Lab 5

Math 241, Week 6

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

Due: Friday, March 1st 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.

Potential API Wrapper Packages

Problem 1: Predicting the ~~Un~~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:

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
```

We have 25 stations in Multnomah County.

c. January was not so rainy this year, was it? Let's grab the precipitation data for site `GHCND:US1ORMT0006` for this past January.

```
# First fill-in and run to following to determine the
# datatypeid
```

```r
ncdc_datatypes(datasetid = "GHCND",
               stationid = "GHCND:US1ORMT0006")
```

```
## $meta
##   offset count limit
## 1      1     5    25
##
## $data
##      mindate    maxdate                                   name datacoverage
## 1 1750-02-01 2024-02-27                          Precipitation            1
## 2 1840-05-01 2024-02-27                               Snowfall            1
## 3 1857-01-18 2024-02-27                             Snow depth            1
## 4 1952-07-01 2024-02-27 Water equivalent of snow on the ground            1
## 5 1998-06-01 2024-02-27           Water equivalent of snowfall            1
##     id
## 1 PRCP
## 2 SNOW
## 3 SNWD
## 4 WESD
## 5 WESF
##
## attr(,"class")
## [1] "ncdc_datatypes"
```

```r
# Now grab the data using ncdc()
precip_se_pdx <- ncdc(datasetid = "GHCND", stationid = "GHCND:US1ORMT0006", datatypeid = "PRCP", startda
data.frame(precip_se_pdx[2])
```

```
##              data.date data.datatype      data.station data.value data.fl_m
## 1  2024-01-01T00:00:00          PRCP GHCND:US1ORMT0006          0         T
## 2  2024-01-02T00:00:00          PRCP GHCND:US1ORMT0006          0
## 3  2024-01-03T00:00:00          PRCP GHCND:US1ORMT0006         58
## 4  2024-01-04T00:00:00          PRCP GHCND:US1ORMT0006        107
## 5  2024-01-05T00:00:00          PRCP GHCND:US1ORMT0006         28
## 6  2024-01-06T00:00:00          PRCP GHCND:US1ORMT0006        135
## 7  2024-01-07T00:00:00          PRCP GHCND:US1ORMT0006         97
## 8  2024-01-08T00:00:00          PRCP GHCND:US1ORMT0006         56
## 9  2024-01-09T00:00:00          PRCP GHCND:US1ORMT0006        221
## 10 2024-01-10T00:00:00          PRCP GHCND:US1ORMT0006        