Lab 4

Math 241, Week 5

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

Due: Thursday, March 4th at 8:30am

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.

```
# Don't forget to install it first!
library(rnoaa)
```

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

```
# Then change eval to TRUE!
options(noaakey = "aGYuXQPNVgiIDSWMDagjrxqmmPieXldN")
```

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?

```
# Change to eval = TRUE when have your API key stored!
stations <- ncdc_stations(datasetid = "GHCND",
                          locationid = "FIPS:41051")

mult_stations <- stations$data
```

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 = "2021-02-25",
                      stationid = "GHCND:US1ORMT0006",
                      limit = 1000)

snow_se_pdx <- ncdc(datasetid = "GHCND", datatypeid = "SNOW",
                      startdate = "2021-01-01",
                      enddate = "2021-02-25",
                      stationid = "GHCND:US1ORMT0006",
                      limit = 1000)
```

d. What is the class of `precip_se_dpx` 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.

Both `precip_se_dpx` and `snow_se_pdx` are ncdc_data.

```
# Leave eval = FALSE
se_pdx_data <- bind_rows(precip_se_pdx$data, snow_se_pdx$data)

stringr::str_replace(se_pdx_data$date, "T"," ")

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

```
# Read the file in here!
com_se_pdx_data <- read_csv("se_pdx_data.csv")
```

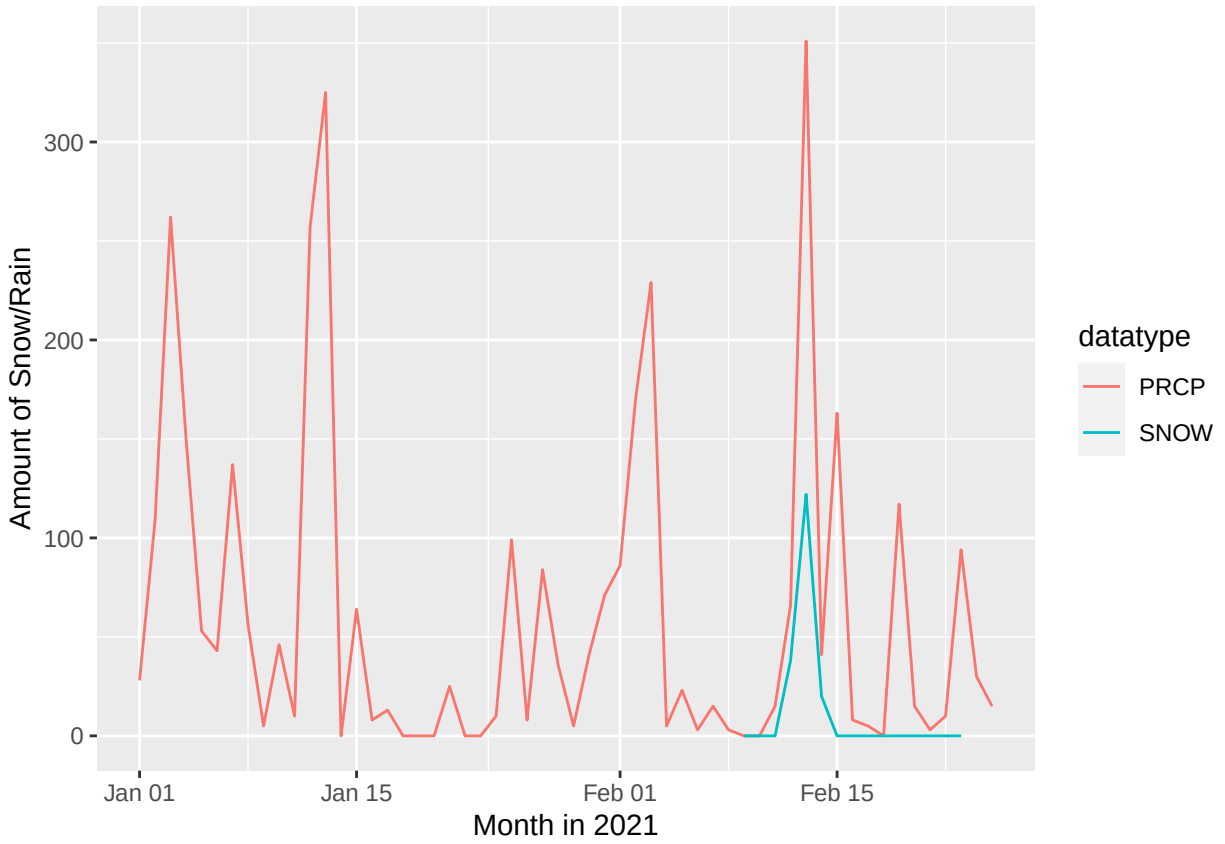
e. Use `ymd_hms()` in the package `lubridate` to wrangle the date column into the correct format.

```
clean_sepdx_data <- ymd_hms(com_se_pdx_data$date)

#omitted from this problems set

clean_sepdx_data
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

f. Plot the precipitation and snowfall data for this site in Portland over time. Comment on any trends. There is a lot of rain towards the end of the month. The amount of snow and rain overlap with each other.

