Lab 5

Ashwin Dev

Math 241, Week 6

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

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 = "mEqPbHQRJEpSMWZAMXLxLNawcEsSkMUR")
```

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
```

```
nrow(mult_stations)
```

```
## [1] 25
```

There are 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
year <- 2023

# specifying the datatypeid for precipitation, and including startdate and enddate
precip_se_pdx <- ncdc(datasetid = "GHCND",
                      stationid = "GHCND:US1ORMT0006",
                      datatypeid = "PRCP",
                      startdate = paste0(year, "-01-01"),
                      enddate = paste0(year, "-01-31"),
                      limit = 1000)
```

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`.

---

precip_se_dpx is a list.

```
precip_se_pdx_data <- precip_se_pdx$data
```

---

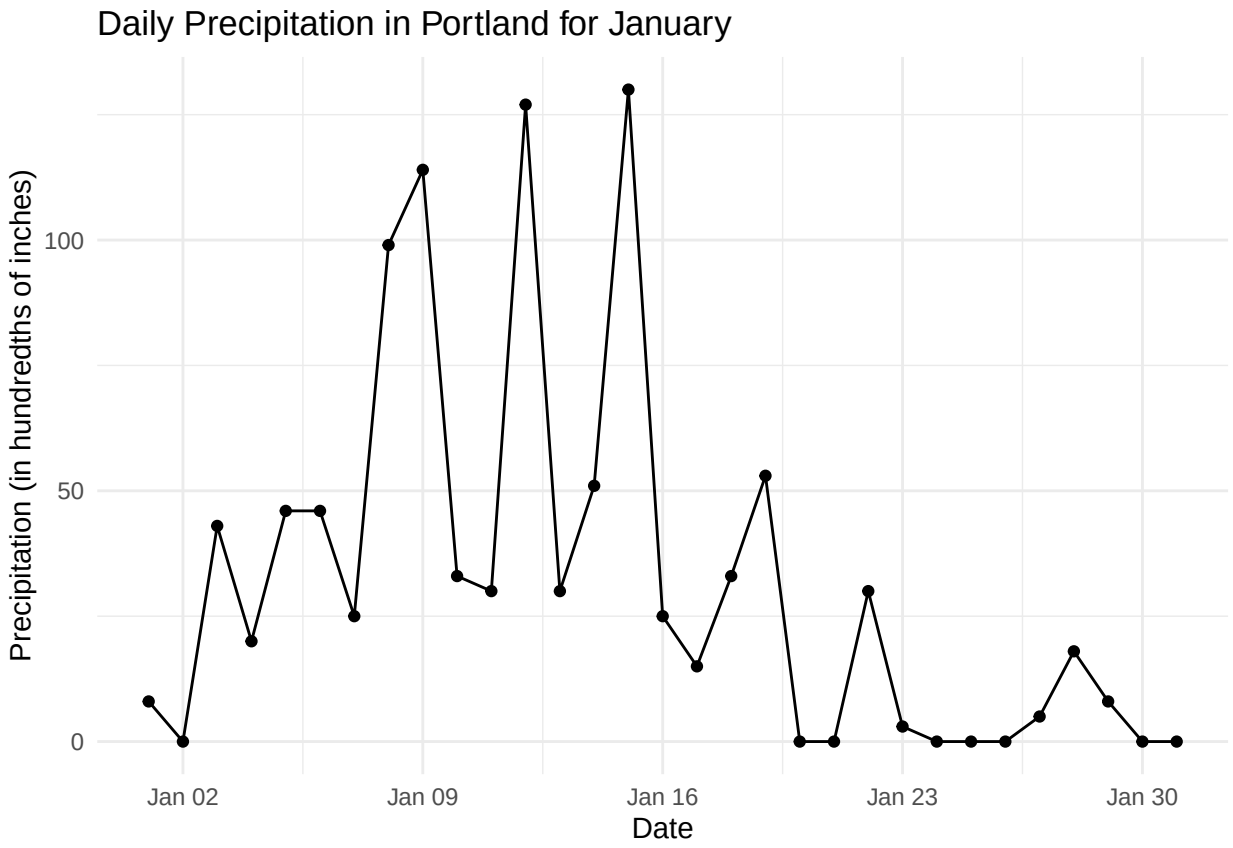
e. Use `ymd_hms()` in the package `lubridate` to wrangle the date column into the correct format.

