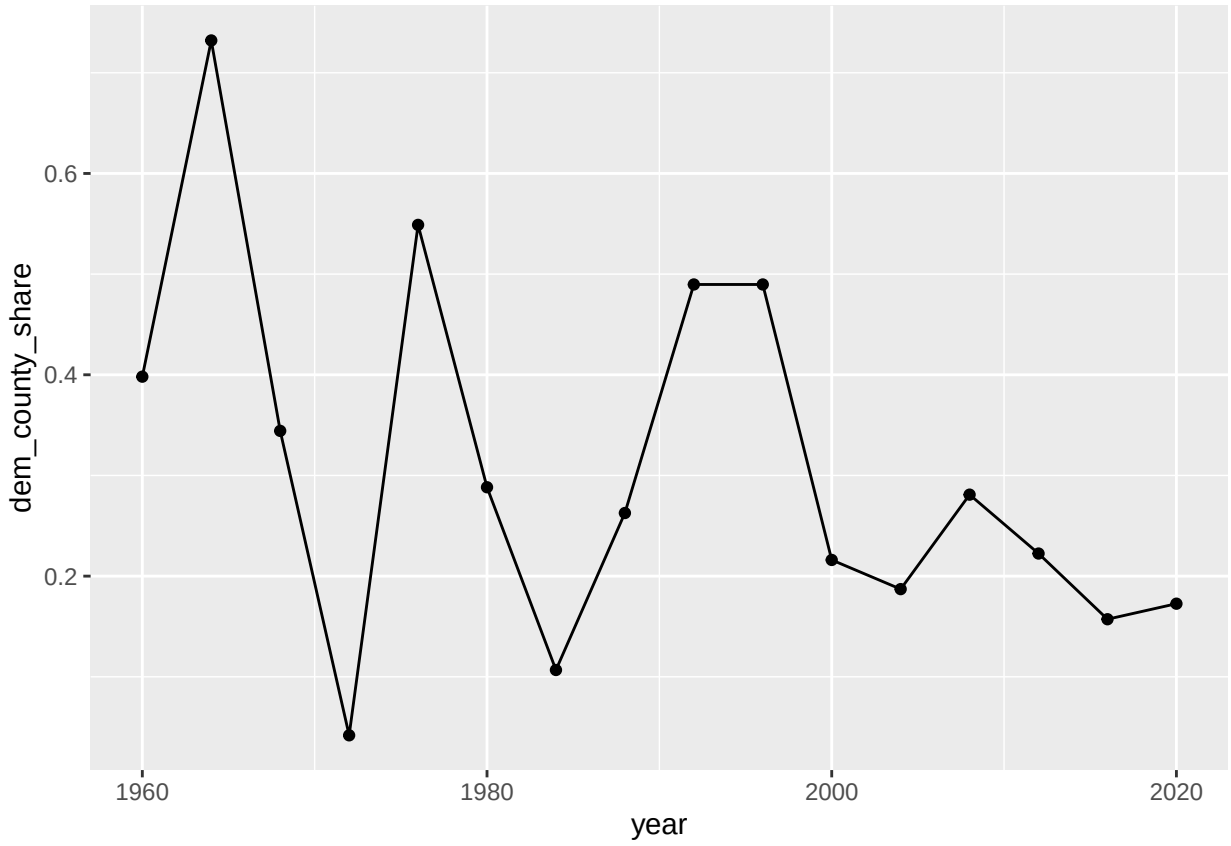
```
df %>%
  group_by(year) %>%
  summarize(dem_sum = sum(dem_vote, na.rm = T),
            demrep_sum = sum(dem_vote + rep_vote, na.rm = T)) %>%
  mutate(dem_vote_share = dem_sum/demrep_sum) -> df_vote_share

df_vote_share %>%
  ggplot(aes(x = year, y = dem_vote_share)) +
  geom_line() +
  geom_point()
```



(3e) Using `group_by()` and `summarize()`, produce and store a new `tibble` showing the proportion of counties in which the Democrat got more votes than the Republican in each election year. Use it to make a plot showing the share of counties won by the Democrat (vertical axis) across