```
com_se_pdx_data %>%
  ggplot(aes(x = date, y = value, color = datatype)) +
  geom_line() +
  labs(x = "Month in 2021", y = "Amount of Snow/Rain")
```



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.

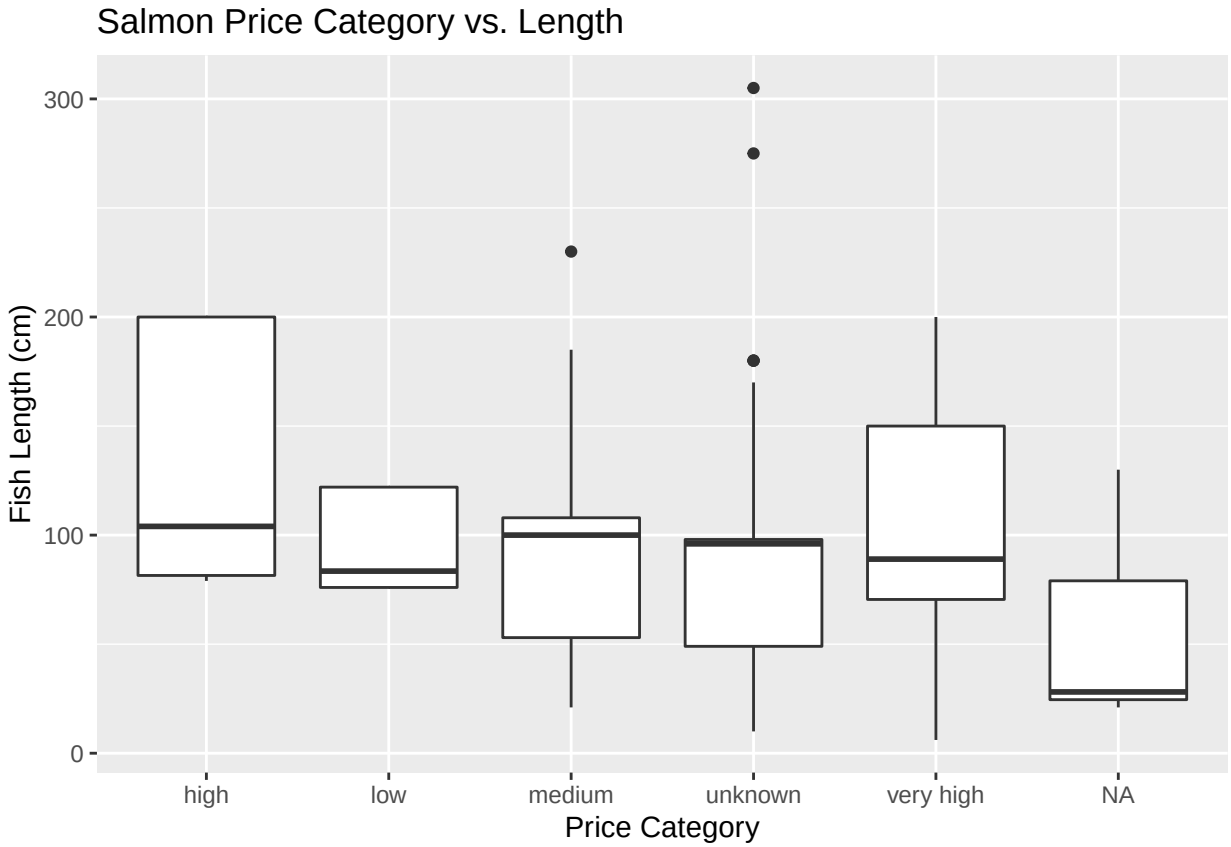
- spotifyr
- ieugwasr
- VancouvR
- traveltime
- nbastatR
- eia
- tradestatistics
- fbicrime
- wbstats
- rtweet

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

