Lab 5

Math 241, Week 6

```
# Put all necessary libraries here
library(tidyverse)
library(rnoaa)
library(rvest)
library(httr)
library(ggplot2)
```

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:

```
options(noaakey = "YHYMXktJADOYAHDGHqMwoIxtVCzdnHXT")
```

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.

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

```
## $meta
##   offset count limit
## 1      1     5    25
##
## $data
##      mindate    maxdate                                     name datacoverage
## 1 1750-02-01 2024-02-26                            Precipitation            1
## 2 1840-05-01 2024-02-26                                 Snowfall            1
## 3 1857-01-18 2024-02-26                               Snow depth            1
## 4 1952-07-01 2024-02-26 Water equivalent of snow on the ground            1
## 5 1998-06-01 2024-02-26           Water equivalent of snowfall            1
##     id
## 1 PRCP
## 2 SNOW
## 3 SNWD
## 4 WESD
## 5 WESF
##
## attr(,"class")
## [1] "ncdc_datatypes"
```

```
# Now grab the data using ncdc()
precip_se_pdx <- ncdc(
                      datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2024-01-01",
                      enddate = "2024-01-31")

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 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 15 more rows
##
## attr(,"class")
## [1] "ncdc_data"
```

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

```
class(precip_se_pdx)
```

```
## [1] "ncdc_data"
```

```
precip_se_pdx_data <- precip_se_pdx$data

precip_se_pdx_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 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 15 more rows
```

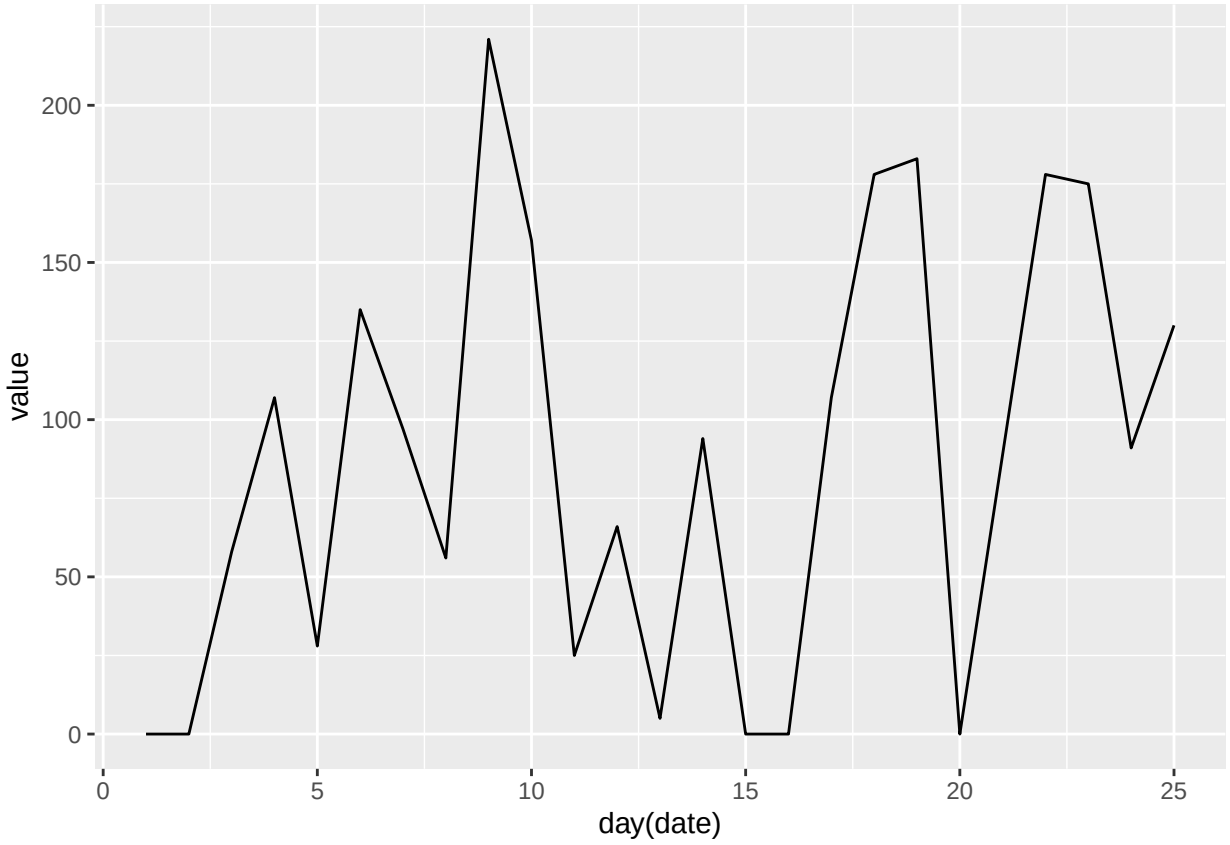
precip_se_pdx class is listed as "ncdc_data" In the environemnt it shows up as a list

