Lab 4

Alyssa Andrichik

Math 241, Week 5

```
# Put all necessary libraries here
library(tidyverse)
library(rnoaa)
library(rvest)
library(httr)
```

Due: Friday, March 5th at noon

Goals of this lab

1. Practice grabbing data from the internet.
2. Learn to navigate new R packages.
3. Grab data from an API (either directly or using an API wrapper).
4. Scrape data from the web.

Problem 1: Predicting the (usually) predictable: Portland Weather

In this problem let's get comfortable with extracting data from the National Oceanic and Atmospheric Administration's (NOAA) API via the R API wrapper package `rnoaa`.

You can find more information about the datasets and variables here.

```
library(rnoaa)
```

a. First things first, go to this NOAA website to get a key emailed to you. Then insert your key below:

```
options(noaakey = "uYGvtRFZvFXSjMZGASDiPkJnKTERxmwL")
```

b. From the National Climate Data Center (NCDC) data, use the following code to grab the stations in Multnomah County. How many stations are in Multnomah County?

```
stations <- ncdc_stations(datasetid = "GHCND",
                          locationid = "FIPS:41051")

mult_stations <- stations$data
```

There are 25 stations in Multnomah County.

c. For 2021, grab the precipitation data and the snowfall data for site `GHCND:US1ORMT0006`. Leave in `eval = FALSE` as we are going to write the data to a csv in the next part.

```
# First fill-in and run to following to determine the
# datatypeid
ncdc_datatypes(datasetid = "GHCND",
               stationid = "GHCND:US1ORMT0006")
# Now grab the data using ncdc()
```

```
precip_se_pdx <- ncdc(datasetid = "GHCND", datatypeid = "PRCP",
                      startdate = "2021-01-01",
                      enddate = "2021-02-28",
                      stationid = "GHCND:US1ORMT0006",
                      limit = 1000)
precip <- precip_se_pdx$data
snow_se_pdx <- ncdc(datasetid = "GHCND", datatypeid = "SNOW",
                      startdate = "2021-01-01",
                      enddate = "2021-02-28",
                      stationid = "GHCND:US1ORMT0006",
                      limit = 1000)
snow <- snow_se_pdx$data
```

d. What is the class of `precip_se_pdx` and `snow_se_pdx`? Grab the data frame nested in each and create a new dataset called `se_pdx_data` which combines the data from both data frames using `bind_rows()`. Write the file to a CSV.

```
se_pdx_data <- bind_rows(precip_se_pdx$data, snow_se_pdx$data)

write_csv(se_pdx_data, file = "se_pdx_data.csv")
```

```
se_pdx_data_new <- read_csv("se_pdx_data.csv")
```

Both are characters.

e. Use `ymd_hms()` in the package `lubridate` to wrangle the date column into the correct format.

