Lab 5

Jefferson Ratliff

Math 241, Week 6

```r
# Put all necessary libraries here
library(tidyverse)
library(rnoaa)
library(rvest)
library(httr)
library(dplyr)
library(ggplot2)
library(reshape2)
library(data.table)
library(lubridate)
```

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.

```r
library(rnoaa)
```

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

```r
options(noaakey = "VcGYvlXNkEqHZJdSpYslZaIUcUhkIvKK")
```

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 the 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.

```
options(noaakey = "VcGYvlXNkEqHZJdSpYslZaIUcUhkIvKK")

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"
```

```
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"
```

```
precip_se_pdx <- ncdc(datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2023-01-01",
                      enddate = "2023-01-31")

precip_se_pdx
```

```
## $meta
## $meta$totalCount
## [1] 31
##
## $meta$pageCount
## [1] 25
##
## $meta$offset
## [1] 1
##
##
## $data
## # A tibble: 25 x 8
##    date                datatype station           value fl_m  fl_q  fl_so fl_t
##    <chr>               <chr>    <chr>             <int> <chr> <chr> <chr> <chr>
##  1 2023-01-01T00:00:00 PRCP     GHCND:US1ORMT0006     8 ""    ""    N     0830
##  2 2023-01-02T00:00:00 PRCP     GHCND:US1ORMT0006     0 ""    ""    N     0700
##  3 2023-01-03T00:00:00 PRCP     GHCND:US1ORMT0006    43 ""    ""    N     0826
##  4 2023-01-04T00:00:00 PRCP     GHCND:US1ORMT0006    20 ""    ""    N     0826
##  5 2023-01-05T00:00:00 PRCP     GHCND:US1ORMT0006    46 ""    ""    N     0822
##  6 2023-01-06T00:00:00 PRCP     GHCND:US1ORMT0006    46 ""    ""    N     0822
##  7 2023-01-07T00:00:00 PRCP     GHCND:US1ORMT0006    25 ""    ""    N     0822
##  8 2023-01-08T00:00:00 PRCP     GHCND:US1ORMT0006    99 ""    ""    N     0822
##  9 2023-01-09T00:00:00 PRCP     GHCND:US1ORMT0006   114 ""    ""    N     0806
## 10 2023-01-10T00:00:00 PRCP     GHCND:US1ORMT0006    33 ""    ""    N     0806
## # i 15 more rows
##
## attr(,"class")
## [1] "ncdc_data"
```

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`.

```
precip_se_pdx <- ncdc(datasetid = "GHCND", datatypeid = "PRCP",
                      startdate = "2024-01-01",
                      enddate = "2024-01-31",
                      stationid = "GHCND:US1ORMT0006",
```

```r
                      limit = 1000)

precip_se_pdx_data <- precip_se_pdx$data

print(precip_se_pdx_data)
```

```
## # A tibble: 31 x 8
##    date                datatype station           value fl_m  fl_q  fl_so fl_t
##    <chr>               <chr>    <chr>             <int> <chr> <chr> <chr> <chr>
##  1 2024-01-01T00:00:00 PRCP     GHCND:US1ORMT0006     0 "T"   ""    N     0747
##  2 2024-01-02T00:00:00 PRCP     GHCND:US1ORMT0006     0 ""    ""    N     0700
##  3 2024-01-03T00:00:00 PRCP     GHCND:US1ORMT0006    58 ""    ""    N     0842
##  4 2024-01-04T00:00:00 PRCP     GHCND:US1ORMT0006   107 ""    ""    N     0847
##  5 2024-01-05T00:00:00 PRCP     GHCND:US1ORMT0006    28 ""    ""    N     0835
##  6 2024-01-06T00:00:00 PRCP     GHCND:US1ORMT0006   135 ""    ""    N     0836
##  7 2024-01-07T00:00:00 PRCP     GHCND:US1ORMT0006    97 ""    ""    N     0738
##  8 2024-01-08T00:00:00 PRCP     GHCND:US1ORMT0006    56 ""    ""    N     0840
##  9 2024-01-09T00:00:00 PRCP     GHCND:US1ORMT0006   221 ""    ""    N     0840
## 10 2024-01-10T00:00:00 PRCP     GHCND:US1ORMT0006   157 ""    ""    N     0845
## # i 21 more rows
```