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

```
library(lubridate)

precip_pdx_data_clean <- precip_se_pdx_data %>%
  mutate(date = ymd_hms(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?

```
precip_pdx_data_clean %>%
  ggplot(aes(x = day(date), y = value))+
  geom_line()
```

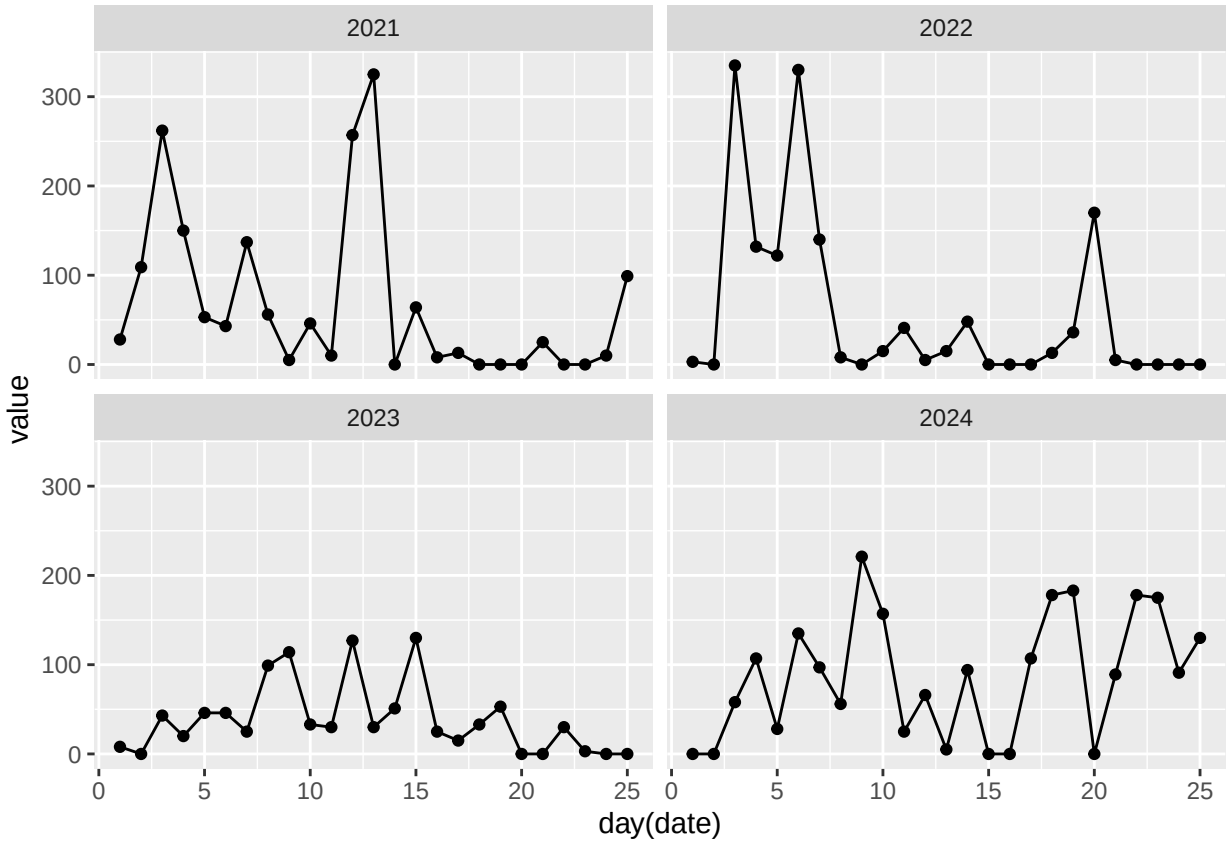


No this claim is false. We can see that there were multiplke days where there was no rain in January.

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_24 <- ncdc(
                      datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2024-01-01",
                      enddate = "2024-01-31")

precip_23 <- ncdc(
                      datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2023-01-01",
                      enddate = "2023-01-31")
precip_22 <- ncdc(
```

```
                      datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2022-01-01",
                      enddate = "2022-01-31")

precip_21 <- ncdc(
                      datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = "2021-01-01",
                      enddate = "2021-01-31")

precip_24_data <- precip_24$data
precip_23_data <- precip_23$data
precip_22_data <- precip_22$data
precip_21_data <- precip_21$data


precip <- bind_rows(precip_24_data, precip_23_data, precip_22_data, precip_21_data)

precip_clean <- precip %>%
  mutate(date = ymd_hms(date))

precip_clean %>%
  ggplot(aes(x = day(date), y = value))+
  geom_point()+
  geom_line()+
  facet_wrap(~year(date))
```