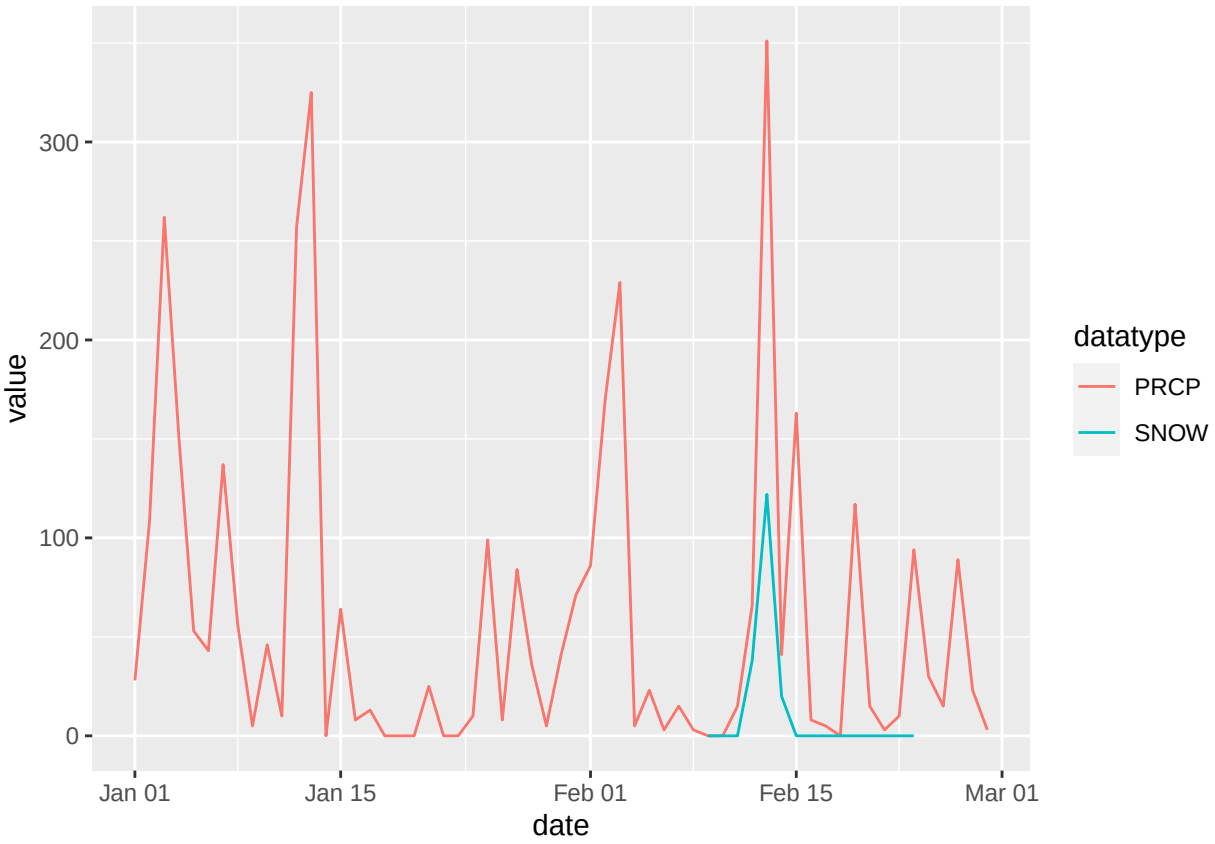
```
library(lubridate)
```

```
se_pdx_data_new$date <- ymd(se_pdx_data_new$date)
class(se_pdx_data_new$date)
```

```
## [1] "Date"
```

f. Plot the precipitation and snowfall data for this site in Portland over time. Comment on any trends.

```
ggplot(se_pdx_data_new, mapping = aes(x = date, y = value, color = datatype)) +
  geom_line()
```



When in snowed in the middle of February, there is a peak in the precipitation line as well meaning that it snowed and rained at the same time. Overall, it does not snow often in Portland, but rains quite a bit.

Problem 2: From API to R

For this problem I want you to grab web data by either talking to an API directly with `httr` or using an API wrapper. It must be an API that we have NOT used in class yet.

Once you have grabbed the data,

- Write the data to a csv file.

- Make sure the code to grab the data and write the csv is in an `eval = FALSE` r chunk.
- In an `eval = TRUE` r chunk, do any necessary wrangling to graph it and/or produce some relevant/interesting/useful summary statistics.
- Draw some conclusions from your graph and summary statistics.

