Lab 5

Kiana Fields

Math 241, Week 6

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

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
# 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:

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

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?

```r
stations <- ncdc_stations(datasetid = "GHCND",
                          locationid = "FIPS:41051")

mult_stations <- stations$data
```

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
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"
```

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

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

precip_se_dpx is list

```
class(precip_se_pdx)
```

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

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

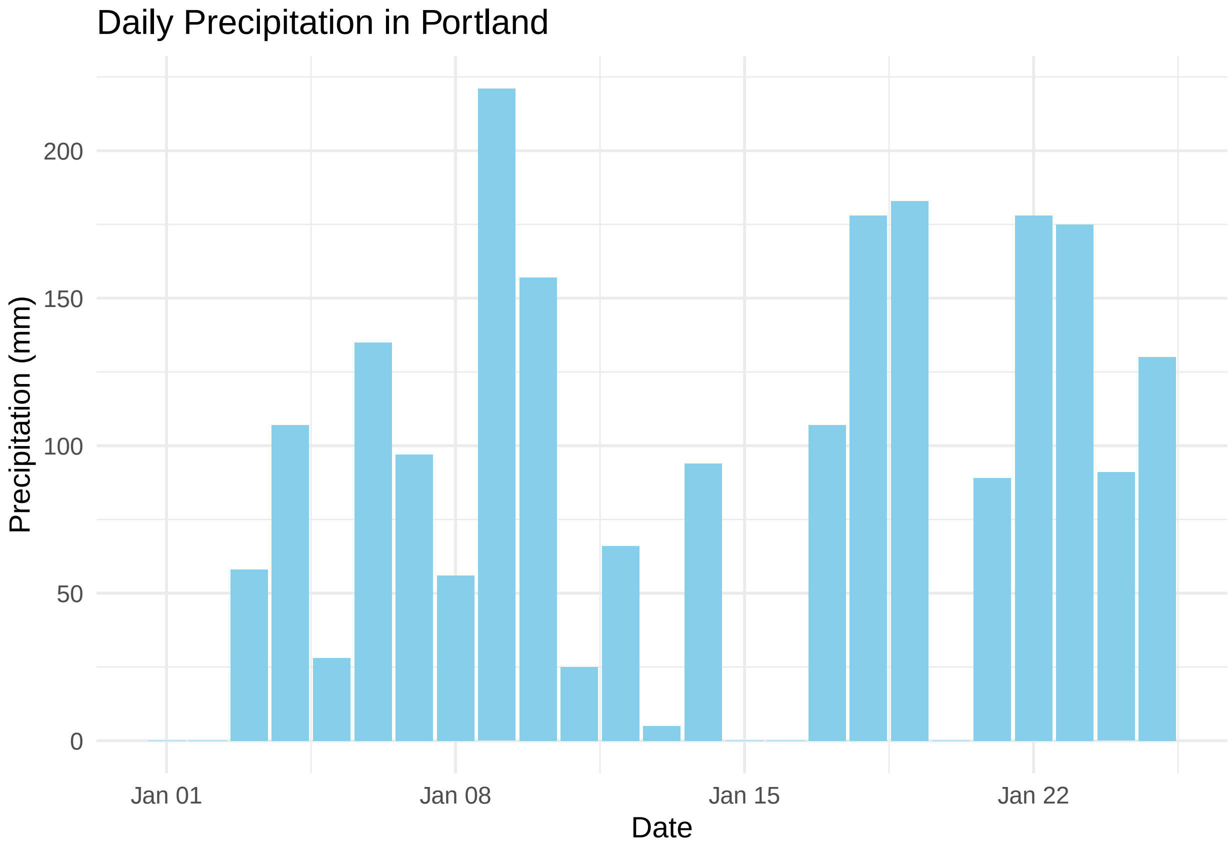
e. Use `ymd_hms()` in the package `lubridate` to wrangle the date column into the correct format.