Trends: it seems as though over the last few years the amount of rain in January has evened out. It still rains white a bit in January but the highest levels in 2023 and 2024 r emuch lower than 2021 and 2022.

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

```
library(rfishbase)
library(dplyr)
```

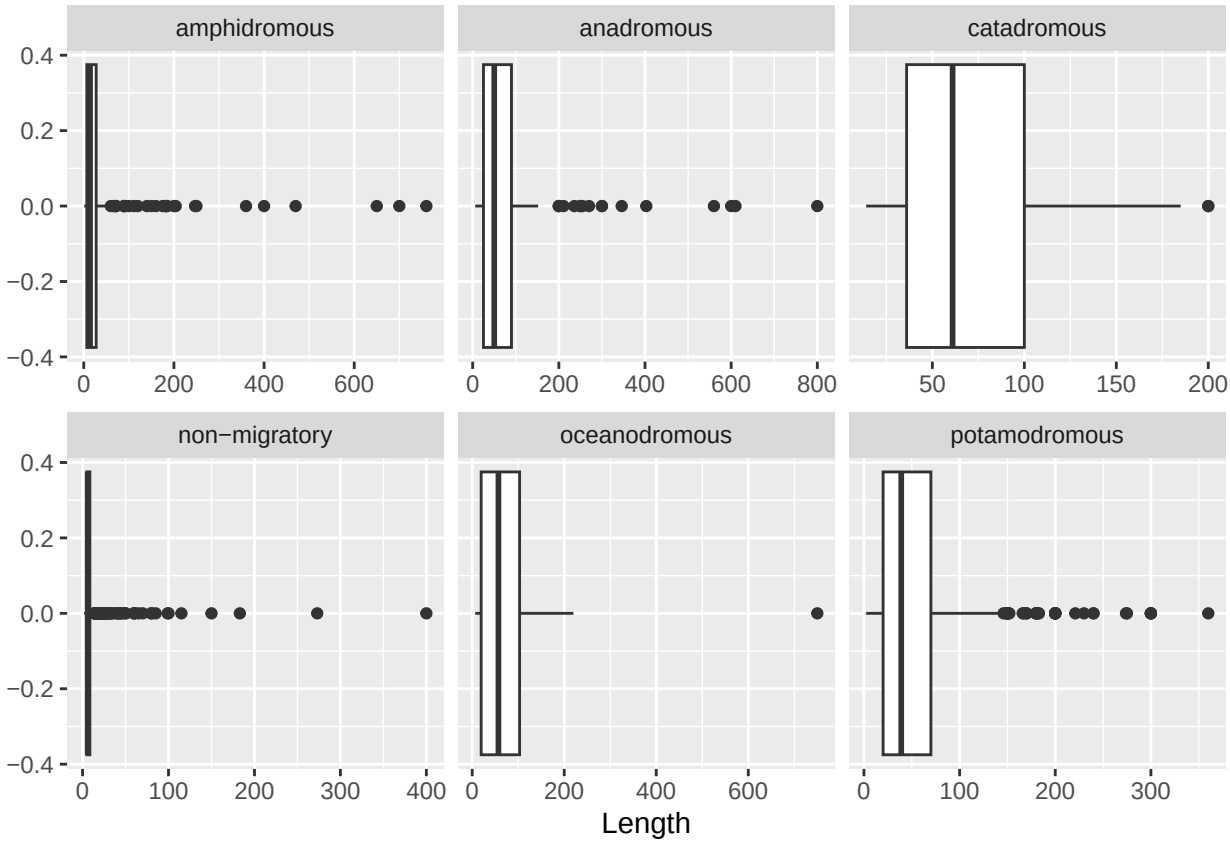
```r
spec <- fb_tbl("species")

fresh_spec <- spec %>%
  filter(Fresh == "1")

fresh_spec %>%
  count(AnaCat) %>%
  arrange(desc(n))
```

```
## # A tibble: 14 x 2
##    AnaCat                n
##    <chr>             <int>
##  1  <NA>              14142
##  2 " "                 1353
##  3 "non-migratory"      896
##  4 "potamodromous"      575
##  5 "amphidromous"       366
##  6 "anadromous"         163
##  7 "catadromous"         77
##  8 "oceanodromous"       17
##  9 "anadromous?"          2
## 10 "amphidromous?"        1
## 11 "diadromous"           1
## 12 "oceano-estuarine"     1
## 13 "potamodromous?"       1
## 14 "unknown"              1
```

```r
fresh_spec_mig <- fresh_spec %>%
  filter(AnaCat %in% c("non-migratory","oceanodromous", "potamodromous", "amphidromous", "anadromous",

fresh_spec_mig %>%
  group_by(AnaCat) %>%
  summarise(Mean = mean(Length, na.rm = TRUE),
            Low = quantile(Length, 0.1, na.rm = TRUE),
            High = quantile(Length, 0.9, na.rm = TRUE),
            Count = n())
```

```
## # A tibble: 6 x 5
##   AnaCat          Mean   Low  High Count
##   <chr>          <dbl> <dbl> <dbl> <int>
## 1 amphidromous   33.6   4.09   50    366
## 2 anadromous     85.8  12.1   200    163
## 3 catadromous    74.1  30     150     77
## 4 non-migratory   9.75  3.51   15    896
## 5 oceanodromous 107.   12.8   178     17
## 6 potamodromous  55.3  11     122.   575
```

```r
fresh_spec_mig %>%
  ggplot(aes(x = Length)) +
  geom_boxplot() +
  facet_wrap(~ AnaCat, scales = "free_x")
```



Conclusions:

Migration patterns may have an effect on fish length. This would need to be tested further however as we can see nonmigratory fish tend to be much smaller than other types of fish. Additionally, oceanodromous fish were on average the largest in length however they are also the smallest sampled group and as such this data may be skewed. Anadromous fish have the most varaibility it seems when it comes to length.

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.