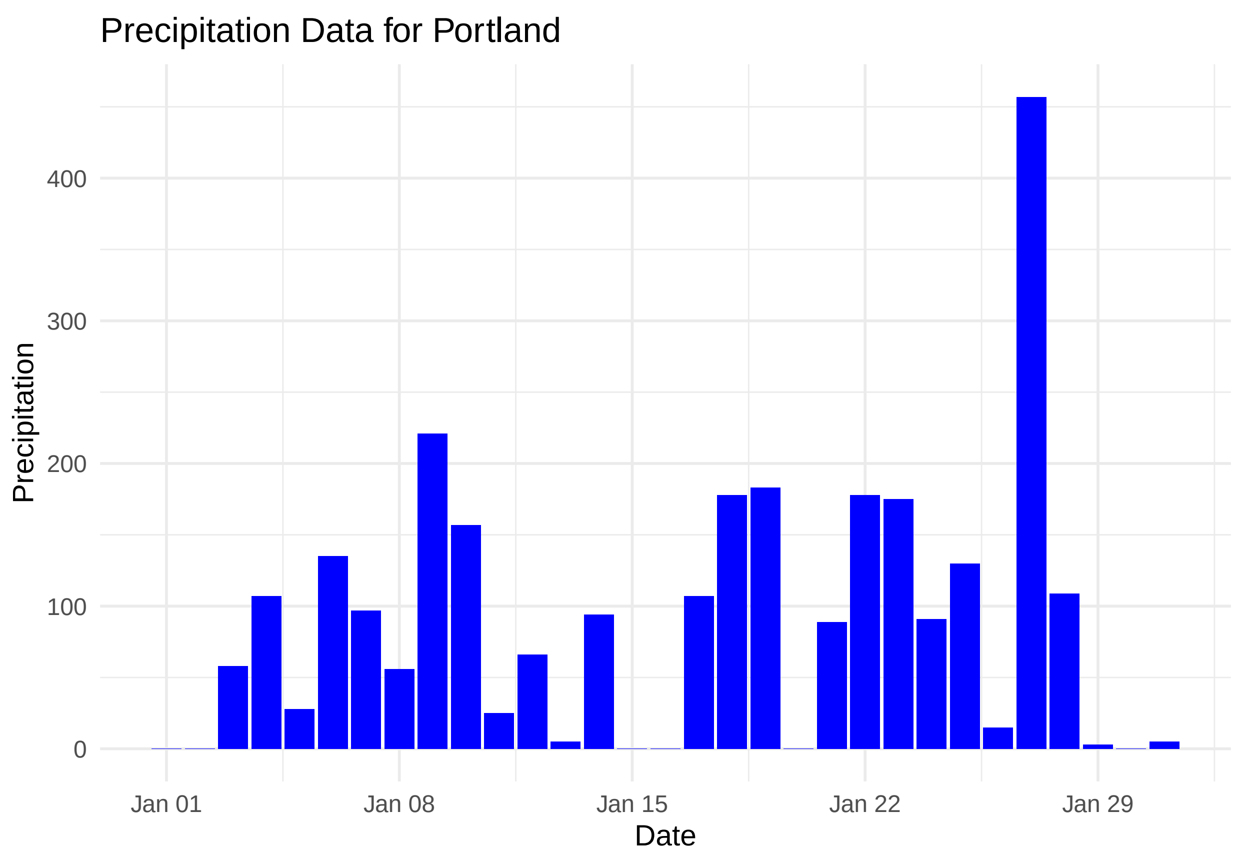
e. Use `ymd_hms()` in the package `lubridate` to wrangle the date column into the correct format.

```r
precip_se_pdx_data$date <- ymd_hms(precip_se_pdx_data$date)

print(precip_se_pdx_data)
```

```
## # A tibble: 31 x 8
##    date                datatype station           value fl_m  fl_q  fl_so fl_t
##    <dttm>              <chr>    <chr>             <int> <chr> <chr> <chr> <chr>
##  1 2024-01-01 00:00:00 PRCP     GHCND:US1ORMT0006     0 "T"   ""    N     0747
##  2 2024-01-02 00:00:00 PRCP     GHCND:US1ORMT0006     0 ""    ""    N     0700
##  3 2024-01-03 00:00:00 PRCP     GHCND:US1ORMT0006    58 ""    ""    N     0842
##  4 2024-01-04 00:00:00 PRCP     GHCND:US1ORMT0006   107 ""    ""    N     0847
##  5 2024-01-05 00:00:00 PRCP     GHCND:US1ORMT0006    28 ""    ""    N     0835
##  6 2024-01-06 00:00:00 PRCP     GHCND:US1ORMT0006   135 ""    ""    N     0836
##  7 2024-01-07 00:00:00 PRCP     GHCND:US1ORMT0006    97 ""    ""    N     0738
##  8 2024-01-08 00:00:00 PRCP     GHCND:US1ORMT0006    56 ""    ""    N     0840
##  9 2024-01-09 00:00:00 PRCP     GHCND:US1ORMT0006   221 ""    ""    N     0840
## 10 2024-01-10 00:00:00 PRCP     GHCND:US1ORMT0006   157 ""    ""    N     0845
## # i 21 more rows
```