```
remotes::install_github("ropensci/rfishbase")
library("rfishbase")

salmon <- common_to_sci("salmon")
salmon
```

```
## # A tibble: 221 x 4
##    Species                 ComName           Language SpecCode
##    <chr>                   <chr>             <chr>       <dbl>
##  1 Salmo obtusirostris     Adriatic salmon   English      6210
##  2 Salmo trutta            Aral salmon       English       238
##  3 Salmo salar             Atlantic salmon   English       236
##  4 Arripis truttacea       Australian salmon English     14606
##  5 Arripis trutta          Australian salmon English      5048
##  6 Salmo salar             Bay salmon        English       236
##  7 Leptobrama muelleri     Beach salmon      English      4338
##  8 Leptobrama muelleri     Beachsalmon       English      4338
##  9 Gonorynchus gonorynchus Beaked salmon     English      4528
## 10 Gonorynchus greyi       Beaked salmon     English     12749
## # ... with 211 more rows
```

```
sal <- species(salmon$Species, fields = c("Species",
                                          "PriceCateg",
                                          "Weight",
                                          "Length"))

sal_fishbase <- write_csv(sal, "sal_fishbase.csv")
sal_fishbase <- read_csv("sal_fishbase.csv")

sal_fishbase <- sal_fishbase %>%
  mutate(price = fct_relevel(PriceCateg, levels = c("low", "medium","high")))

ggplot(sal_fishbase, aes(x = PriceCateg, y = Length)) +
  geom_boxplot() +
  labs(title = "Salmon Price Category vs. Length") +
  labs(
    x = "Price Category",
    y = "Fish Length (cm)"
  )
```



The median for each price category varies with length of salmon.

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.

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

robotstxt::paths_allowed(url)
```

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

```
tables <- url %>%
  read_html() %>%
  html_nodes(css = "table")

R1_table <- html_table(tables[[1]], fill = TRUE)
R1_table
```

```
##                       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%
```

```
R2_table <- html_table(tables[[2]], fill = TRUE)
R2_table
```

```
##                 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.
```

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?

```
R2_table <- html_table(tables[[2]], fill = TRUE)
R2_table
```

```
##                 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.
```

Each row should correspond to one observation but there are many observations for each row.

c. Wrangle the data into a tidy format.

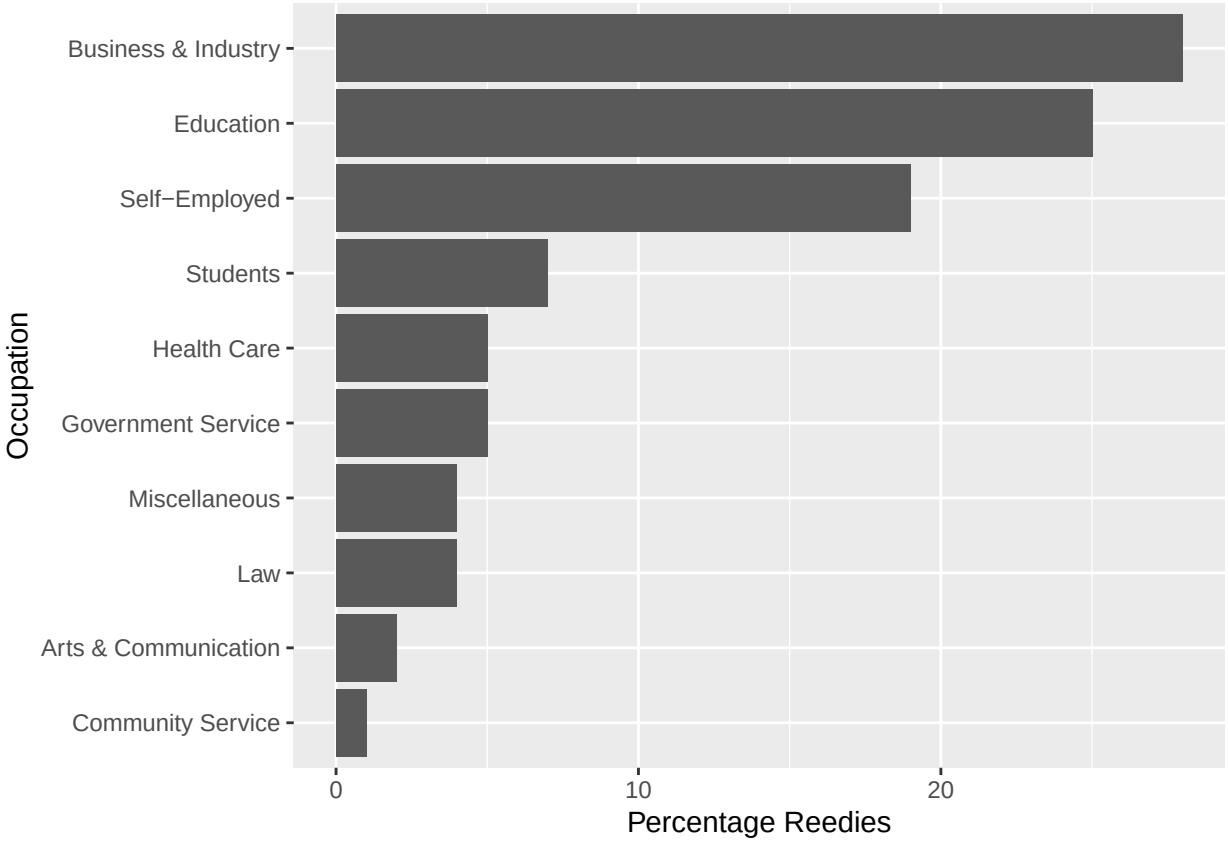
```
R2_table_tidy <- R2_table %>%
  pivot_longer(c(MBAs, JDs, PhDs, MDs), names_to = "TypeDegree", values_to = "Name of School")

R2_table_tidy
```

```
## # A tibble: 60 x 2
##    TypeDegree `Name of School`
##    <chr>      <chr>
##  1 MBAs       U. of Chicago
##  2 JDs        Lewis & Clark  Law School
##  3 PhDs       U.C., Berkeley
##  4 MDs        Oregon Health  Sciences U.†
##  5 MBAs       Harvard U.
##  6 JDs        U.C., Berkeley
##  7 PhDs       U. of Washington
##  8 MDs        U. of Washington
##  9 MBAs       Portland State U.
## 10 JDs        U. of Oregon
## # ... 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?

