Lab 5

Cameron Adams

Math 241, Week 6

```
# Put all necessary libraries here
library(tidyverse)
library(rnoaa)
library(rvest)
library(httr)
library(lubridate)
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 = "IGFTMJiYFRcmoLFPAvUiycwHFSzkdVlh")
```

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?

```
library(rnoaa)

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

# mult_stations <- stations$data
```

There are 24 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. change start data to 2024-01-01 end date end of the month. GHCND:US1ORMT0006

This was working fine but now im getting this error when I try to knit:

Quitting from lines 78-92 [unnamed-chunk-5] (lab05.Rmd) Error in `getOption()`: ! need an API key for NOAA data Backtrace: 1. rnoaa::ncdc_datatypes(datasetid = “GHCND”, stationid = “GHCND:US1ORMT0006”) 2. rnoaa:::check_key(token) 3. base::getOption(“noaakey”, stop(“need an API key for NOAA data”)) Execution halted

And since it is 2 am im just trying to submit

Quitting from lines 78-92 [unnamed-chunk-5] (lab05.Rmd) Error in `getOption()`: ! need an API key for NOAA data Backtrace: 1. rnoaa::ncdc_datatypes(datasetid = “GHCND”, stationid = “GHCND:US1ORMT0006”) 2. rnoaa:::check_key(token) 3. base::getOption(“noaakey”, stop(“need an API key for NOAA data”)) Execution halted

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



#precip_se_pdx <- ncdc(datasetid = "GHCND",
  #                    stationid = "GHCND:US1ORMT0006",
  #                    datatypeid = "PRCP",
  #                    startdate = "2024-01-01",
  #                    enddate = "2024-01-31",
  #                    var = c("date", "datatype", "value"))
```

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`.

```
#class(precip_se_pdx)

#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.

```
# Convert the date column to 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?

```
#ggplot(precip_se_pdx_data, aes(x = date, y = value)) +
 # geom_bar(stat = "identity", fill = "blue") +
 # labs(x = "Date", y = "Precipitation (mm)", title = "Precipitation in Portland (Jan 2024)")
```

From this plot we can see that it actually didn't rain on at least 3 different days, maybe 5 in total.

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?

```
#start_dates <- c("2021-01-01", "2022-01-01", "2023-01-01", "2024-01-01")
#end_dates <- c("2021-01-31", "2022-01-31", "2023-01-31", "2024-01-31")
```

```
#precip_data <- lapply(1:4, function(i) {
#  ncdc(datasetid = "GHCND",
 #      stationid = "GHCND:US1ORMT0006",
 #      datatypeid = "PRCP",
 #      startdate = start_dates[i],
   ##    enddate = end_dates[i])$data
#})

#all_precip_data <- do.call(rbind, precip_data)

#all_precip_data$date <- as.Date(all_precip_data$date)

#ggplot(all_precip_data, aes(x = date, y = value, color = factor(year(date)))) +
#  geom_line() +
#  labs(x = "Date", y = "Precipitation (mm)", title = "Precipitation in Portland (January, 2021-2024)")
#  facet_grid(year(date) ~ ., scales = "free_y") +
#  scale_color_manual(values = c("blue", "red", "green", "orange"))
```

Its not a very good graph but it seems that the weather has been getting worse each year with more rainfall in 2024 than every year before and 2022 and 2021 seming not too bad

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

###I am sorry I didn't really get to do this part rip

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