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(precip_se_pdx_data, aes(x = date, y = value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Precipitation Data for Portland",
       x = "Date",
       y = "Precipitation") +
  theme_minimal()
```



```r
days_without_rain <- sum(precip_se_pdx_data$value == 0)
cat("Number of days without rain:", days_without_rain, "\n")
```

```
## Number of days without rain: 6
```

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?

```r
ggplot(precip_se_pdx_data, aes(x = date, y = value)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Precipitation Data for Portland (January)",
       x = "Date",
       y = "Precipitation") +
  theme_minimal() +
  facet_wrap(~ (date), scales = "free_y", ncol = 1)
```

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

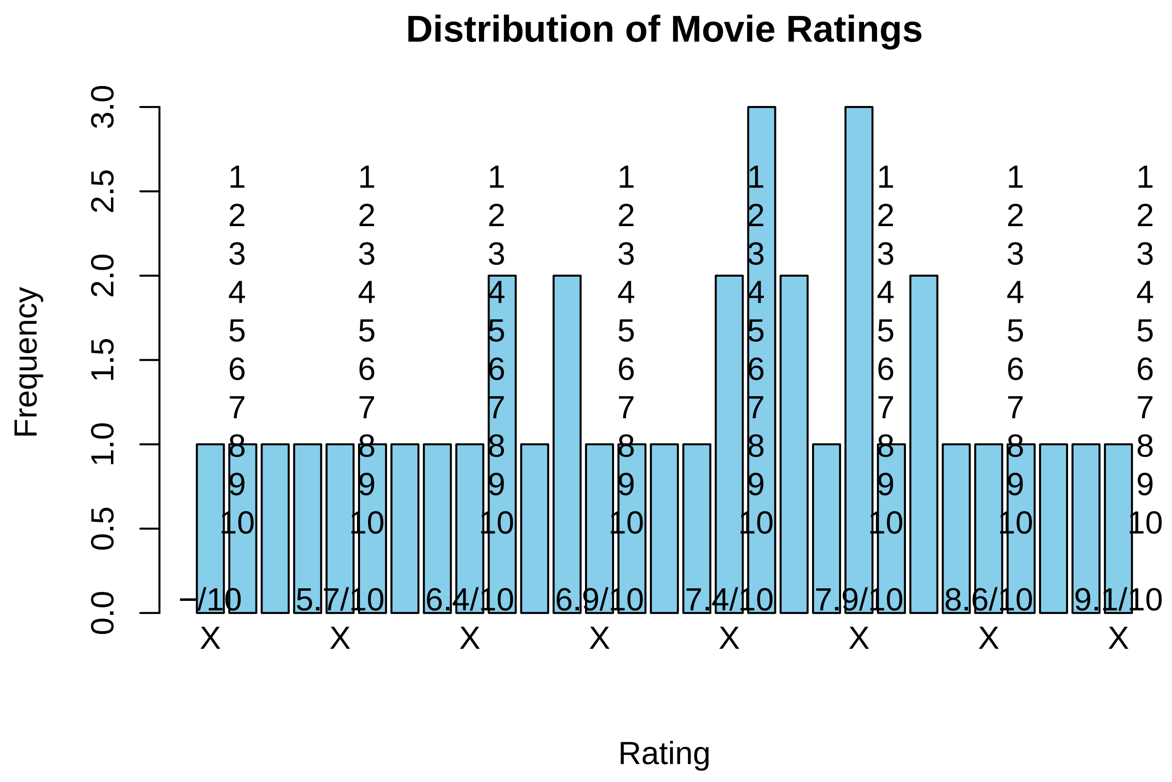
Here are some potential API wrapper packages. Feel free to use one not included in this list for Problem 2.

- `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