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

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

tbl1 <- html_table(tables[[1]], fill = TRUE)
tbl1
```

```
## # 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%
```

```
tbl2 <- html_table(tables[[2]], fill = TRUE)
tbl2
```

```
## # 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 ~
```

```
tbl3 <- html_table(tables[[3]], fill = TRUE)
tbl3
```

```
## # 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?

Not sure if I am supposed to answer this question or not but the reason its format inst tidy is because every observation doesnt have its own rows.

Each variable must have its own column. Each observation must have its own row. Each value must have its own cell.

```
tbl2
```

```
## # 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 ~
```

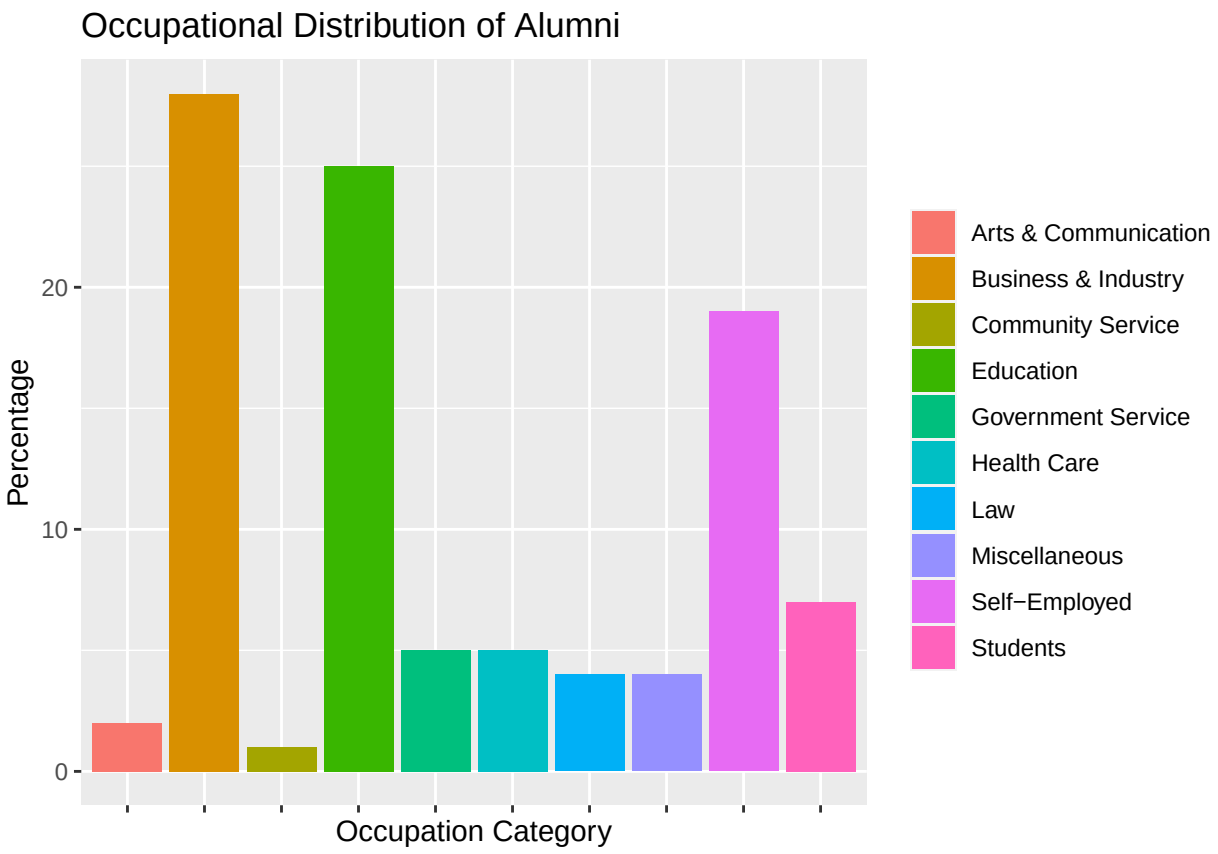
c. Wrangle the data into a tidy format. Glimpse the resulting data frame.

```
tbl2_pivot <- tbl2 %>%
   pivot_longer(c(`MBAs`, `JDs`,`PhDs`,`MDs`), names_to = "Degree",  values_to = "School")

glimpse(tbl2_pivot)
```

```
## Rows: 44
## Columns: 2
## $ Degree <chr> "MBAs", "JDs", "PhDs", "MDs", "MBAs", "JDs", "PhDs", "MDs", "MB~
## $ School <chr> "U. of Chicago", "Lewis & Clark  Law School", "U.C., Berkeley",~
```