```
precip_se_pdx_data$date <- ymd_hms(precip_se_pdx_data$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(precip_se_pdx_data, aes(x = date, y = value)) +
  geom_line() +
  geom_point() + # adds points to lines
  labs(title = "Daily Precipitation in Portland for January",
       x = "Date",
       y = "Precipitation (in hundredths of inches)") +
  theme_minimal()
```



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?

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.

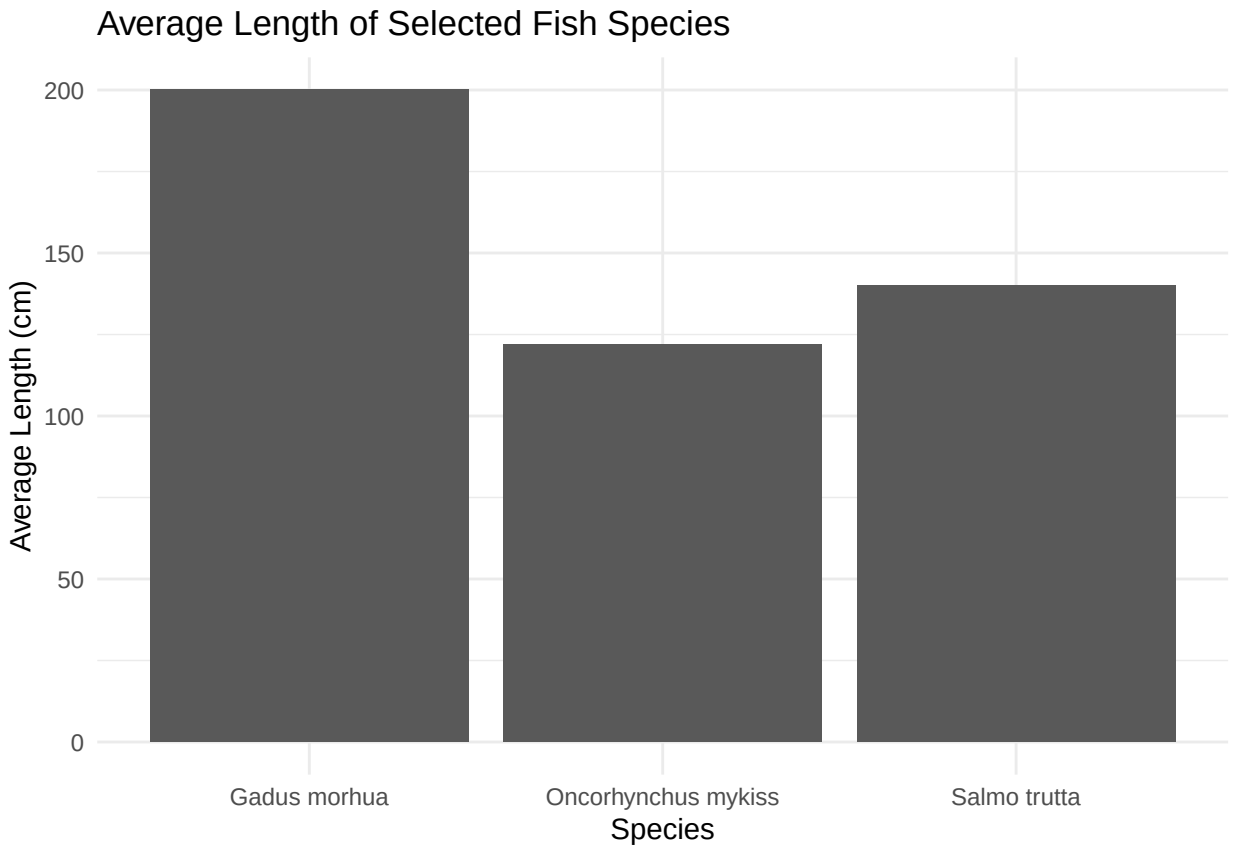
Once you have grabbed the data, do any necessary wrangling to graph it and/or produce some summary statistics. Draw some conclusions from your graph and summary statistics.

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
library(rfishbase)
library(ggplot2)

# A selection of fish species
species_list <- c("Gadus morhua", "Oncorhynchus mykiss", "Salmo trutta")

# length data for each species
length_data <- lapply(species_list, function(species) {
  data <- species(species)
  data.frame(Species = species, Length = as.numeric(data$Length))
})

# combines the data into a single data frame
length_data_combined <- do.call(rbind, length_data)

# average length for each species
average_lengths <- aggregate(Length ~ Species, length_data_combined, mean, na.rm = TRUE)

ggplot(average_lengths, aes(x = Species, y = Length)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Length of Selected Fish Species",
       x = "Species",
       y = "Average Length (cm)") +
  theme_minimal()
```