API Wrapper Suggestions for Problem 2

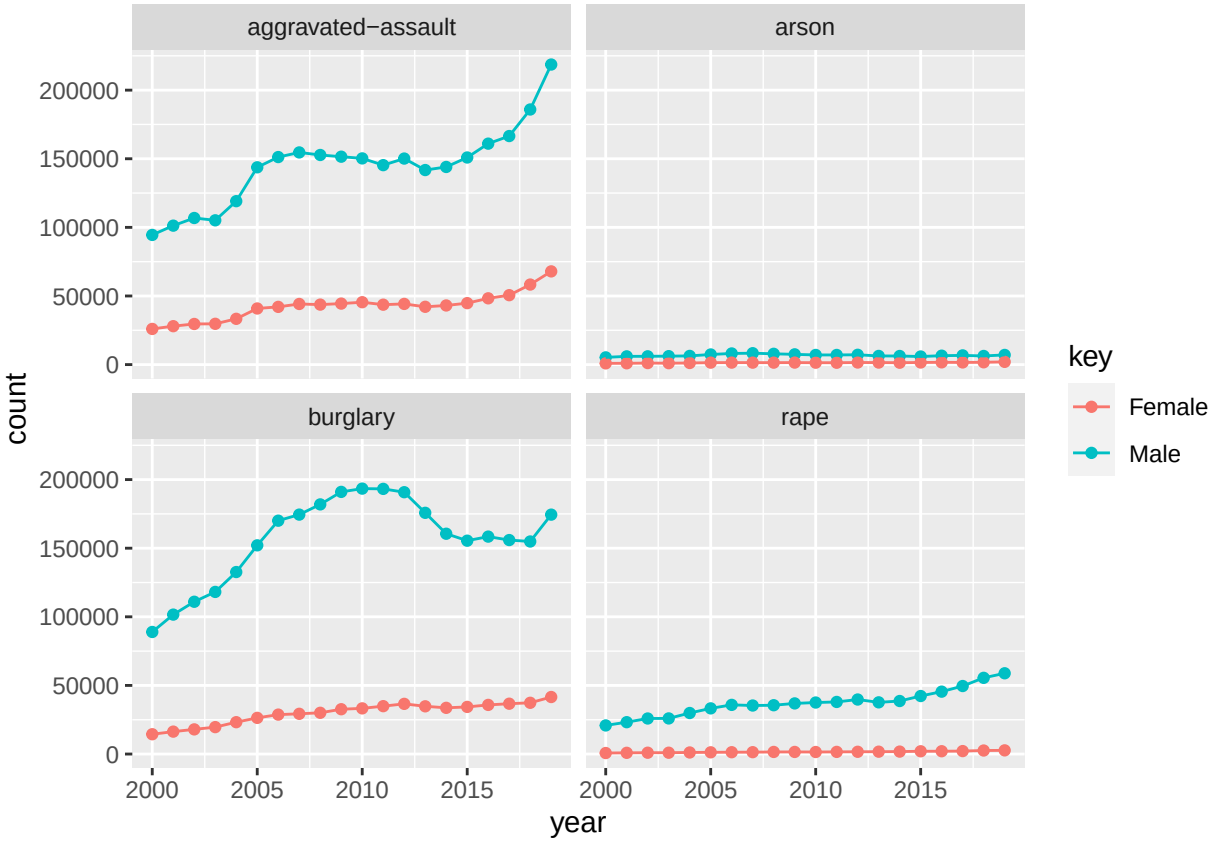
Here are some potential API wrapper packages. Feel free to use one not included in this list for Problem 2.

- spotifyr
- ieugwasr
- VancouvR
- traveltime
- nbastatR
- eia
- tradestatistics
- fbicrime

- wbstats
- rtweet
- rfishbase
- darksky
- And so many more on this page under the heading: Web-based Open Data

```
devtools::install_github("SUN-Wenjun/fbicrime")
library(fbicrime)
set_fbi_crime_api_key('VvDP7By81YvDjukIqYAmXOcq11mgJzF1gCTvEr00')
fbi_offenses <- summarize_offender(offense = c('burglary', 'arson',
                                               'aggravated-assault',
                                               'rape'),
                                   level = 'national',
                                   level_detail = NULL,
                                   variable = 'sex')

fbi_offenses <- fbi_offenses %>%
  unnest(key)

write_csv(fbi_offenses, file = "fbi_offenders.csv")
```

```
fbi_data <- read_csv("fbi_offenders.csv")
fbi_data %>%
  filter(!key == "Unknown", year >= 2000) %>%
  ggplot(aes(x = year, y = count, colour = key)) +
  geom_point() +
  geom_line() +
  facet_wrap(~type)
```



This graph that shows the count of arrests across the nation for aggravated-assault, arson, burglary, and rape by sex from 2000-2019. Some conclusions one can draw from this plot is that men commit more of all these specific crimes than women, or at least they are arrested and convicted more often. Arson is the least common offense of the offenses included in this dataset. Notably, there is a general increase of arrests for aggregated-assault and burglary over that 19-year time span for both sex, though arrests for rape have also increased over time but only with men. Aggregated-assault arrest rates were pretty steady from 2005 to 2015, but started to exponentially increase (especially for men) those last 4 years.

Problem 3: Scraping Reedie Data

Let's see what lovely data we can pull from Reed's own website.

a. Go to https://www.reed.edu/ir/success.html and scrap the two tables. But first check whether or not the website allows scraping.

```
#Store url
url <- "https://www.reed.edu/ir/success.html"

# Ask first
robotstxt::paths_allowed(url)
```

```
## [1] TRUE
```

```
## Scrape html and store table

#Option 1: Grab all the tables and then navigate to the one you wanted.
tables <- url %>%
  read_html() %>%
  html_nodes(css = "table")
```