```
# Hint: Use `parse_number()` within `mutate()` to fix one of the columns
R1_table <- R1_table %>%
  mutate(pct = parse_number(X2)) %>%
  mutate(occupation = reorder(X1, pct))

R1_table %>%
  ggplot(mapping = aes(x = occupation, y = pct)) +
```

```
  geom_col() +
  labs(x = "Occupation", y = "Percentage Reedies") +
  coord_flip()
```



Many Reedies pursue industry and education after Reed. Seems that not many Reedies pursue arts & communication or community service after graduation.

e. Let's now grab the Reed graduation rates over time. Grab the data from here. Do the following to clean up the data:

```
# Hint
grad_rate <- "https://www.reed.edu/ir/gradrateshist.html"

rate_table <- grad_rate %>%
  read_html() %>%
  html_nodes(css = "table")

grad_rate_table <- html_table(rate_table[[1]], fill = TRUE)
```

- Rename the column names.

```
colnames(grad_rate_table) <- c("Year", "Cohort_size", "gradfour", "gradfive", "gradsix")
```

- Remove any extraneous rows.

```
# Hint
grad_rate_table1 <- grad_rate_table %>%
  slice(-1)
```

- Reshape the data so that there are columns for

  - Entering class year
  - Cohort size
  - Years to graduation
  - Graduation rate
- Make sure each column has the correct class.

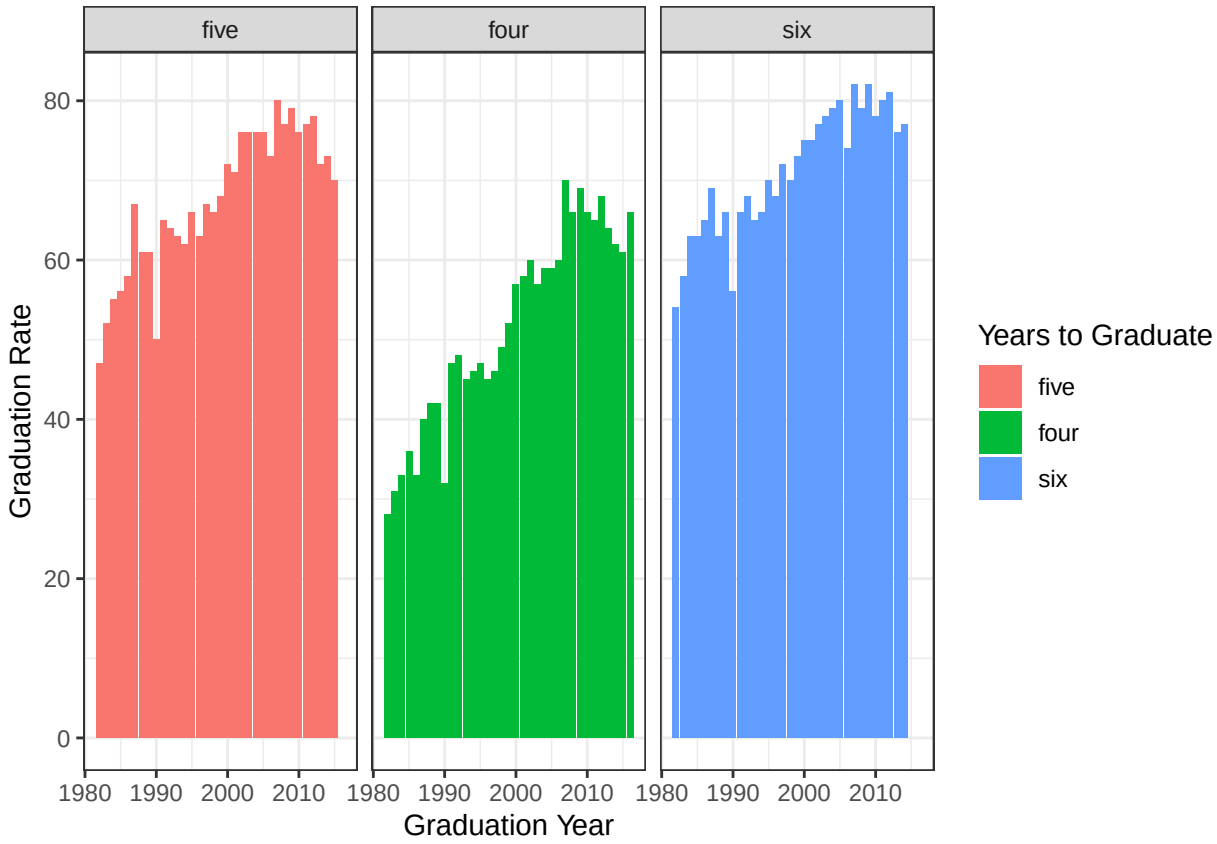
```
gradratetable <- grad_rate_table %>%
  mutate(Grad_year = as.numeric(Year),
         cohort = parse_number(Cohort_size),
         four = parse_number(gradfour),
         five = parse_number(gradfive),
         six = parse_number(gradsix)
         ) %>%
  dplyr::select(Grad_year, cohort, four, five, six)
```

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

```
grad_rate_table_final <- pivot_longer(gradratetable, cols = c(four, five, six),
                                names_to = "Years to Graduate",
                                values_to = "Graduation Rate") %>%
  select(Grad_year, cohort, `Years to Graduate`, `Graduation Rate`)

head(grad_rate_table)