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.

```
library(rvest)

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

page <- read_html(url)

tables <- html_table(page)

# first table
first_table <- tables[[1]]

#  the second table:
second_table <- tables[[2]]

print(head(first_table))
```

```
## # A tibble: 6 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%
```

```
print(head(second_table))
```

```
## # A tibble: 6 x 4
##   MBAs               JDs                        PhDs             MDs
##   <chr>              <chr>                      <chr>            <chr>
## 1 U. of Chicago      Lewis & Clark  Law School U.C., Berkeley   Oregon Health &~
## 2 Portland State U.  U.C., Berkeley             U. of Washington U. of Washington
## 3 Harvard U.         U. of Oregon               U. of Chicago    Washington U. (~
## 4 U. of Washington   U. of Washington           Stanford U.      UC., San Fransi~
## 5 Columbia U.        New York U.                U. of Oregon     Stanford U.
## 6 U of Pennsylvania. U. of Chicago              Harvard U.       Harvard 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?

```
print(head(second_table))
```

```
## # A tibble: 6 x 4
##   MBAs               JDs                        PhDs             MDs
##   <chr>              <chr>                      <chr>            <chr>
## 1 U. of Chicago      Lewis & Clark  Law School U.C., Berkeley   Oregon Health &~
## 2 Portland State U.  U.C., Berkeley             U. of Washington U. of Washington
## 3 Harvard U.         U. of Oregon               U. of Chicago    Washington U. (~
## 4 U. of Washington   U. of Washington           Stanford U.      UC., San Fransi~
## 5 Columbia U.        New York U.                U. of Oregon     Stanford U.
## 6 U of Pennsylvania. U. of Chicago              Harvard U.       Harvard U..
```

The table is not in a tidy format since the data is spread across four different columns, each representing a different type of graduate program (MBAs, JDs, PhDs, MDs). In a tidy dataset, each row should represent one observation, and each observation should have its own row and column. In this case, the table should ideally have one column for the type of graduate program and one column for the name of the school, with each row representing a single instance of an alumni attending a graduate school.

---

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

```
library(tidyr)
library(dplyr)

# Assuming `second_table` is the one we need to tidy up
tidy_table <- second_table %>%
  pivot_longer(
    cols = everything(),
    names_to = "Degree",
    values_to = "School"
  ) %>%
  drop_na() # Remove NA values that may arise from reshaping