```r
library(lubridate)

precip_se_pdx_data$date <- ymd_hms(precip_se_pdx_data$date)

head(precip_se_pdx_data)
```

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

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 = "skyblue") +
  labs(x = "Date", y = "Precipitation (mm)", title = "Daily Precipitation in Portland") +
  theme_minimal()
```



```
num_days_without_rain <- sum(precip_se_pdx_data$value == 0)
num_days_without_rain
```

```
## [1] 5
```

That is not true! We had 5 days without rain.

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_pdx_years <- ncdc(datasetid = "GHCND",
#                       stationid = "GHCND:US1ORMT0006",
#                       startdate = "2020-01-01",
#                       enddate = "2024-01-31",
#                       datatypeid = "PRCP")
#precip_se_pdx_years_data <- precip_se_pdx_years$data
#precip_se_pdx_years_data$date <- ymd_hms(precip_se_pdx_years_data$date)
#precip_januarys <- precip_se_pdx_years_data %>%
#  filter(month(date) == 1)

#library(ggplot2)
#ggplot(precip_januarys, aes(x = date, y = value, color = as.factor(year(date)))) +
#  geom_line() +
#  labs(x = "Date", y = "Precipitation (mm)", title = "January Precipitation in Portland (Last 4 Years)
#  theme_minimal() +
#  scale_color_manual(values = c("#B370F2", "#70D0F2", "#F2C470", "#F270C4"))

#ik this doesn't work but i feel like it should ? </3
```

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.

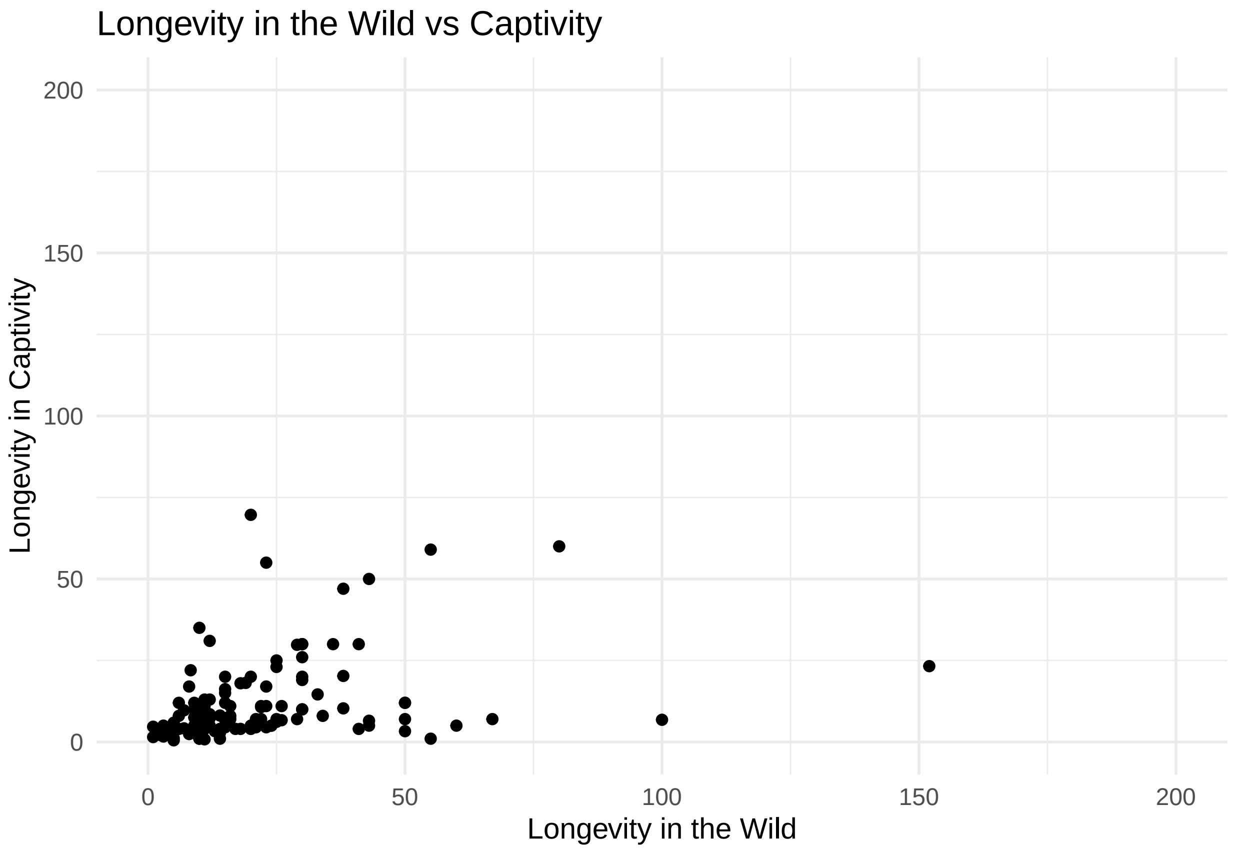
API Wrapper Suggestions for Problem 2

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