```
fish_data <- species()
str(fish_data)
```

```
## tibble [35,135 x 101] (S3: tbl_df/tbl/data.frame)
##  $ SpecCode            : int [1:35135] 24523 65802 24524 67300 67609 268 14737 10768 6652 6517 ...
##  $ Genus               : chr [1:35135] "Aborichthys" "Aborichthys" "Aborichthys" "Aborichthys" ...
##  $ SpeciesRefNo        : int [1:35135] 39226 95217 4832 95164 95217 59043 9743 9743 7247 7247 ...
##  $ Author              : chr [1:35135] "Chaudhuri, 1913" "Sen, 2009" "Barman, 1985" "Arunachalam, Ra
##  $ FBname              : chr [1:35135] NA NA NA NA ...
##  $ PicPreferredName    : chr [1:35135] "Abkem_u1.jpg" NA NA NA ...
##  $ PicPreferredNameM   : chr [1:35135] NA NA NA NA ...
##  $ PicPreferredNameF   : chr [1:35135] NA NA NA NA ...
##  $ PicPreferredNameJ   : chr [1:35135] NA NA NA NA ...
##  $ FamCode             : int [1:35135] 692 692 692 692 692 756 559 559 350 350 ...
##  $ Subfamily           : chr [1:35135] NA NA NA NA ...
##  $ GenCode             : int [1:35135] 784 784 784 784 784 50 3231 3231 7803 7803 ...
##  $ SubGenCode          : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ BodyShapeI          : chr [1:35135] "elongated" "Elongated" "Elongated" "elongated" ...
##  $ Source              : chr [1:35135] "O" "O" "O" "O" ...
##  $ AuthorRef           : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ Remark              : chr [1:35135] NA NA NA NA ...
##  $ TaxIssue            : int [1:35135] 0 0 0 NA NA 0 0 0 0 0 ...
##  $ Fresh               : int [1:35135] 1 1 1 1 1 1 1 1 0 0 ...
##  $ Brack               : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ Saltwater           : int [1:35135] 0 0 0 0 0 0 0 0 1 1 ...
##  $ DemersPelag         : chr [1:35135] "demersal" "demersal" "demersal" "demersal" ...
##  $ AirBreathing        : chr [1:35135] "WaterAssumed" "WaterAssumed" "WaterAssumed" "WaterAssumed" .
##  $ AirBreathingRef     : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ AnaCat              : chr [1:35135] NA NA NA NA ...
##  $ MigratRef           : int [1:35135] NA NA NA NA NA 51243 NA NA NA NA ...
##  $ DepthRangeShallow   : int [1:35135] NA NA NA NA NA 1 NA NA 1 1 ...
##  $ DepthRangeDeep      : int [1:35135] NA NA NA NA NA NA NA NA 50 6 ...
##  $ DepthRangeRef       : int [1:35135] NA NA NA NA NA 9696 NA NA 9710 7247 ...
##  $ DepthRangeComShallow: int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ DepthRangeComDeep   : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ DepthComRef         : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ LongevityWild       : num [1:35135] NA NA NA NA NA 23 NA NA NA NA ...
##  $ LongevityWildRef    : int [1:35135] NA NA NA NA NA 796 NA NA NA NA ...
##  $ LongevityCaptive    : num [1:35135] NA NA NA NA NA 17 NA NA NA NA ...
##  $ LongevityCapRef     : int [1:35135] NA NA NA NA NA 72462 NA NA NA NA ...
##  $ Vulnerability       : num [1:35135] 10 10 10 10 10 ...
##  $ VulnerabilityClimate: num [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ Length              : num [1:35135] 8.1 NA 10.9 7.75 6.8 ...
##  $ LTypeMaxM           : chr [1:35135] "SL" NA "SL" "SL" ...
##  $ LengthFemale        : num [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ LTypeMaxF           : chr [1:35135] NA NA NA NA ...
##  $ MaxLengthRef        : int [1:35135] 39226 NA 95217 95164 95217 6114 96636 7020 9710 9710 ...
##  $ CommonLength        : num [1:35135] NA NA NA NA NA 25 NA NA NA NA ...
##  $ LTypeComM           : chr [1:35135] NA NA NA NA ...
##  $ CommonLengthF       : num [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ LTypeComF           : chr [1:35135] NA NA NA NA ...
##  $ CommonLengthRef     : int [1:35135] NA NA NA NA NA 3561 NA NA NA NA ...
##  $ Weight              : num [1:35135] NA NA NA NA NA 6010 151 NA NA NA ...