# Glimpse the resulting data frame to check its structure
tidy_table
```

```
## # A tibble: 44 x 2
##    Degree School
##    <chr>  <chr>
##  1 MBAs   U. of Chicago
##  2 JDs    Lewis & Clark  Law School
##  3 PhDs   U.C., Berkeley
##  4 MDs    Oregon Health & Sci Univ.
##  5 MBAs   Portland State U.
##  6 JDs    U.C., Berkeley
##  7 PhDs   U. of Washington
##  8 MDs    U. of Washington
##  9 MBAs   Harvard U.
## 10 JDs    U. of Oregon
## # i 34 more rows
```

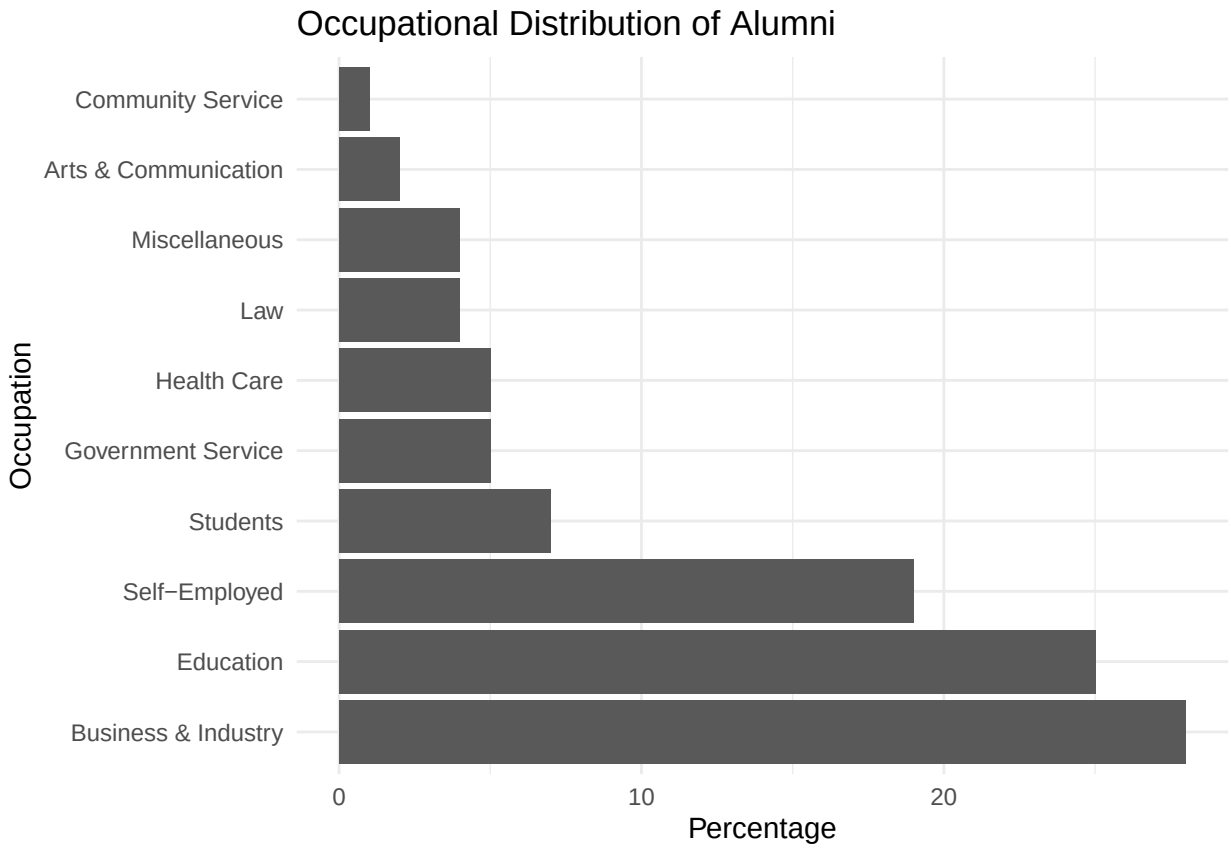
---

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)

```
print(head(first_table))
```

```
## # A tibble: 6 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%
```

```r
library(tibble)
library(dplyr)
library(ggplot2)
library(readr) # Load the readr package

first_table_df <- as_tibble(first_table)

# Assuming first_table_df is already a data frame with the correct columns
first_table_df <- first_table_df %>%
  mutate(Percentage = parse_number(X2))

# Now, create a bar plot with ggplot
ggplot(first_table_df, aes(x = reorder(X1, -Percentage), y = Percentage)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(x = "Occupation", y = "Percentage", title = "Occupational Distribution of Alumni") +
  coord_flip() # Flipping coordinates for horizontal bars
```



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.

```
# Hint
colnames(___) <- c("name 1", "name 2", ...)
```

- Remove any extraneous rows.