157
## 11 2024-01-11T00:00:00          PRCP GHCND:US1ORMT0006         25
## 12 2024-01-12T00:00:00          PRCP GHCND:US1ORMT0006         66
## 13 2024-01-13T00:00:00          PRCP GHCND:US1ORMT0006          5
## 14 2024-01-14T00:00:00          PRCP GHCND:US1ORMT0006         94
## 15 2024-01-15T00:00:00          PRCP GHCND:US1ORMT0006          0
## 16 2024-01-16T00:00:00          PRCP GHCND:US1ORMT0006          0
## 17 2024-01-17T00:00:00          PRCP GHCND:US1ORMT0006        107
## 18 2024-01-18T00:00:00          PRCP GHCND:US1ORMT0006        178
## 19 2024-01-19T00:00:00          PRCP GHCND:US1ORMT0006        183
## 20 2024-01-20T00:00:00          PRCP GHCND:US1ORMT0006          0
## 21 2024-01-21T00:00:00          PRCP GHCND:US1ORMT0006         89
## 22 2024-01-22T00:00:00          PRCP GHCND:US1ORMT0006        178
## 23 2024-01-23T00:00:00          PRCP GHCND:US1ORMT0006        175
## 24 2024-01-24T00:00:00          PRCP GHCND:US1ORMT0006         91
## 25 2024-01-25T00:00:00          PRCP GHCND:US1ORMT0006        130
##    data.fl_q data.fl_so data.fl_t
## 1                     N      0747
## 2                     N      0700
## 3                     N      0842
## 4                     N      0847
## 5                     N      0835
## 6                     N      0836
## 7                     N      0738
## 8                     N      0840
## 9                     N      0840
## 10                    N      0845
## 11                    N      0820
## 12                    N      0841
## 13                    N      0830
## 14                    N      0847
## 15                    N      0700
## 16                    N      0700
## 17                    N      0818
## 18                    N      0843
## 19                    N      0828
## 20                    N      0835
## 21                    N      0841
## 22                    N      0741
## 23                    N      0830
## 24                    N      0830
## 25                    N      0735
```

d. What is the class of `precip_se_dpx`? Grab the data frame nested in `precip_se_dpx` and call it `precip_se_dpx_data`.

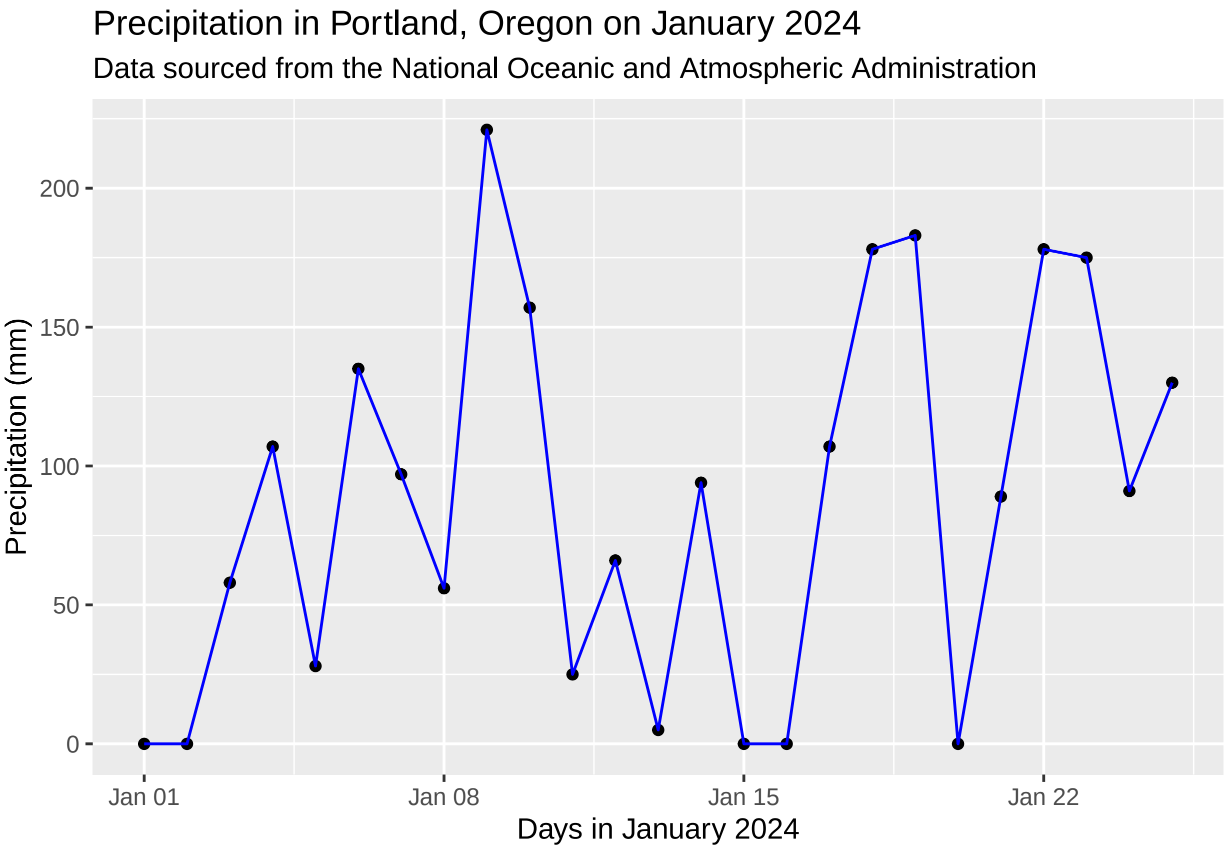
The class of it is ncdc_data (which is a list of multiple (2) things).

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