b. Grab and print out the table that is entitled "GRADUATE SCHOOLS MOST FREQUENTLY ATTENDED BY REED ALUMNI". Why is this data frame not in a tidy format?

```
graduate_schools <- html_table(tables[[2]], fill = TRUE)
graduate_schools
```

```
##                  MBAs                        JDs                    PhDs
## 1       U. of Chicago Lewis & Clark  Law School          U.C., Berkeley
## 2          Harvard U.            U.C., Berkeley         U. of Washington
## 3   Portland State U.              U. of Oregon            U. of Chicago
## 4  U. of Pennsylvania          U. of Washington              Stanford U.
## 5    U. of Washington             U. of Chicago             U. of Oregon
## 6         Columbia U.               New York U.               Harvard U.
## 7         Stanford U.                   Yale U.               Cornell U.
## 8             Yale U.                Harvard U.               Columbia U.
## 9      U.C., Berkeley               Cornell U.                   Yale U.
## 10       U. of Oregon             Georgetown U.        U.C., Los Angeles
## 11      Georgetown U.  U.C. Hastings Law School U. of Wisconsin, Madison
## 12  U.C., Los Angeles         U.C., Los Angeles         Johns Hopkins U.
## 13         Cornell U.           Northwestern U.             Princeton U.
## 14      Pepperdine U.           Northeastern U.                   M.I.T.
## 15        New York U.               Columbia U.          U.C., San Diego
##                             MDs
## 1   Oregon Health  Sciences U.†
## 2              U. of Washington
## 3    Washington U.  (St. Louis)
## 4                   Stanford U.
```

```
## 5           U.C., San Francisco
## 6                    Harvard U.
## 7       Case Western Reserve U.
## 8              Johns Hopkins U.
## 9                    Cornell U.
## 10                   U. Chicago
## 11                      Yale U.
## 12    U. of Southern California
## 13 U. of Minnesota, Minneapolis
## 14              U. of Rochester
## 15                  New York U.
```

The rows do not represent observations.

c. Wrangle the data into a tidy format.

```
graduate_schools_tidy <- pivot_longer(
  graduate_schools, cols = c(MBAs, JDs, PhDs, MDs),
                                    names_to = "grad.program",
                                    values_to = "school"
  ) %>%
  arrange(grad.program)
graduate_schools_tidy
```

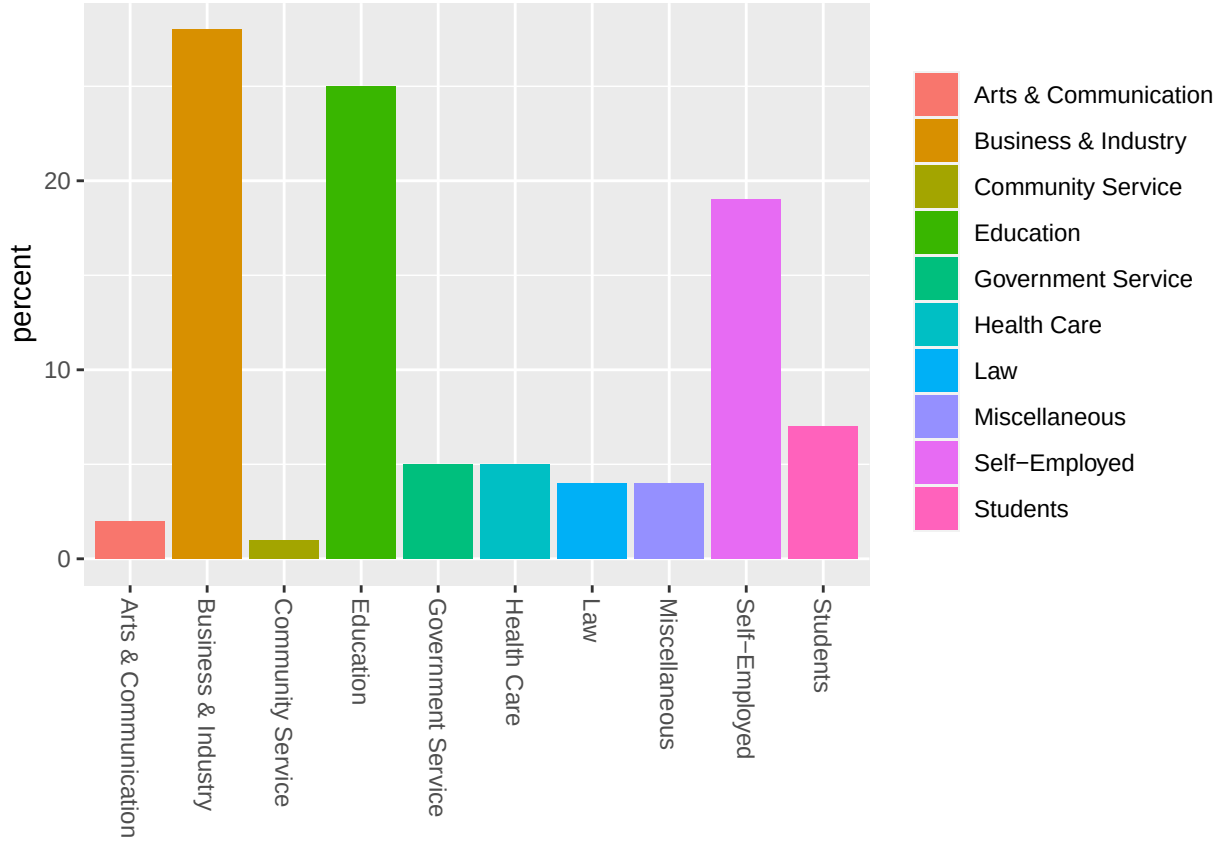
```
## # A tibble: 60 x 2
##    grad.program school
##    <chr>        <chr>
##  1 JDs          Lewis & Clark  Law School
##  2 JDs          U.C., Berkeley
##  3 JDs          U. of Oregon
##  4 JDs          U. of Washington
##  5 JDs          U. of Chicago
##  6 JDs          New York U.
##  7 JDs          Yale U.
##  8 JDs          Harvard U.
##  9 JDs          Cornell U.
## 10 JDs          Georgetown U.
## # ... with 50 more rows
```