```

```
##                                         Year      Cohort_size gradfour gradfive
## 1 First-year students who entered fall of... Number in Cohort  4 Years  5 Years
## 2                                       2016              353     66%*        -
## 3                                       2015              418      61%     70%*
## 4                                       2014              346      62%      73%
## 5                                       2013              354      64%      72%
## 6                                       2012              320      68%      78%
##   gradsix
## 1 6 Years
## 2       -
## 3       -
## 4    77%*
## 5     76%
## 6     81%
```

```
ggplot(grad_rate_table_final, aes(fill = `Years to Graduate`, y = `Graduation Rate`, x = Grad_year)) +
  geom_bar(stat = "identity") +
  facet_grid(. ~ `Years to Graduate`) +
  theme_bw() +
  labs(x = "Graduation Year")
```



The graduation rate has improved over time for people graduating in four, five, and six years.

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.

Conclusions: The United States is producing the most amount of refined oil followed by Russia and the United Arab Emirates. Additionally, the viewer can see that there are many OPEC countries.

```
oil_scraping <- read_html("https://en.wikipedia.org/wiki/List_of_countries_by_oil_production") %>%
 html_nodes("table")

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

oil_csv <- write.csv(oil_scraping, "oil_scraping.csv")
```

```r
oil_csv1 <- read_csv("oil_scraping.csv")

oil_csv2 <- oil_csv1 %>%
  rename(
    oilprod = `Oil production2019 (bbl/day)[1]`
  ) %>%
  filter(oilprod > 1000000,
         Country != "World")

oil_csv2 %>%
  ggplot(mapping = aes(x = Country, y = oilprod)) +
           geom_bar(stat = "identity") +
           theme_bw() +
           labs(title = "Oil export by country", x = "Country", y = "Barrels per Day(bbl/day)") +
  coord_flip()
```