```r
precip_se_dpx_data$data.date <- ymd_hms(precip_se_dpx_data$data.date)
```

f. Plot the precipitation data for this site in Portland over time. Rumor has it that we had only one day where it didn't rain. Is that true?

```r
ggplot(data = precip_se_dpx_data, mapping = aes(x = data.date, y = data.value)) +
  geom_point() +
  geom_line(mode = "lm", color = "blue") +
  labs(x = "Days in January 2024",
       y = "Precipitation (mm)",
       title = "Precipitation in Portland, Oregon on January 2024",
       subtitle = "Data sourced from the National Oceanic and Atmospheric Administration")
```

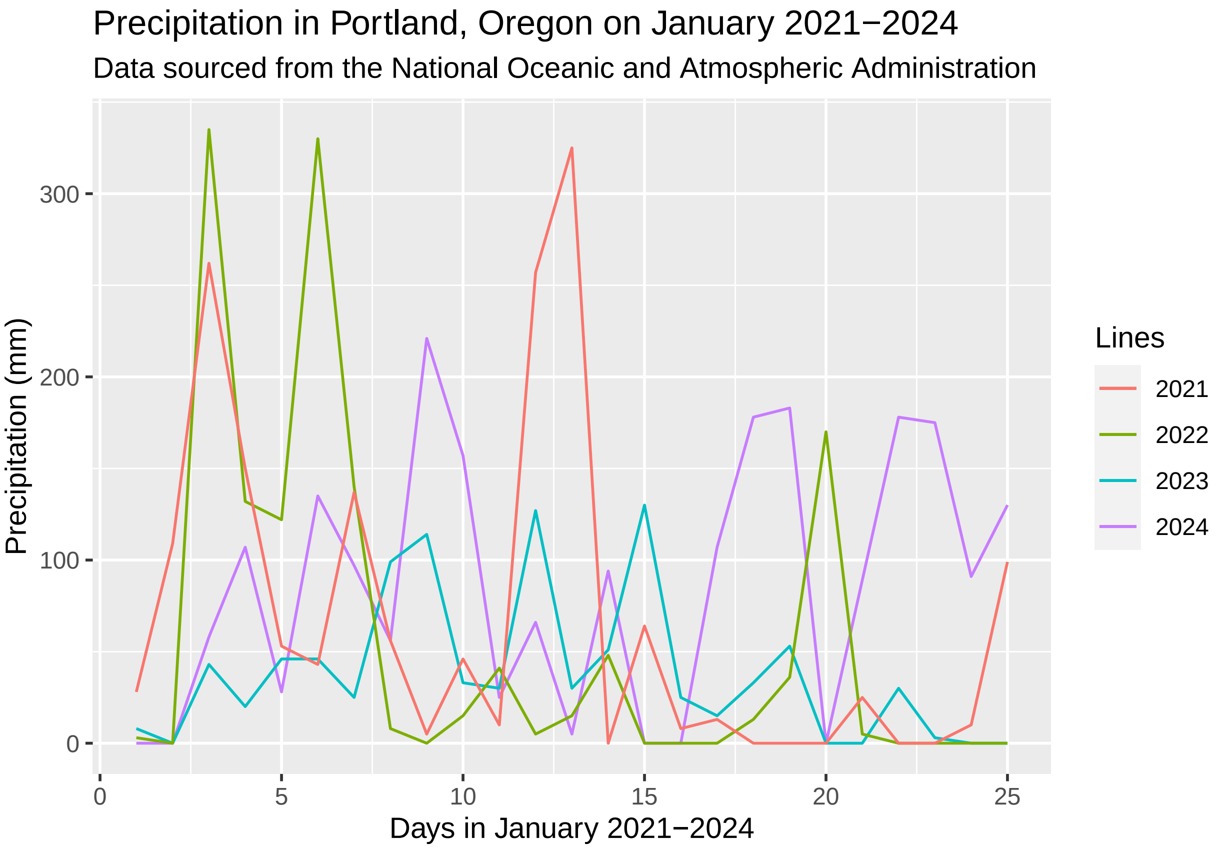


g. (Bonus) Adapt the code to create a visualization that compares the precipitation data for January over the the last four years. Do you notice any trend over time?

```
precip_se_pdx2023 <- data.frame(ncdc(datasetid = "GHCND", stationid = "GHCND:US1ORMT0006", datatypeid =
precip_se_pdx2022 <- data.frame(ncdc(datasetid = "GHCND", stationid = "GHCND:US1ORMT0006", datatypeid =
precip_se_pdx2021 <- data.frame(ncdc(datasetid = "GHCND", stationid = "GHCND:US1ORMT0006", datatypeid =
```

```
precip_se_pdx2023$data.date <- ymd_hms(precip_se_pdx2023$data.date)
precip_se_pdx2022$data.date <- ymd_hms(precip_se_pdx2022$data.date)
precip_se_pdx2021$data.date <- ymd_hms(precip_se_pdx2021$data.date)

precip_se_pdx2024 <- precip_se_dpx_data

precip_se_pdx2024$data.date <- day(precip_se_pdx2024$data.date)
precip_se_pdx2023$data.date <- day(precip_se_pdx2023$data.date)
precip_se_pdx2022$data.date <- day(precip_se_pdx2022$data.date)
precip_se_pdx2021$data.date <- day(precip_se_pdx2021$data.date)

precip_se_pdx2024$year <- "2024"
precip_se_pdx2023$year <- "2023"
precip_se_pdx2022$year <- "2022"
precip_se_pdx2021$year <- "2021"
```