d. Now grab the "OCCUPATIONAL DISTRIBUTION OF ALUMNI" table and turn it into an appropriate graph. What conclusions can we draw from the graph?

```
occupation_dist <- html_table(tables[[1]], fill = TRUE)
occupation_dist
```

```
##                      X1  X2
## 1   Business & Industry 28%
## 2             Education 25%
## 3         Self-Employed 19%
## 4              Students  7%
## 5    Government Service  5%
## 6           Health Care  5%
## 7                   Law  4%
## 8         Miscellaneous  4%
## 9  Arts & Communication  2%
## 10    Community Service  1%
```

```
occupation_dist <- occupation_dist %>%
  mutate(parse_number(X2)) %>%
  select('X1', 'parse_number(X2)')%>%
  rename(
    occupation = X1,
    percent = 'parse_number(X2)'
  )

ggplot(occupation_dist, aes(x = occupation, y = percent, fill = occupation)) +
  geom_bar(stat = 'identity') +
  theme(axis.title.x = element_blank(),
        legend.title = element_blank(),
        axis.text.x = element_text(angle = -90, hjust = 0, vjust = .5))
```



Based on the 2014 alumnis, most Reedies graduate and go into the fields of business & industry, education, or they are self-employed.

e. Let's now grab the Reed graduation rates over time. Grab the data from here.

```
#Store url
url2 <- "https://www.reed.edu/ir/gradrateshist.html"

# Ask first
robotstxt::paths_allowed(url2)
```

```
## [1] TRUE
```

```
## Scrape html and store table
```

```r
#Option 1: Grab all the tables and then navigate to the one you wanted.
tables2 <- url2 %>%
  read_html() %>%
  html_nodes(css = "table")

grad_time <- html_table(tables2[[1]], fill = TRUE)
grad_time
```

```
##    First-year students who entered fall of... Number in Cohort Graduated in:
## 1  First-year students who entered fall of... Number in Cohort       4 Years
## 2                                       2016              353           66%*
## 3                                       2015              418            61%
## 4                                       2014              346            62%
## 5                                       2013              354            64%
## 6                                       2012              320            68%
## 7                                       2011              372            65%
## 8                                       2010              373            66%
## 9                                       2009              367            69%
## 10                                      2008              330            66%
## 11                                      2007              337            70%
## 12                                      2006              371            60%
## 13                                      2005              348            59%
## 14                                      2004              333            59%
## 15                                      2003              298            57%
## 16                                      2002              307            60%
## 17                                      2001              349            58%
## 18                                      2000              358            57%
## 19                                      1999              331            52%
## 20                                      1998              338            49%
## 21                                      1997              315            46%
## 22                                      1996              357            45%
## 23                                      1995              352            47%
## 24                                      1994              301            46%
## 25                                      1993              327            45%
## 26                                      1992              310            48%
## 27                                      1991              293            47%
## 28                                      1990              282            32%
## 29                                      1989              305            42%
## 30                                      1988              311            42%
## 31                                      1987              313            40%
## 32                                      1986              322            33%
## 33                                      1985              300            36%
## 34                                      1984              244            33%
## 35                                      1983              297            31%
## 36                                      1982              242            28%
##    Graduated in: Graduated in:
## 1        5 Years       6 Years
## 2              -             -
## 3          70%*             -
## 4           73%          77%*
## 5           72%           76%
## 6           78%           81%
## 7           77%           80%
## 8           76%           78%
```

```
## 9            79%            82%
## 10           77%            79%
## 11           80%            82%
## 12           73%            74%
## 13           76%            80%
## 14           76%            79%
## 15           76%            78%
## 16           76%            77%
## 17           71%            75%
## 18           72%            75%
## 19           68%            73%
## 20           66%            70%
## 21           67%            72%
## 22           63%            68%
## 23           66%            70%
## 24           62%            66%
## 25           63%            65%
## 26           64%            68%
## 27           65%            66%
## 28           50%            56%
## 29           61%            66%
## 30           61%            63%
## 31           67%            69%
## 32           58%            65%
## 33           56%            63%
## 34           55%            63%
## 35           52%            58%
## 36           47%            54%
```