```r
url <- "https://www.imdb.com/movies-in-theaters/"

page <- read_html(url)

titles <- page %>%
  html_nodes(".title") %>%
  html_text() %>%
  trimws()

ratings <- page %>%
  html_nodes(".rating") %>%
  html_text() %>%
  trimws()

valid_data <- complete.cases(titles, ratings)
titles <- titles[valid_data]
ratings <- ratings[valid_data]

movie_data <- data.frame(title = titles, rating = ratings)


summary_stats <- summary(movie_data$rating)
cat("\nSummary Statistics:\n", summary_stats)
```

```
##
## Summary Statistics:
##  38 character character
```

```
barplot(table(movie_data$rating),
        main = "Distribution of Movie Ratings",
        xlab = "Rating",
        ylab = "Frequency",
        col = "skyblue",
        border = "black")
```



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

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

champ_table1 <- html_table(tables[[1]], fill = TRUE)
print(champ_table1)
```

```
## # 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%
```

```
champ_table2 <- html_table(tables[[2]], fill = TRUE)
print(champ_table2)
```

```
## # 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 ~
```

```
champ_table3 <- html_table(tables[[3]], fill = TRUE)
print(champ_table3)
```

```
## # 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?

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

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

champ_table2 <- html_table(tables[[2]], fill = TRUE)

print(champ_table2)
```

```
## # 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 ~
```

The data frame obtained from web scraping may not adhere to a tidy format due to various common issues. These include the presence of multiple variables in a single column, the simultaneous representation of variables in both rows and columns, redundant information, potential interference from headers or footers in the table, the existence of empty or merged cells, and challenges posed by a non-tabular layout on the web page. To address these issues and achieve a tidy format, careful inspection of the data frame structure is necessary. Utilizing functions from the tidyverse package in R, such as gather(), spread(), and mutate(), can assist in reshaping the data, allowing for more straightforward analysis and visualization.

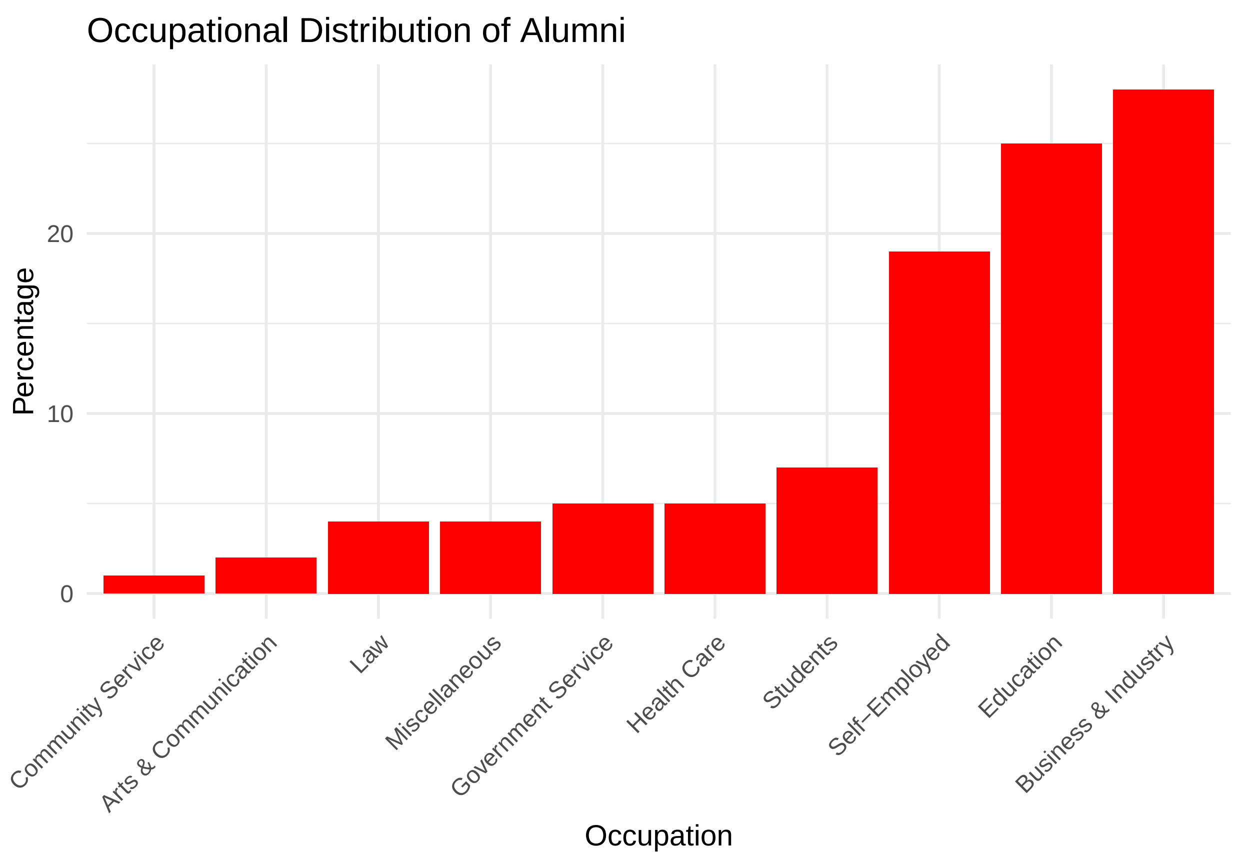
c. Wrangle the data into a tidy format. Glimpse the resulting data frame.

```r
tidy_champ_table2 <- champ_table2 %>%
  as_tibble() %>%
  gather(key = "Variable", value = "Value", -1)

glimpse(tidy_champ_table2)
```

```
## Rows: 33
## Columns: 3
## $ MBAs     <chr> "U. of Chicago", "Portland State U.", "Harvard U.", "U. of Wa~
## $ Variable <chr> "JDs", "JDs", "JDs", "JDs", "JDs", "JDs", "JDs", "JDs", "JDs"~
## $ Value    <chr> "Lewis & Clark  Law School", "U.C., Berkeley", "U. of Oregon"~
```