```
ggplot(data = precip_se_pdx2024, mapping = aes(x = data.date, y = data.value, color = year)) +
  geom_line() +
  geom_line(data = precip_se_pdx2023) +
  geom_line(data = precip_se_pdx2022) +
  geom_line(data = precip_se_pdx2021) +
  labs(x = "Days in January 2021-2024",
       y = "Precipitation (mm)",
       title = "Precipitation in Portland, Oregon on January 2021-2024",
       subtitle = "Data sourced from the National Oceanic and Atmospheric Administration",
       color = "Lines")
```



We can see that outliers have generally decreased, but also a high level of variance from year to year. In terms of average precipitation, it's a little hard to discern a trend from this data.

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 or in Problem 1.

Once you have grabbed the data, do any necessary wrangling to graph it and/or produce some 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.

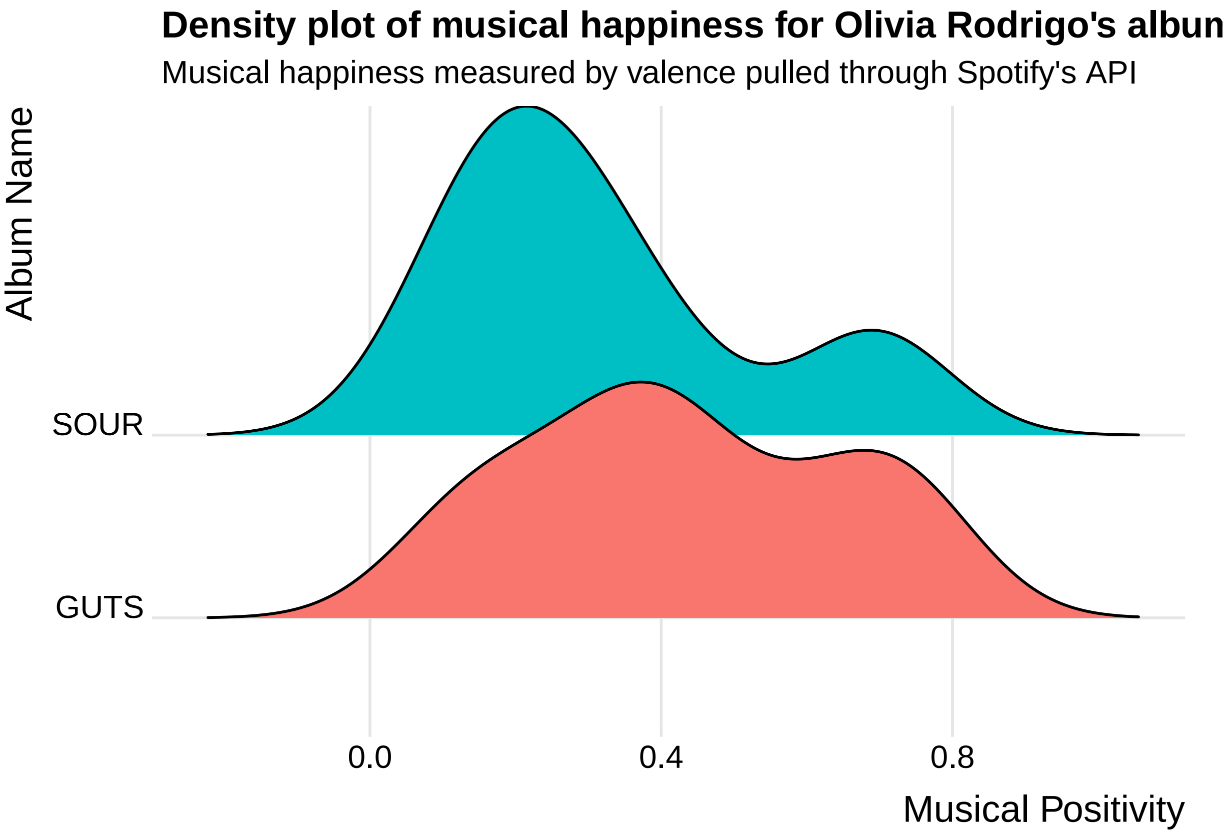
- `gtrendsR`: "An interface for retrieving and displaying the information returned online by Google Trends is provided. Trends (number of hits) over the time as well as geographic representation of the results can be displayed."
- `rfishbase`: For the fish lovers
- `darksky`: For global historical and current weather conditions

This is hidden for my own privacy reasons, but above is code using the spotifyr API wrapped and setting up keys and so.

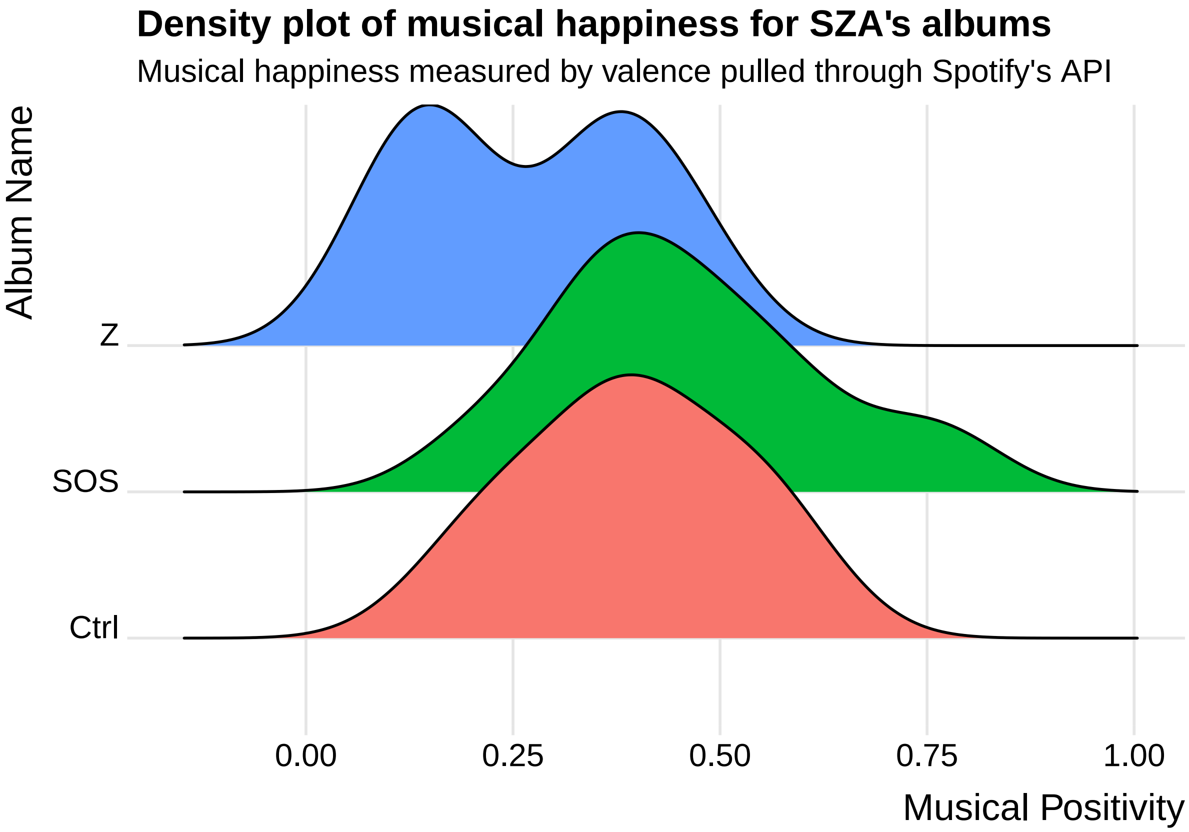
```r
sza_valence <- get_artist_audio_features('sza') %>%
    arrange(-valence) %>%
    select(track_name, valence, album_name) %>%
    filter(album_name != "Dear Evan Hansen (Original Motion Picture Soundtrack)" &
           album_name != "Black Panther The Album Music From And Inspired By" &
           album_name != "Ctrl (Deluxe)")
```