Do the following to clean up the data:

- Rename the column names.

```
# Hint
colnames(grad_time) <- c("entering.class.year", "cohort.size",
                         "4", "5", "6")
```

- Remove any extraneous rows.

```
# Hint
grad_time <- grad_time %>%
  filter(row_number() >= 2)
```

- Reshape the data so that there are columns for
  - Entering class year
  - Cohort size
  - Years to graduation
  - Graduation rate

```
grad_time <- pivot_longer(grad_time, cols = c('4','5','6'),
                          names_to = "years.to.graduation",
                          values_to = "graduation.rate")
```

- Make sure each column has the correct class.

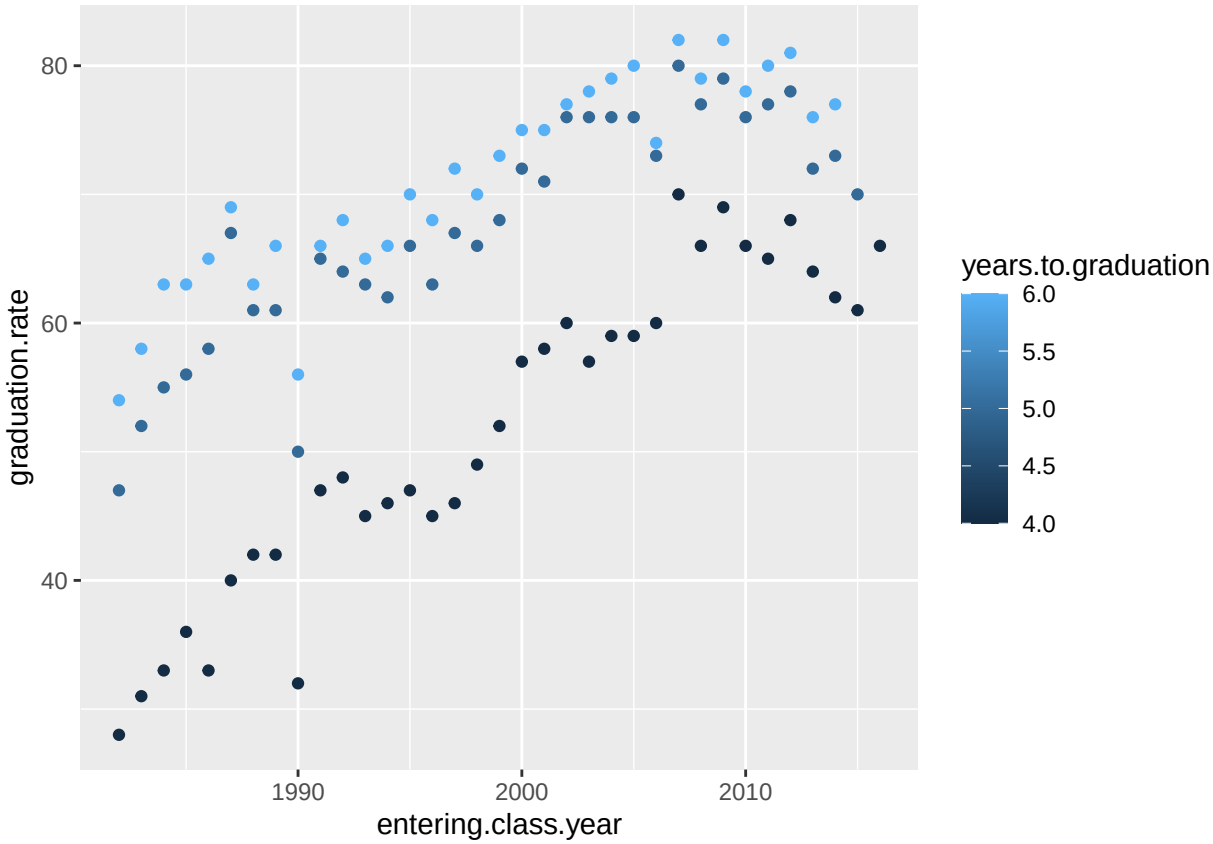
```
grad_time$entering.class.year <- as.numeric(grad_time$entering.class.year)
grad_time$cohort.size <- as.numeric(grad_time$cohort.size)
grad_time$years.to.graduation <- as.numeric(grad_time$years.to.graduation)
```

```
grad_time[grad_time == "-" ] <- NA
grad_time <- grad_time %>%
  mutate(parse_number(graduation.rate)) %>%
  select(entering.class.year, cohort.size, years.to.graduation, 'parse_number(graduation.rate)')%>%
  rename(graduation.rate = 'parse_number(graduation.rate)')
grad_time
```

```
## # A tibble: 105 x 4
##    entering.class.year cohort.size years.to.graduation graduation.rate
##                  <dbl>       <dbl>               <dbl>           <dbl>
##  1                2016         353                   4              66
##  2                2016         353                   5              NA
##  3                2016         353                   6              NA
##  4                2015         418                   4              61
##  5                2015         418                   5              70
##  6                2015         418                   6              NA
##  7                2014         346                   4              62
##  8                2014         346                   5              73
##  9                2014         346                   6              77
## 10                2013         354                   4              64
## # ... with 95 more rows
```