```
#install.packages("rfishbase")
library(rfishbase)
library(dplyr)
```

```r
library(ggplot2)

fish_df <- species()

filtered_fish_df <- fish_df %>%
  filter(!is.na(Length))

x_range <- c(0, 200)

ggplot(filtered_fish_df, aes(x = LongevityWild, y = LongevityCaptive)) +
  geom_point() +
  labs(x = "Longevity in the Wild",
       y = "Longevity in Captivity",
       title = "Longevity in the Wild vs Captivity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  coord_cartesian(xlim = x_range,
                  ylim = x_range)
```



Fish tend to live longer in the wild than in captivity. Additional analysis could reveal what other factors contribute to this observation (if it varies by genus, habitat type, etc)

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"

webpage <- read_html(url)

tables <- html_table(webpage)
```

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(tables[[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 ~
```

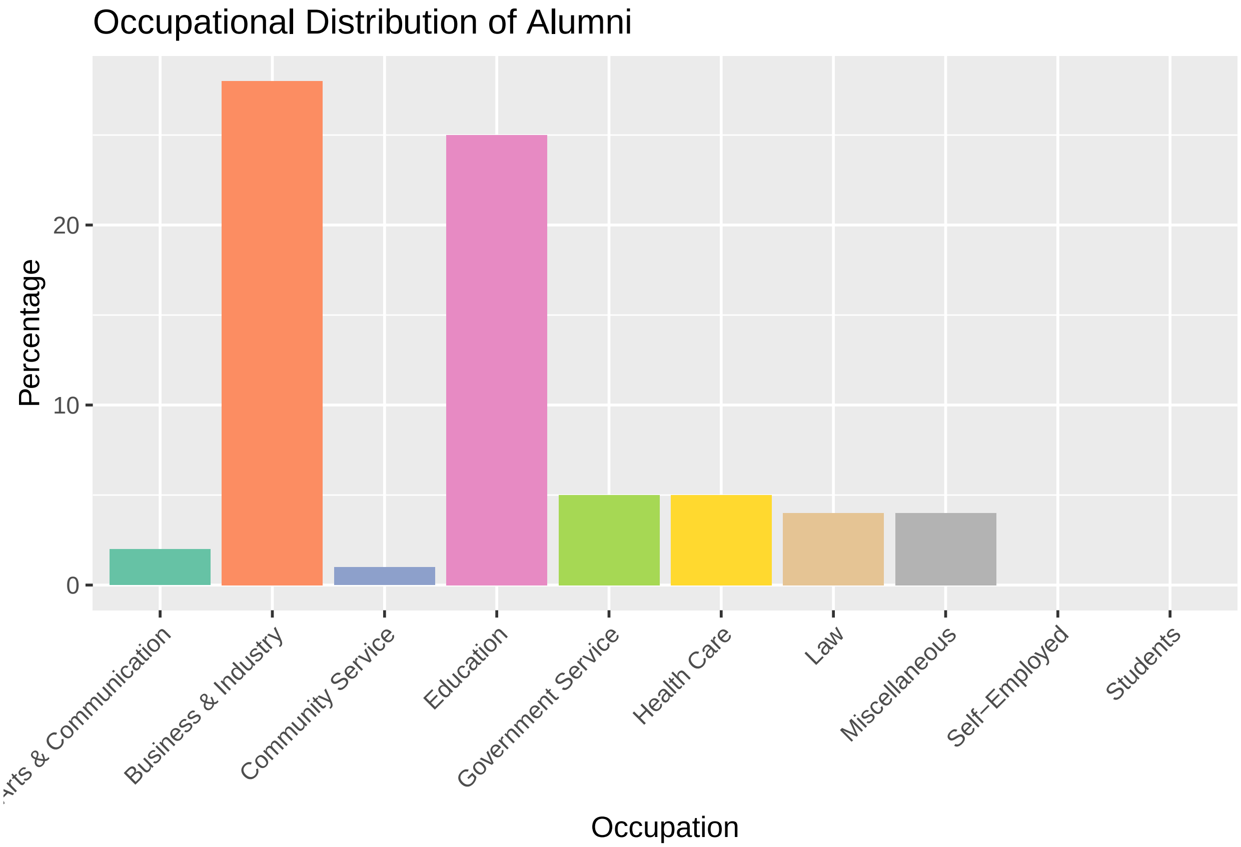
c. Wrangle the data into a tidy format. Glimpse the resulting data frame.