```r
olivia_valence <- get_artist_audio_features('olivia rodrigo') %>%
    arrange(-valence) %>%
    select(track_name, valence, album_name) %>%
    filter(album_name != "The Hunger Games: The Ballad of Songbirds & Snakes (Music From & Inspired By)
```

```r
ggplot(olivia_valence, aes(x= valence, y = album_name, fill = album_name)) +
  geom_joy() +
  theme_joy() +
  labs(title = "Density plot of musical happiness for Olivia Rodrigo's albums",
       subtitle = "Musical happiness measured by valence pulled through Spotify's API",
       x = "Musical Positivity",
       y = "Album Name") +
  theme(legend.position = "none")
```



```r
ggplot(sza_valence, aes(x= valence, y = album_name, fill = album_name, show.legend = F)) +
  geom_joy() +
  theme_joy() +
  labs(title = "Density plot of musical happiness for SZA's albums",
       subtitle = "Musical happiness measured by valence pulled through Spotify's API",
       x = "Musical Positivity",
       y = "Album Name") +
  theme(legend.position = "none")
```



I've created two graphs of two hugely popular artists and measured the positivity in their music (instrumentally). It seems that both artists have variance between their albums, which implies personal growth and development in their styles. Olivia's music also generally seems happier than SZA's music. It is interesting how both Olivia's albums are bimodal and SZA's Z album is also that, while the other two SZA albums are nearly a normal distribution in terms of happiness across songs in the album.

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 scrape the two tables.

```r
html_table(read_html("https://www.reed.edu/ir/success.html"))
```

```
## [[1]]
## # A tibble: 10 x 2
##    X1                     X2
##    <chr>                  <chr>
##  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%
##
## [[2]]
## # A tibble: 11 x 4
##    MBAs                JDs                       PhDs                     MDs
##    <chr>               <chr>                     <chr>                    <chr>
##  1 U. of Chicago       Lewis & Clark  Law School U.C., Berkeley           Oregon~
##  2 Portland State U.   U.C., Berkeley            U. of Washington         U. of ~
##  3 Harvard U.          U. of Oregon              U. of Chicago            Washin~
##  4 U. of Washington    U. of Washington          Stanford U.              UC., S~
##  5 Columbia U.         New York U.               U. of Oregon             Stanfo~
##  6 U of Pennsylvania.  U. of Chicago             Harvard U.               Harvar~
##  7 Stanford U.         Yale U.                   Cornell U.               Case W~
##  8 Yale U.             Harvard U.                Columbia U.              Cornel~
##  9 U.C., Berkeley      U.C. Hastings Law School  U.C., Los Angeles        Johns ~
## 10 U. of Oregon        Cornell U.                Yale U.                  U. of ~
## 11 UC., Los Angeles.   Georgetown U.             U. of Wisconsin, Madison U. of ~
##
## [[3]]
## # A tibble: 5 x 2
##   X1                                                                     X2
##   <chr>                                                               <int>
## 1 National Science Foundation Fellowships                               191
## 2 Fulbright Students                                                    117
## 3 Thomas J. Watson Fellows                                               72
## 4 Guggenheim Fellowships                                                 61
## 5 Rhodes Scholars (second highest number from a liberal arts college)    32
```

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?

This data does not seem in a tidy format at all. An example of this would be that rows are not observations in this format. We also notice that columns are not necessarily variables? (or at least not in a tidy way).