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
data <- data.frame(
  Occupation = c("Business & Industry", "Education", "Self-Employed", "Students",
                 "Government Service", "Health Care", "Law", "Miscellaneous",
                 "Arts & Communication", "Community Service"),
  Percentage = c("28%", "25%", "19%", "7%", "5%", "5%", "4%", "4%", "2%", "1%")
)

data <- data %>%
  mutate(Percentage = parse_number(Percentage))


occupational_plot <- ggplot(data, aes(x = reorder(Occupation, Percentage), y = Percentage)) +
  geom_bar(stat = "identity", fill = "red") +
  labs(title = "Occupational Distribution of Alumni",
       x = "Occupation",
       y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(occupational_plot)
```



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

Do the following to clean up the data:

- Rename the column names.

```
reed_grad_url <- "https://www.reed.edu/ir/gradrateshist.html"
grad_rate_data <- reed_grad_url %>%
  read_html() %>%
  html_nodes(css = "table")

grad_rate_table <- html_table(grad_rate_data[[1]], fill = TRUE)
colnames(grad_rate_table) <- c("Entering_Class_Year", "Cohort_Size", "4", "5", "6")
```

- Remove any extraneous rows.

```
grad_rate_table <- grad_rate_table %>%
  filter(!row_number() %in% c(1))
```

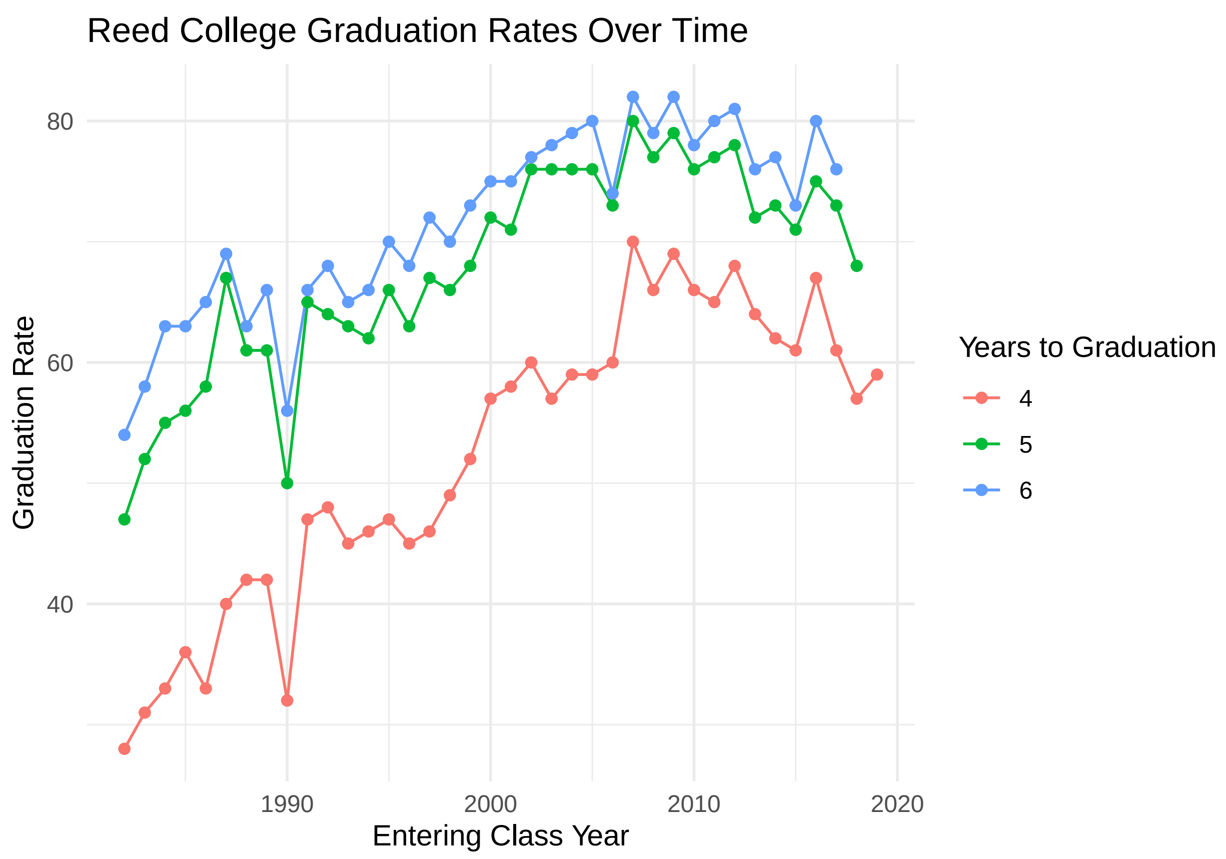
- Reshape the data so that there are columns for
  - Entering class year
  - Cohort size
  - Years to graduation
  - Graduation rate
- Make sure each column has the correct class.

```r
grad_rate_table <- grad_rate_table %>%
  pivot_longer(cols = c("4", "5", "6"),
               names_to = "Years_to_Graduation",
               values_to = "Graduation_Rate") %>%
  mutate(Graduation_Rate = parse_number(Graduation_Rate)) %>%
  filter(!is.na(Graduation_Rate))