##  $ WeightFemale        : num [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ MaxWeightRef        : int [1:35135] NA NA NA NA NA 4699 96636 NA NA NA ...
##  $ Pic                 : chr [1:35135] NA NA NA NA ...
##  $ PictureFemale       : chr [1:35135] NA NA NA NA ...
##  $ LarvaPic            : chr [1:35135] NA NA NA NA ...
##  $ EggPic              : chr [1:35135] NA NA NA NA ...
##  $ ImportanceRef       : int [1:35135] 4832 NA 4832 NA NA 4931 NA 42843 NA NA ...
##  $ Importance          : chr [1:35135] "of no interest" NA "of no interest" NA ...
##  $ PriceCateg          : chr [1:35135] "unknown" NA "unknown" NA ...
##  $ PriceReliability    : chr [1:35135] NA NA NA NA ...
##  $ Remarks7            : chr [1:35135] NA NA NA NA ...
##  $ LandingStatistics   : chr [1:35135] NA NA NA NA ...
##  $ Landings            : chr [1:35135] NA NA NA NA ...
##  $ MainCatchingMethod  : chr [1:35135] NA NA NA NA ...
##  $ II                  : chr [1:35135] NA NA NA NA ...
##  $ MSeines             : int [1:35135] 0 0 0 0 0 0 0 0 0 0 ...
##  $ MGillnets           : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ MCastnets           : int [1:35135] 0 0 0 0 0 0 0 0 0 0 ...
##  $ MTraps              : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ MSpears             : int [1:35135] 0 0 0 0 0 0 0 0 0 0 ...
##  $ MTrawls             : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ MDredges            : int [1:35135] 0 0 0 0 0 0 0 0 0 0 ...
##  $ MLiftnets           : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ MHooksLines         : int [1:35135] 0 0 0 0 0 1 0 0 1 0 ...
##  $ MOther              : int [1:35135] 0 0 0 0 0 0 0 0 1 0 ...
##  $ UsedforAquaculture  : chr [1:35135] "never/rarely" "never/rarely" "never/rarely" "never/rarely" .
##  $ LifeCycle           : chr [1:35135] NA NA NA NA ...
##  $ AquacultureRef      : int [1:35135] NA NA NA NA NA 12108 NA NA NA NA ...
##  $ UsedasBait          : chr [1:35135] "never/rarely" "never/rarely" "never/rarely" "never/rarely" .
##  $ BaitRef             : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ Aquarium            : chr [1:35135] "never/rarely" "never/rarely" "never/rarely" "never/rarely" .
##  $ AquariumFishII      : chr [1:35135] NA NA NA NA ...
##  $ AquariumRef         : int [1:35135] NA NA NA NA NA 274 NA 7020 5358 NA ...
##  $ GameFish            : int [1:35135] 0 0 0 0 0 1 0 0 0 0 ...
##  $ GameRef             : int [1:35135] NA NA NA NA NA 4699 NA NA NA NA ...
##  $ Dangerous           : chr [1:35135] "harmless" "harmless" "harmless" "harmless" ...
##  $ DangerousRef        : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ Electrogenic        : chr [1:35135] "no special ability" "no special ability" "no special ability
##  $ ElectroRef          : int [1:35135] NA NA NA NA NA NA NA NA NA NA ...
##  $ Complete            : chr [1:35135] NA NA NA NA ...
##  $ GoogleImage         : int [1:35135] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Comments            : chr [1:35135] "Occurs in streams with pebbly bottom (Ref. 41236)." NA "Occu
##  $ Profile             : chr [1:35135] NA NA NA NA ...
##  $ PD50                : num [1:35135] 0.504 0.504 0.504 0.504 0.504 ...
##  $ Emblematic          : int [1:35135] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Entered             : int [1:35135] 10 10 10 10 10 2 113 10 10 14 ...
##  $ DateEntered         : POSIXct[1:35135], format: "1996-07-18 00:00:00" "2010-10-05 00:00:00" ...
##  $ Modified            : int [1:35135] 65 65 65 10 NA 2 65 65 1472 2 ...
##  $ DateModified        : POSIXct[1:35135], format: "2013-07-18 00:00:00" "2021-10-25 00:00:00" ...
##  $ Expert              : int [1:35135] 10 NA 10 NA NA 97 437 3 65 65 ...
##   [list output truncated]
```