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

alum <- tbl1 %>%
  mutate(X2 = parse_number(X2))

alum %>%
  ggplot(aes(x = X1, y = X2, fill = X1)) +
  geom_bar(stat = "identity") +
  labs(
    y = "Percentage",
    x = "Occupation Category",
    title = "Occupational Distribution of Alumni") +
  theme(
    axis.text.x = element_blank(),
    legend.title = element_blank())
```



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.

```
url_2 <- "https://www.reed.edu/ir/gradrateshist.html"

grads <- url_2 %>%
  read_html() %>%
  html_nodes(css = "table")

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

colnames(grad_rates) <- c("first_year_fall", "count", "four_years", "five_years", "six_years")

grad_parse <- grad_rates %>% mutate(
                                four_years = parse_number(four_years),
                                five_years = parse_number(five_years),
                                six_years = parse_number(six_years))
grad_parse
```

```
## # A tibble: 39 x 5
##    first_year_fall                      count four_years five_years six_years
```

```
##    <chr>                                    <chr>      <dbl>      <dbl>     <dbl>
##  1 First-year students who entered fall o~ Numb~          4          5         6
##  2 2019                                    393           59         NA        NA
##  3 2018                                    361           57         68        NA
##  4 2017                                    411           61         73        76
##  5 2016                                    353           67         75        80
##  6 2015                                    418           61         71        73
##  7 2014                                    346           62         73        77
##  8 2013                                    354           64         72        76
##  9 2012                                    320           68         78        81
## 10 2011                                    372           65         77        80
## # i 29 more rows
```

- Remove any extraneous rows.

```
grad_rm <- grad_parse %>% filter(row_number() != 1)

grad_rm
```

```
## # A tibble: 38 x 5
##    first_year_fall count four_years five_years six_years
##    <chr>           <chr>      <dbl>      <dbl>     <dbl>
##  1 2019            393           59         NA        NA
##  2 2018            361           57         68        NA
##  3 2017            411           61         73        76
##  4 2016            353           67         75        80
##  5 2015            418           61         71        73
##  6 2014            346           62         73        77
##  7 2013            354           64         72        76
##  8 2012            320           68         78        81
##  9 2011            372           65         77        80
## 10 2010            373           66         76        78
## # i 28 more rows
```

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