print(grad_rate_table)
```

```
## # A tibble: 111 x 4
##    Entering_Class_Year Cohort_Size Years_to_Graduation Graduation_Rate
##    <chr>               <chr>       <chr>                         <dbl>
##  1 2019                393         4                                59
##  2 2018                361         4                                57
##  3 2018                361         5                                68
##  4 2017                411         4                                61
##  5 2017                411         5                                73
##  6 2017                411         6                                76
##  7 2016                353         4                                67
##  8 2016                353         5                                75
##  9 2016                353         6                                80
## 10 2015                418         4                                61
## # i 101 more rows
```

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

```r
grad_rate_table$Entering_Class_Year <- as.numeric(grad_rate_table$Entering_Class_Year)


ggplot(grad_rate_table, aes(x = Entering_Class_Year, y = Graduation_Rate, color = Years_to_Graduation))
  geom_line() +
  geom_point() +
  labs(title = "Reed College Graduation Rates Over Time",
       x = "Entering Class Year",
       y = "Graduation Rate",
       color = "Years to Graduation") +
  theme_minimal()
```



The graduation rate has seen a significant upward trend over the past four decades and is effected by how long you spend to get your degree as students who stay at Reed for longer then 4 years have a much higher graduation rate. There now has been consistent decline in the late 2010s' in the gradulation rates for students who spend 5 years on their degree.