```
alumni_school <- data.frame(html_table(read_html("https://www.reed.edu/ir/success.html"))[2])
alumni_school
```

```
##                   MBAs                       JDs                     PhDs
## 1       U. of Chicago Lewis & Clark  Law School           U.C., Berkeley
## 2   Portland State U.            U.C., Berkeley         U. of Washington
## 3          Harvard U.              U. of Oregon            U. of Chicago
## 4    U. of Washington          U. of Washington              Stanford U.
## 5         Columbia U.               New York U.             U. of Oregon
## 6  U of Pennsylvania.             U. of Chicago               Harvard U.
## 7         Stanford U.                   Yale U.               Cornell U.
## 8             Yale U.                Harvard U.              Columbia U.
## 9      U.C., Berkeley  U.C. Hastings Law School        U.C., Los Angeles
## 10       U. of Oregon                Cornell U.                  Yale U.
## 11  UC., Los Angeles.             Georgetown U. U. of Wisconsin, Madison
##                              MDs
## 1     Oregon Health & Sci Univ.
## 2              U. of Washington
## 3     Washington U. (St. Louis)
## 4            UC., San Fransisco
## 5                   Stanford U.
## 6                   Harvard U..
## 7       Case Western Reserve U.
## 8                    Cornell U.
## 9              Johns Hopkins U.
## 10 U. of Minnesota, Twin Cities
## 11    U. of Southern California
```

c. Wrangle the data into a tidy format. Glimpse the resulting data frame.

```r
temp <- alumni_school %>% gather(key = "variable", value = "value")
tidy_alumni_school <- data.frame(school = unique(temp$value))
tidy_alumni_school %>% mutate(MBA = ifelse(school %in% alumni_school$MBAs, "Yes", "No"),
                              JD = ifelse(school %in% alumni_school$JDs, "Yes", "No"),
                              PhD = ifelse(school %in% alumni_school$PhDs, "Yes", "No"),
                              MD = ifelse(school %in% alumni_school$MDs, "Yes", "No"))
```