f. Create a graph comparing the graduation rates over time and draw some conclusions.

```
ggplot(grad_time, aes(x = entering.class.year, y = graduation.rate, colour = years.to.graduation)) +
  geom_point()
```



There has been a drastic increase in graduation rate overall during the past 25 years or so, especially in the

percent of each class graduating in 4 years. It was not til the 2000's that Reed hit more than 80% of a class within 6 years since they enrolled. Looks like Reed stepped up graduation rates drastically, but it is looks like there might be a dip in recent graduates, but it is not clear enough to call that.

Problem 4: Scraping the Wild We(b)st

Find a web page that contains at least one table and scrap it using `rvest`. Once you've pulled the data into R,

- write it to a csv so that you aren't pulling the data each time you knit the document.
- load the dataset.
- use the data to construct a graph or compute some summary statistics.

- State what conclusions can be drawn from the data.

Notes:

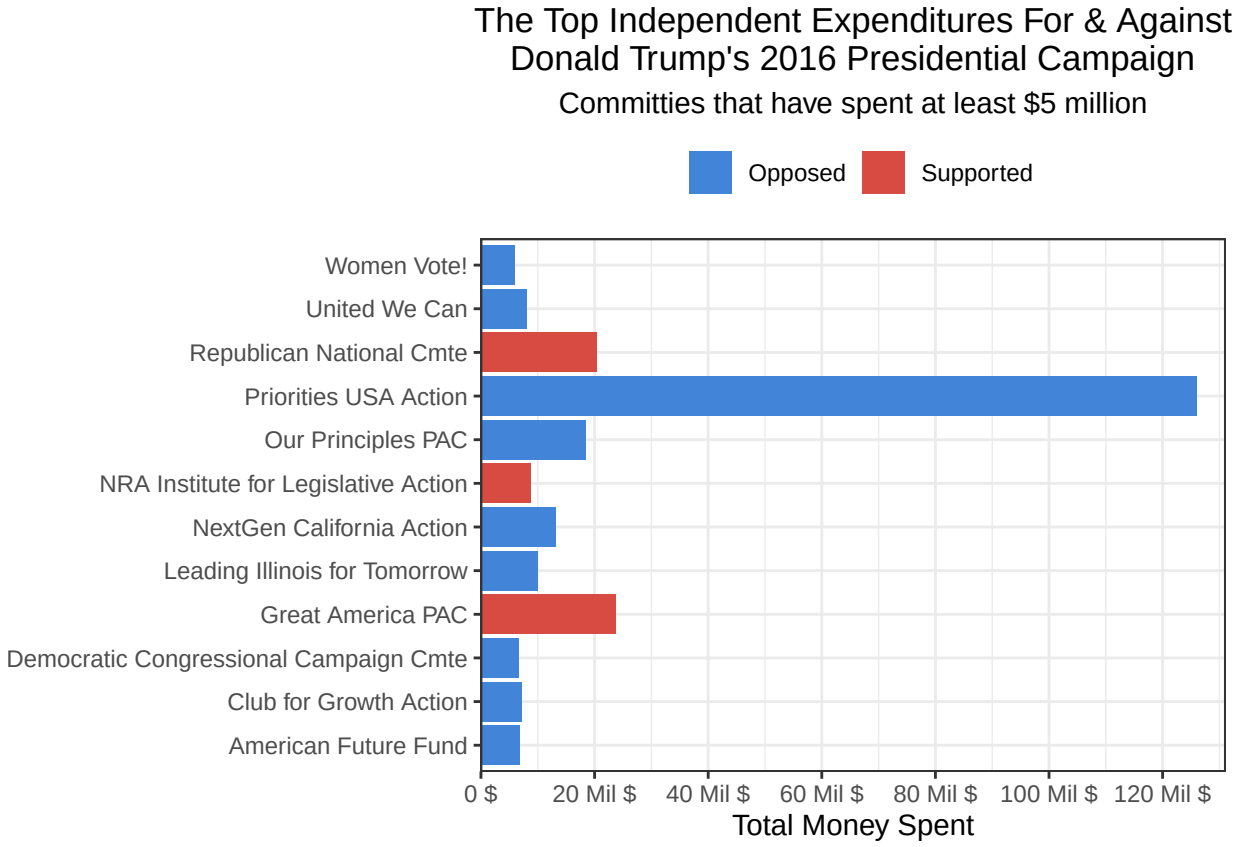
1. Don't try to scrap data that is on multiple pages.