```
# Hint
filter(row_number() ...)
```

- 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.

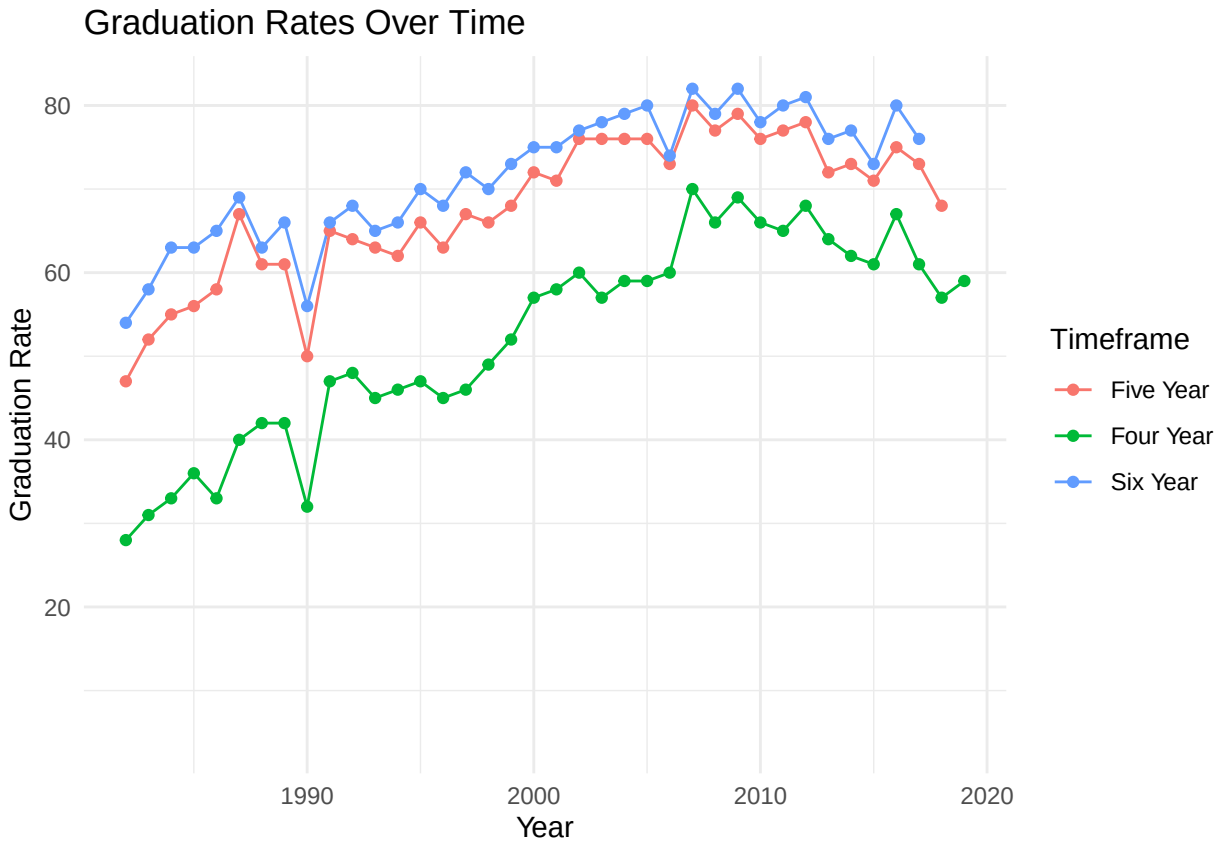
```
library(rvest)
library(dplyr)

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

grad_rates_page <- read_html(grad_rates_url)

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

colnames(grad_rates_table) <- c("Year", "Cohort", "Four Year", "Five Year", "Six Year")

# removes any extraneous rows that don't contain the data of interest
grad_rates_table <- grad_rates_table %>%
  filter(!is.na(Year) & Year != "" & !grepl("Note", Year)) # Adjust the condition based on the actual c

# convert percentage columns to numeric
grad_rates_table <- grad_rates_table %>%
  mutate_at(vars(`Four Year`, `Five Year`, `Six Year`), ~readr::parse_number(as.character(.)))

glimpse(grad_rates_table)
```

```
## Rows: 39
## Columns: 5
## $ Year        <chr> "First-year students who entered fall of...", "2019", "201~
## $ Cohort      <chr> "Number in Cohort", "393", "361", "411", "353", "418", "34~
## $ `Four Year` <dbl> 4, 59, 57, 61, 67, 61, 62, 64, 68, 65, 66, 69, 66, 70, 60,~
## $ `Five Year` <dbl> 5, NA, 68, 73, 75, 71, 73, 72, 78, 77, 76, 79, 77, 80, 73,~
## $ `Six Year`  <dbl> 6, NA, NA, 76, 80, 73, 77, 76, 81, 80, 78, 82, 79, 82, 74,~
```

---

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

```r
library(ggplot2)

grad_rates_table$Year <- as.numeric(grad_rates_table$Year)

grad_rates_long <- grad_rates_table %>%
  gather(key = "Timeframe", value = "Rate", `Four Year`, `Five Year`, `Six Year`)

grad_rates_long$Rate <- as.numeric(grad_rates_long$Rate)

ggplot(grad_rates_long, aes(x = Year, y = Rate, color = Timeframe, group = Timeframe)) +
  geom_line() +
  geom_point() +
  labs(title = "Graduation Rates Over Time", x = "Year", y = "Graduation Rate") +
  theme_minimal()
```



The four-year graduation rate shows an overall upward trend from the early 1990s, reaching a plateau in the early 2000s. The five-year rate appears to be the highest among the three, indicating that a significant portion of students graduate within five years rather than four.
The six-year rate is generally above the four-year rate, suggesting that most students who do not graduate within four years tend to do so by the sixth year.
There are points where the four and six-year rates converge, which might suggest particular years when students were either accelerating their studies or taking longer, possibly due to external factors.