```
##                          school MBA  JD PhD  MD
## 1                 U. of Chicago Yes Yes Yes  No
## 2             Portland State U. Yes  No  No  No
## 3                    Harvard U. Yes Yes Yes  No
## 4              U. of Washington Yes Yes Yes Yes
## 5                   Columbia U. Yes  No Yes  No
## 6            U of Pennsylvania. Yes  No  No  No
## 7                   Stanford U. Yes  No Yes Yes
## 8                       Yale U. Yes Yes Yes  No
## 9                U.C., Berkeley Yes Yes Yes  No
## 10                 U. of Oregon Yes Yes Yes  No
## 11            UC., Los Angeles. Yes  No  No  No
## 12    Lewis & Clark  Law School  No Yes  No  No
## 13                  New York U.  No Yes  No  No
## 14     U.C. Hastings Law School  No Yes  No  No
## 15                   Cornell U.  No Yes Yes Yes
## 16                Georgetown U.  No Yes  No  No
## 17            U.C., Los Angeles  No  No Yes  No
## 18     U. of Wisconsin, Madison  No  No Yes  No
## 19    Oregon Health & Sci Univ.  No  No  No Yes
## 20    Washington U. (St. Louis)  No  No  No Yes
## 21           UC., San Fransisco  No  No  No Yes
## 22                  Harvard U..  No  No  No Yes
## 23      Case Western Reserve U.  No  No  No Yes
## 24             Johns Hopkins U.  No  No  No Yes
## 25 U. of Minnesota, Twin Cities  No  No  No Yes
## 26    U. of Southern California  No  No  No Yes
```

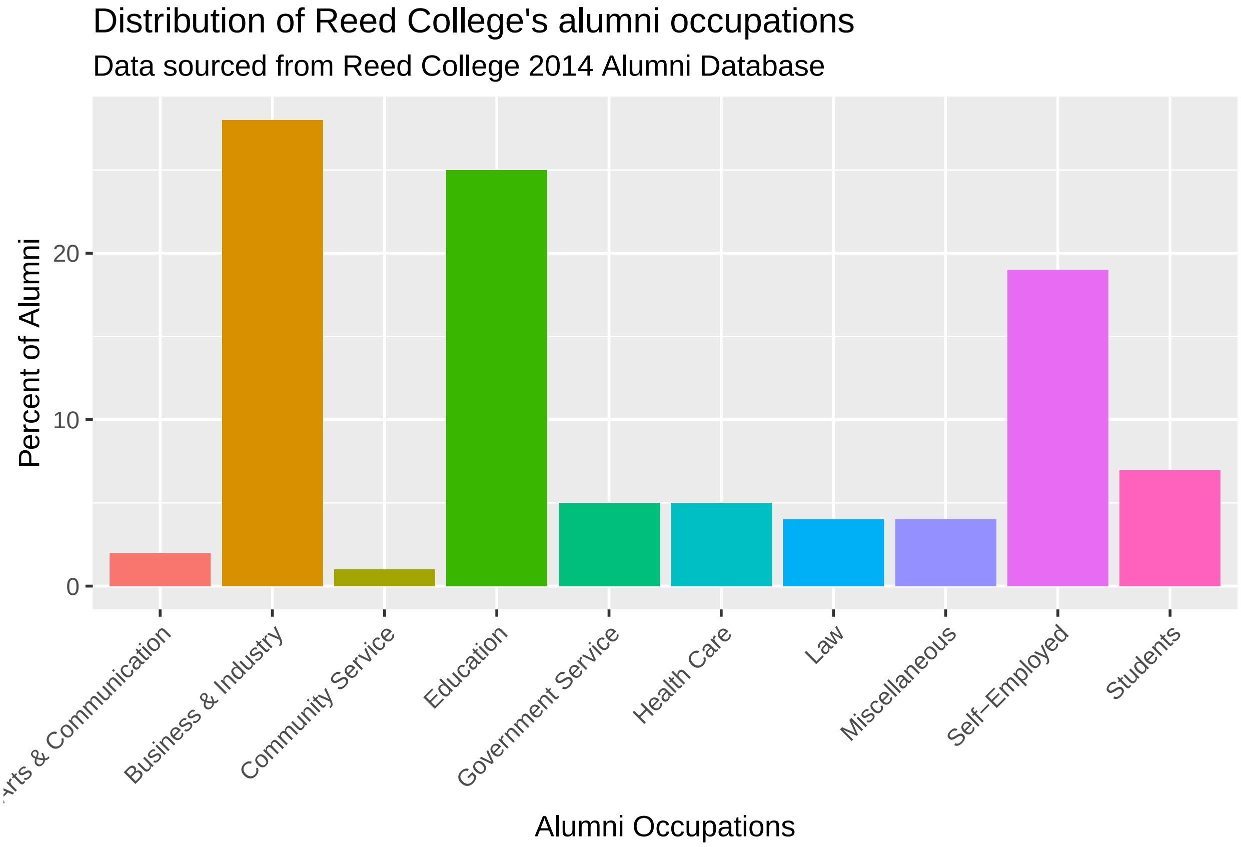
```r
rm(temp)
glimpse(tidy_alumni_school)
```

```
## Rows: 26
## Columns: 1
## $ school <chr> "U. of Chicago", "Portland State U.", "Harvard U.", "U. of Wash~
```

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

```r
# Hint: Use `parse_number()` within `mutate()` to fix one of the columns
alumni_occupation <- data.frame(html_table(read_html("https://www.reed.edu/ir/success.html"))[1])
alumni_occupation <- alumni_occupation %>% mutate(percent = parse_number(alumni_occupation$X2)) %>% sel
```

```
ggplot(data = alumni_occupation, mapping = aes(x = X1, y = percent, fill = X1)) +
  geom_col() +
  labs(x = "Alumni Occupations",
       y = "Percent of Alumni",
       title = "Distribution of Reed College's alumni occupations",
       subtitle = "Data sourced from Reed College 2014 Alumni Database") +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45, hjust = 1))
```



We can draw pretty good conclusions from this graph. It seems like a noticeably high majority of Reedies work in Business & Industry or Education or are self-employed. Almost every other field is under 5%, which tells us that there is also a noticeable portion of Reedies in various fields, but it is most likely going to be difficult to connect with others there due to the small percentage.

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

Do the following to clean up the data:

```
reed_grad <- data.frame(html_table(read_html("https://www.reed.edu/ir/gradrateshist.html"))[1])
```

- Rename the column names.

```
# Hint
colnames(reed_grad) <- c("enter_year", "cohort_count", "grad_4yr", "grad_5yr", "grad_6yr")
```

- Remove any extraneous rows.