```
max_length <- fish_data$max_length_cm

summary(max_length)
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

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"
page <- read_html(url)
```

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?

```
graduate_schools_table <- page %>% html_table() %>% .[[2]]
print(graduate_schools_table)
```

```
## # 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.

```
graduate_schools_tidy <- graduate_schools_table %>%
  as_tibble() %>%
  rename(
    School = 1,
    Frequency = 2
  )

head(graduate_schools_tidy)
```

```
## # A tibble: 6 x 4
##   School             Frequency                 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..
```

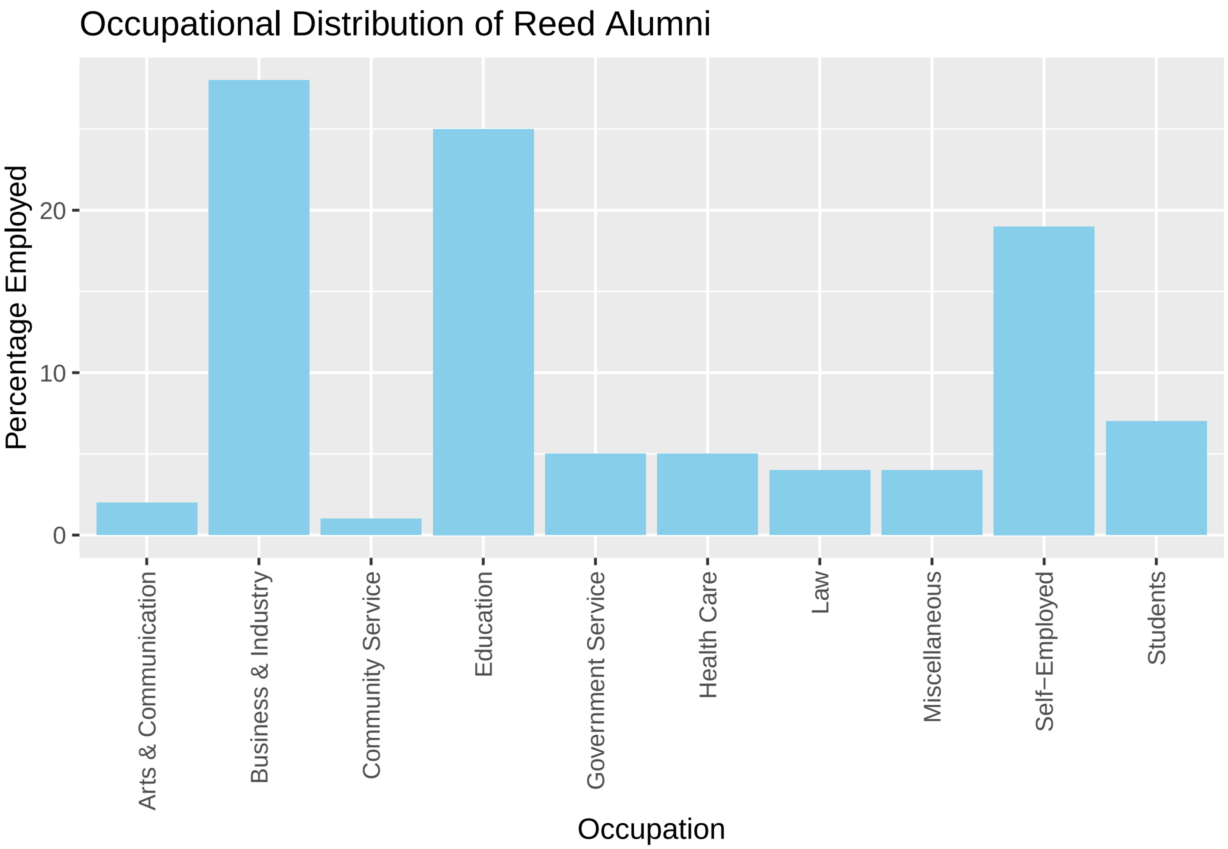
d. Now grab the “OCCUPATIONAL DISTRIBUTION OF ALUMNI” table and turn it into an appropriate graph. What conclusions can we draw from the graph?

```
occupational_distribution_table <- page %>% html_table() %>% .[[1]]

colnames(occupational_distribution_table) <- c("Occupation", "Percentage Employed")

occupational_distribution_tidy <- occupational_distribution_table %>%
  as_tibble() %>%
  mutate(
    Percentage = parse_number(`Percentage Employed`)
  ) %>%
  select(Occupation, Percentage)

ggplot(occupational_distribution_tidy, aes(x = Occupation, y = Percentage)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Occupational Distribution of Reed Alumni",
       x = "Occupation",
       y = "Percentage Employed") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



WE can see that business and indusrty, education and self-employed are the main occupations of reed alumni with very few going into community service

#got Errors here and below so not commplete

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.

```
# Define the URL for graduation rates data
grad_rates_url <- "https://www.reed.edu/ir/gradrateshist.html"

# Read the HTML content
page <- read_html(grad_rates_url)

# Extract the table
grad_rates_table <- page %>% html_table(fill = TRUE) %>% .[[1]]



print(grad_rates_table)
```

```
## # A tibble: 39 x 5
##    First-year students who ~1 `Number in Cohort` `Graduated in:` `Graduated in:`
##    <chr>                      <chr>              <chr>           <chr>
##  1 First-year students who e~ Number in Cohort   4 Years         "5 Years"
##  2 2019                       393                59%*            ""
##  3 2018                       361                57%             "68%*"
##  4 2017                       411                61%             "73%"
##  5 2016                       353                67%             "75%"
##  6 2015                       418                61%             "71%"
##  7 2014                       346                62%             "73%"
##  8 2013                       354                64%             "72%"
##  9 2012                       320                68%             "78%"
## 10 2011                       372                65%             "77%"
## # i 29 more rows
## # i abbreviated name: 1: `First-year students who entered fall of...`
## # i 1 more variable: `Graduated in:` <chr>
```

```
#grad_rates_clean <- grad_rates_table %>%
 #
#mutate(across(starts_with("Graduated"), ~ as.numeric(gsub("[^0-9.]", "", .)))) %>%  # Extract numeric
 # pivot_longer(cols = starts_with("Graduated"),
  #             names_to = "Years",
  #             values_to = "Graduation rate")

#glimpse(grad_rates_clean)
```

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

```
 #ggplot(grad_rates_clean, aes(x = `Entering class year`, y = `Graduation rate`, color = Years)) +
 # geom_line() +
 # labs(title = "Graduation Rates Over Time at Reed College",
 #      x = "Entering Class Year",
 #      y = "Graduation Rate",
 #      color = "Years to Graduation") +
 # theme_minimal()
```