```
grad_long <- grad_rm %>%
  pivot_longer(c("four_years", "five_years", "six_years"),
               names_to = "years_to_graduation",
               values_to = "graduation_rate") %>%
        mutate(cohort_size = as.numeric(count)) %>%
   select(first_year_fall, cohort_size, years_to_graduation, graduation_rate)
grad_long
```

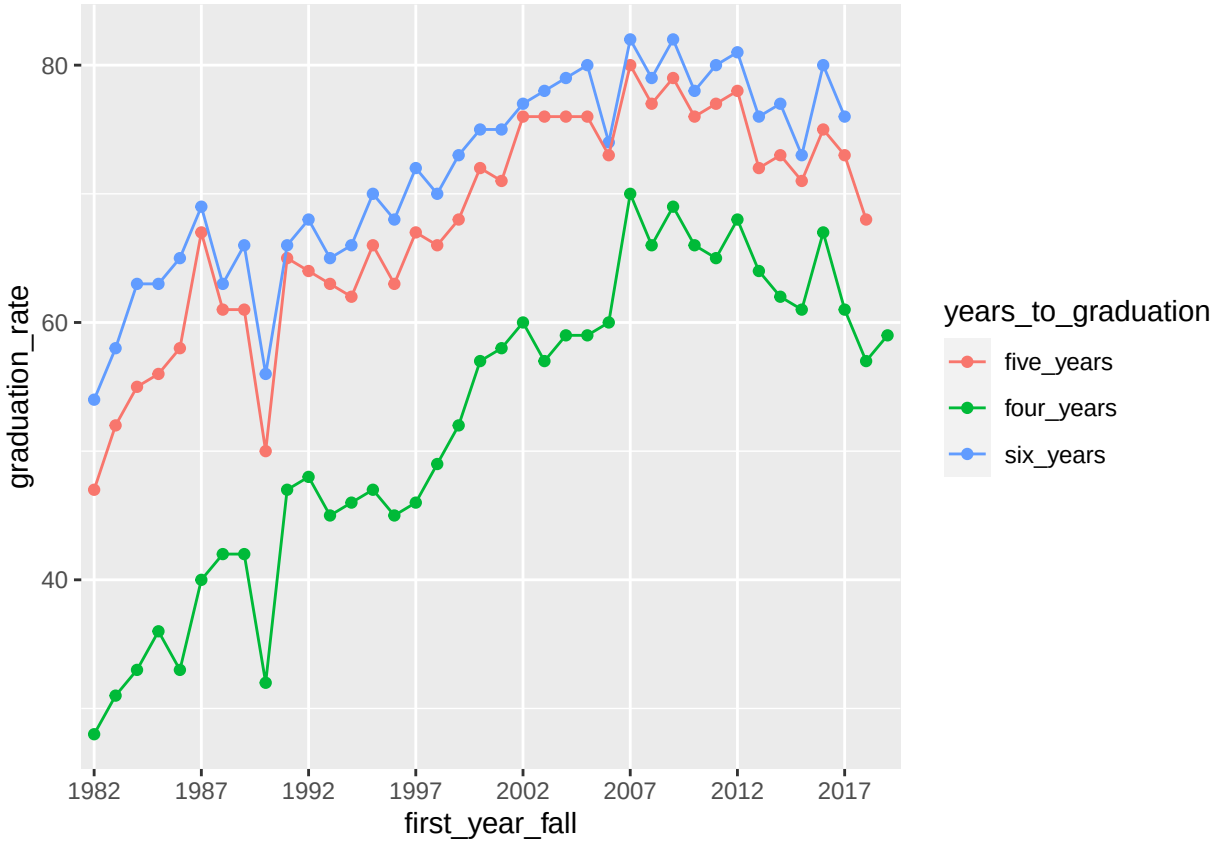
```
## # A tibble: 114 x 4
##    first_year_fall cohort_size years_to_graduation graduation_rate
##    <chr>                 <dbl> <chr>                         <dbl>
```

```
##  1 2019                393 four_years                59
##  2 2019                393 five_years                NA
##  3 2019                393 six_years                 NA
##  4 2018                361 four_years                57
##  5 2018                361 five_years                68
##  6 2018                361 six_years                 NA
##  7 2017                411 four_years                61
##  8 2017                411 five_years                73
##  9 2017                411 six_years                 76
## 10 2016                353 four_years                67
## # i 104 more rows
```

- Make sure each column has the correct class.

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

```
grad_long %>%
ggplot(aes(x= first_year_fall, y= graduation_rate, group=years_to_graduation, color = years_to_graduati
  geom_point() +
  geom_line() +
  scale_x_discrete(breaks = seq(1982,2019, by = 5))
```



Over time, graduation rates have risen. Additionally the more years it takes for students to graduate the higher the graduation rate.