```
# Hint
reed_grad <- reed_grad %>% slice(2:39)
reed_grad$grad_4yr <- parse_number(reed_grad$grad_4yr, na = c("", "NA"))
```

```
reed_grad$grad_5yr <- parse_number(reed_grad$grad_5yr, na = c("", "NA"))
reed_grad$grad_6yr <- parse_number(reed_grad$grad_6yr, na = c("", "NA"))
```

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

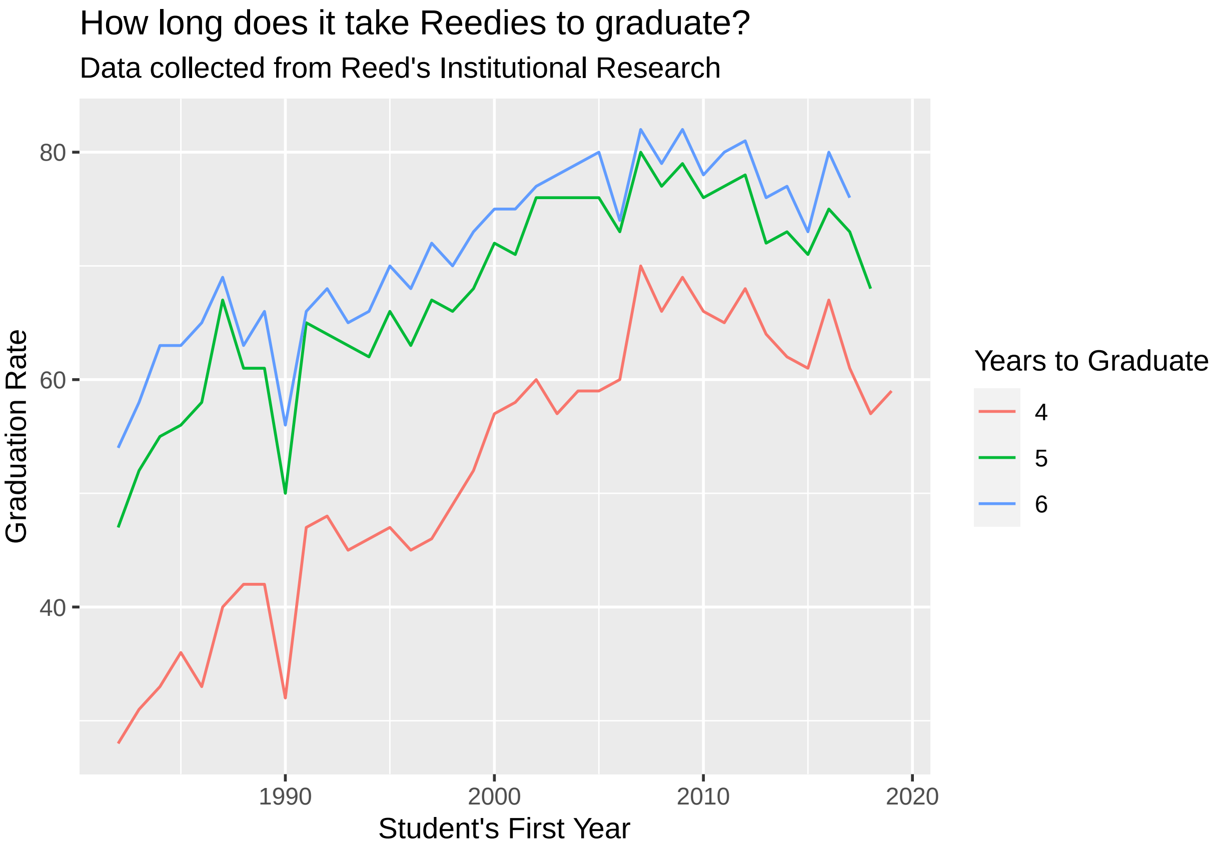
```
reed_grad <- reed_grad %>% gather(years_to_grad, grad_rate, starts_with("grad_")) %>%
  mutate(years_to_grad = as.character(parse_number(years_to_grad)),
         cohort_count = as.numeric(cohort_count),
         enter_year = as.numeric(enter_year))

reed_grad <- reed_grad[complete.cases(reed_grad),]
```

- Make sure each column has the correct class.

They do in this case for the purpose of the analysis in the following graph.

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

```
ggplot(data = reed_grad, mapping = aes(x = enter_year, y = grad_rate, color = years_to_grad)) +
  geom_line() +
  labs(x = "Student's First Year",
       y = "Graduation Rate",
       color = "Years to Graduate",
       title = "How long does it take Reedies to graduate?",
       subtitle = "Data collected from Reed's Institutional Research")
```



It seems like there generally is always a huge similarity between the ones who graduate in 5 years and in 6 years. It also is interesting how the ones who graduate in 4 years always have a gap between the other 2 regardless of what year it was. The overall graduation rate seems incredibly low, which is not surprising knowing how Reed likes to set its academic support and rigor, but also is very indicative of the college as an education. There seems to be a glaring issue with graduation rates, and it seems like there was work that was done to improve this over time.