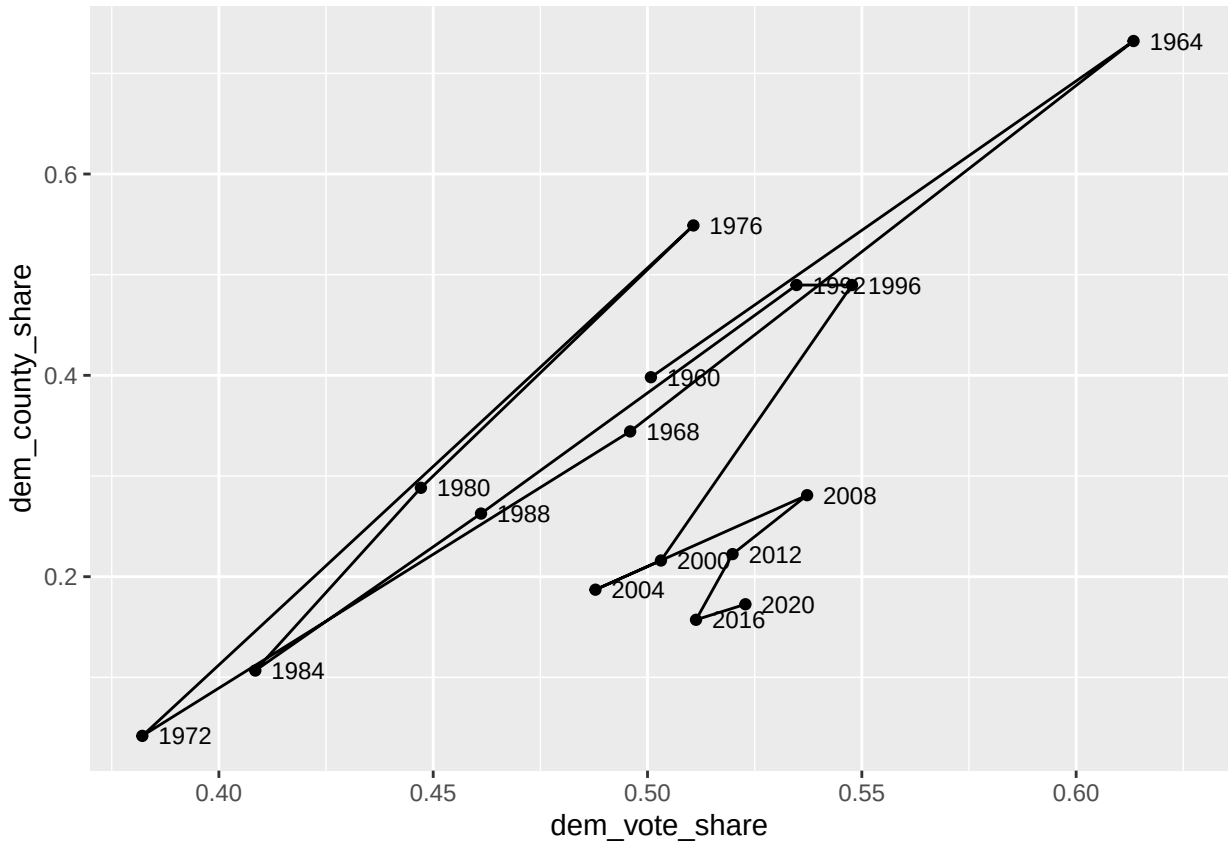
years (horizontal axis).

```
df %>%
  mutate(dem_beats_rep = dem_vote > rep_vote) %>%
  group_by(year) %>%
  summarize(dem_county_share = mean(dem_beats_rep, na.rm = T)) -> df_county_share

df_county_share %>%
  ggplot(aes(x = year, y = dem_county_share)) +
  geom_line() +
  geom_point()
```