```
grad_schools <- tables[[2]]

grad_schools_tidy <- grad_schools %>%
  pivot_longer(cols = c(MBAs, JDs, PhDs, MDs), names_to = "degree_type", values_to = "college")

glimpse(grad_schools_tidy)
```

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

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
library(dplyr)

occ_data <- tables[[1]]

occ_data <- occ_data %>%
  mutate(perc_occ = parse_number(X2))

names(occ_data)[1] <- "occ"


ggplot(occ_data, aes(x = occ, y = perc_occ, fill = occ)) +
  geom_bar(stat = "identity") +
  labs(x = "Occupation", y = "Percentage", title = "Occupational Distribution of Alumni") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set2") +
  theme(legend.position = "none")
```



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.

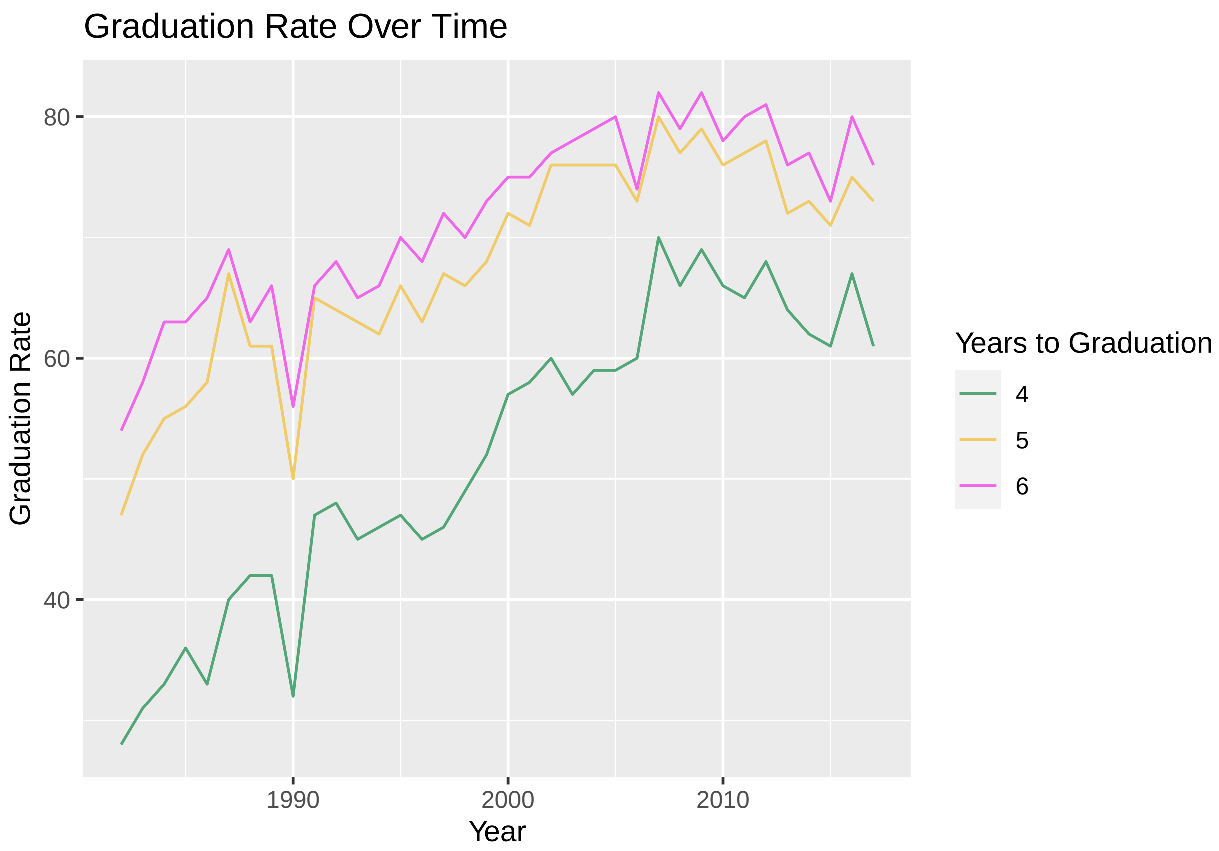
```r
# Hint
filter(row_number() ...)
```

- 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.

```r
url2 <- "https://www.reed.edu/ir/gradrateshist.html"
webpage2 <- read_html(url2)
tables2 <- html_table(webpage2)
grad_rate <- tables2[[1]]
grad_rate <- grad_rate[-1, ]
grad_rate <- grad_rate[-1, ]
grad_rate <- grad_rate[-1, ]

colnames(grad_rate) <- c("startyr", "numCohort", "four_yrs", "five_yrs", "six_yrs")