2. On some websites, how the data are stored is very messy. If you are struggling to determine the correct CSS, try a new page.
3. SelectorGadget (a Chrome Add-on) can be a helpful tool for determining the CSS selector.

```
#Store url
url3 <- "https://www.opensecrets.org/pres16/outside-spending?id=N00023864"

# Ask first
robotstxt::paths_allowed(url3)

## Scrape html and store table

#Option 1: Grab all the tables and then navigate to the one you wanted.
tables3 <- url3 %>%
  read_html() %>%
  html_nodes(css = "table")

trump_ind_expend <- html_table(tables3[[1]], fill = TRUE)
trump_ind_expend

write_csv(trump_ind_expend, file = "trump_ind_expend.csv")
```

```
trump_ind_expend <- read_csv("trump_ind_expend.csv",
    col_types = cols(`Entire Cycle Total` = col_number(),
        Supported = col_number(), Opposed = col_number()))
trump_ind_expend <- trump_ind_expend %>%
  select(Committee, `Entire Cycle Total`, Supported, Opposed) %>%
  pivot_longer(cols = c(Supported, Opposed),
    names_to = "Position",
    values_to = "Money.Spent")

filter(trump_ind_expend, Money.Spent >= 5000000) %>%
  ggplot(aes(x = Committee, y = Money.Spent, fill = Position)) +
  geom_bar(stat = 'identity')+
  theme_bw() +
  scale_y_continuous(
    breaks = c(0, 20000000, 40000000, 60000000, 80000000, 100000000,
```

```
              120000000, 140000000),
    labels = c("0 $", "20 Mil $", "40 Mil $", "60 Mil $", "80 Mil $",
               "100 Mil $", "120 Mil $", "140 Mil $"),
    expand = expansion(add = c(0, 5000000))
  ) +
  labs(
    title = "The Top Independent Expenditures For & Against\nDonald Trump's 2016 Presidential Campaign",
    subtitle = "Committies that have spent at least $5 million",
    y = "Total Money Spent"
  ) +
  theme(
    legend.title = element_blank(),
    plot.title = element_text(hjust = .5),
    plot.subtitle = element_text(hjust = .5),
    axis.title.y = element_blank(),
    legend.position = "top"
  ) +
  scale_fill_manual(
    values = c("#4285D8", "#D84B42"),
    breaks = c("Opposed", "Supported")
  ) +
  coord_flip()
```



The top spending PACs or SuperPACs on Donald Trump's 2016 presidential campaign, or at least the ones that ave spent more than 5 million dollars, most tend to be money used to oppose Trump (only 3 of the 12 are in support of trump's campaign). The top spending committee, Priorities USA PAC, far outspent the other top contenders by around 100 million dollars to oppose Trump's campaign. That leads me to believe

that a lot more money was spent against Trump by the top spending committees than for Trump.