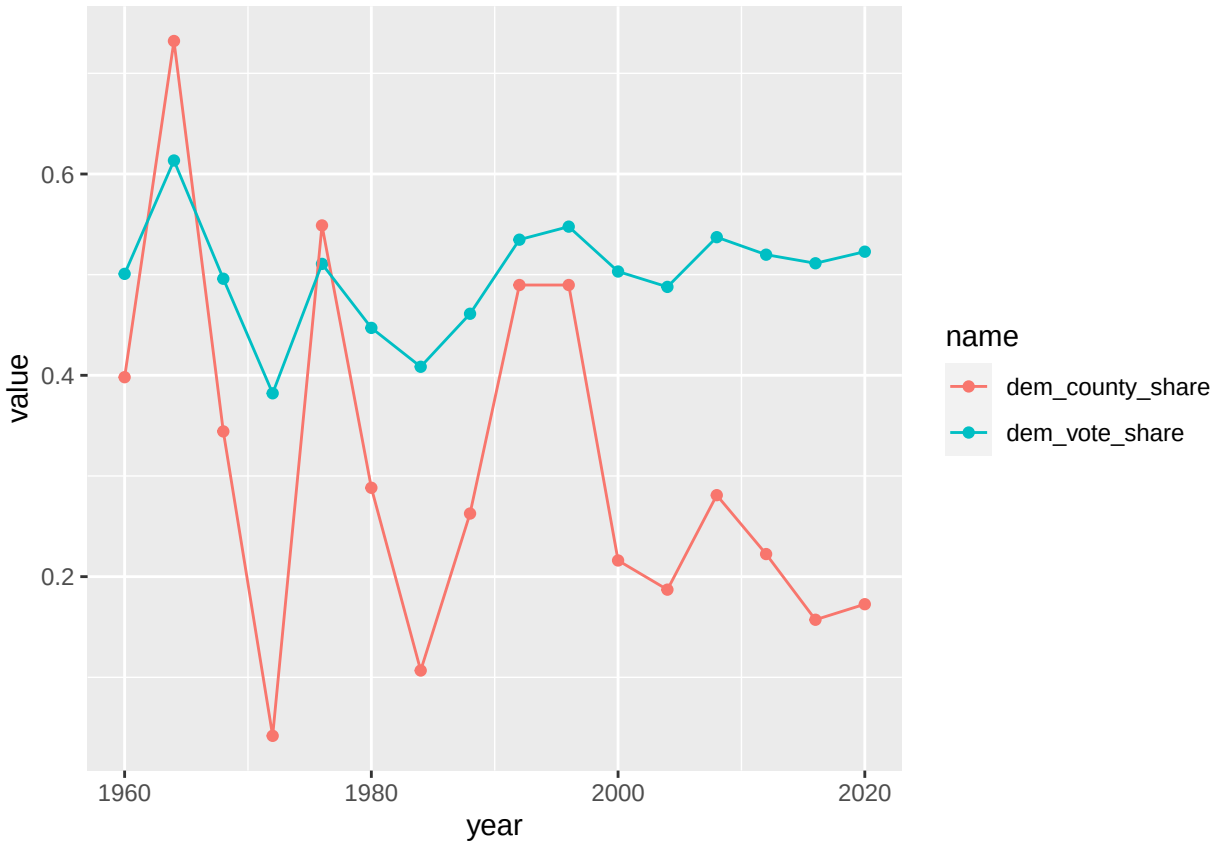
(3f) Use `left_join()` to merge the two tibbles (one with county share, the other with vote share) and store the result. Use this new tibble to plot the Democratic county share (vertical axis) against the Democratic vote share (horizontal axis) over time, as in the last problem set.

```
df_county_share %>%
  left_join(df_vote_share, by = "year") -> df_joined

df_joined %>%
  ggplot(aes(x = dem_vote_share, y = dem_county_share, label = year)) +
  geom_path() +
  geom_point() +
  geom_text(nudge_x = .01, size = 3)
```



(3g) Use `pivot_longer()` to convert the tibble created in the last question to a format appropriate for plotting both the Democratic vote share and the Democratic county share (vertical axis) against the year (horizontal axis) as on the last problem set. Make that plot.

```
df_joined %>%
  pivot_longer(cols = c(dem_county_share, dem_vote_share)) %>%
  ggplot(aes(x = year, y = value, col = name)) +
  geom_point() +
  geom_line()
```



Question 4: independent project data

Choose a dataset that you could use for your independent project. As noted last week, it should have many observations (but not too many) and many variables. And it should be interesting to you! Load the data and make a figure using tools we have learned in class.