grad_rate <- grad_rate %>%
  mutate(four_yrs_pc = parse_number(four_yrs),
         five_yrs_pc = parse_number(five_yrs),
         six_yrs_pc = parse_number(six_yrs))
grad_rate$startyr <- as.numeric(grad_rate$startyr)
grad_rate$numCohort <- as.numeric(grad_rate$numCohort)
grad_rate_piv <- grad_rate %>%
  pivot_longer(cols = c(four_yrs_pc, five_yrs_pc, six_yrs_pc),
               names_to = "grad_yrs",
               values_to = "grad_rate")
grad_rate_piv <- grad_rate_piv %>%
  mutate(grad_yrs = case_when(
    grad_yrs == "four_yrs_pc" ~ 4,
    grad_yrs == "five_yrs_pc" ~ 5,
    grad_yrs == "six_yrs_pc" ~ 6))
grad_rate_piv <- grad_rate_piv %>%
  select(-c("four_yrs", "five_yrs", "six_yrs"))
```

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

```r
ggplot(grad_rate_piv, aes(x = startyr, y = grad_rate, color = factor(grad_yrs))) +
  geom_line() +
  labs(x = "Year", y = "Graduation Rate", title = "Graduation Rate Over Time") +
  scale_color_manual(values = c("#53a677", "#f0cb67", "#f067e9"), name = "Years to Graduation")
```



As the number of years till graduation increases, so does graduation rate. Overall, the graduation rate has been increasing over time. I would be interested to look into what happened in 1990 that dropped those rates.