DS5110 Homework 2

Ameya Santosh Gidh

2024-02-08

Part A

Problem 1

Overview of Netflix Dataset:

Summary:

- This dataset offers insights into TV shows and movies accessible on Netflix, capturing information up to the onset of 2021.
- Comprising 7787 records and 12 variables, the dataset provides a comprehensive view.
- The dataset's columns include:

  Show__ID - a distinct identifier for Netflix content

  type - categorization of shows as Movies or TV shows

  title - the name or title of the Netflix content

  director - the director's name for the show

  Cast - the acting cast involved in the show

  country - the country of origin for the content

  date__added - the date of release on Netflix

  release year - the year the content was released

  rating - the assigned rating for the content

  duration - the length of the movie or show

  genre - the genre of the content

  Description - a brief summary of the content
- The dataset contains a limited number of movies from the year 2021.
- The dataset's source is attributed to: https://www.kaggle.com/datasets/senapatirajesh/netflix-tv-shows-and-movies?select=NetFlix.csv

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
# install.packages("readr")
# install.packages("dplyr")
library(dplyr)
library("readr")
```

Importing Data sets and loading libraries

```r
#  Citing source of dataset: https://www.kaggle.com/datasets/senapatirajesh/netflix-tv-shows-and-movies
netflix_dataset <- read_csv("NetFlix.csv", na = c(""))
```

```
## Rows: 7787 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): show_id, type, title, director, cast, country, date_added, rating,...
## dbl  (2): release_year, duration
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
tibble(netflix_dataset)
```

```
## # A tibble: 7,787 x 12
##    show_id type    title   director cast  country date_added release_year rating
##    <chr>   <chr>   <chr>   <chr>    <chr> <chr>   <chr>             <dbl> <chr>
##  1 s1      TV Show 3%      <NA>     João~ Brazil  14-Aug-20          2020 TV-MA
##  2 s10     Movie   1920    Vikram ~ Rajn~ India   15-Dec-17          2008 TV-MA
##  3 s100    Movie   3 Hero~ Iman Br~ Reza~ Indone~ 05-Jan-19          2016 TV-PG
##  4 s1000   Movie   Blue M~ Lev L. ~ Alan~ United~ 01-Mar-16          2016 R
##  5 s1001   TV Show Blue P~ <NA>     Davi~ United~ 03-Dec-18          2017 TV-G
##  6 s1002   Movie   Blue R~ Jeremy ~ Maco~ United~ 25-Feb-19          2013 R
##  7 s1003   Movie   Blue S~ Les May~ Mart~ German~ 01-Jan-21          1999 PG-13
##  8 s1004   Movie   Blue V~ Derek C~ Ryan~ United~ 05-Jul-18          2010 R
##  9 s1005   Movie   BluffM~ Rohan S~ Abhi~ India   08-Jan-21          2005 TV-14
## 10 s1006   Movie   Blurre~ Barry A~ <NA>  Canada  31-Dec-17          2017 TV-MA
## # i 7,777 more rows
## # i 3 more variables: duration <dbl>, genres <chr>, description <chr>
```

Pre-process the data

Dataset Cleaning / Preparation Steps

- **Handled mixed date formats in the date_added column**
- **Filtered and retained entries in the date_added column with dates prior to the current date**
- **Replaced durations falling between 1-10 minutes with the average duration value**
- **Substituted any missing values in the rating column with the designation 'Unknown'**
- **Created a new column to extract the airing year information from the date_added column**

```r
library(dplyr)
library(lubridate)

# Convert the "date_added" column to a consistent format using lubridate
netflix_dataset <- netflix_dataset %>%
  mutate(date_added = lubridate::ymd(date_added))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `date_added = lubridate::ymd(date_added)`.
## Caused by warning:
## !  88 failed to parse.
```

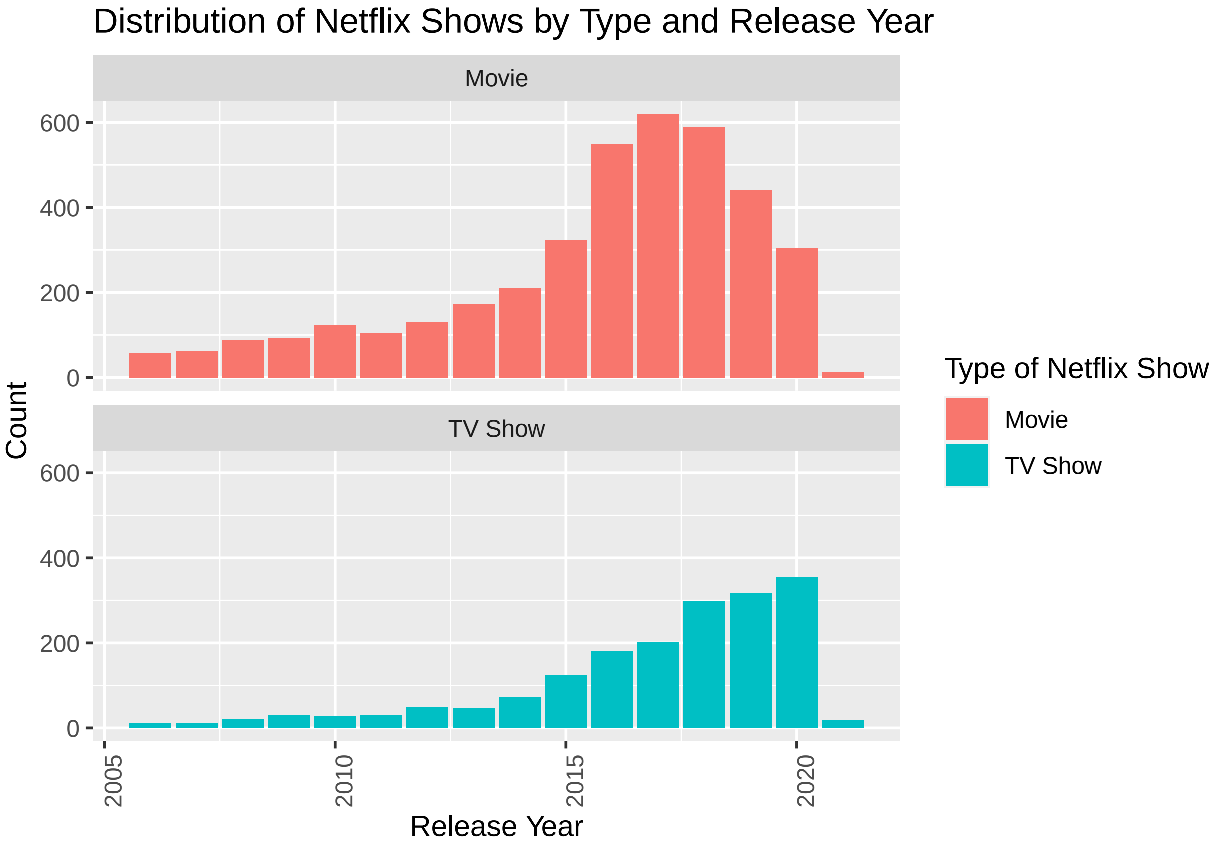
```r
# Filter out entries with future dates in the "date_added" column
netflix_dataset <- netflix_dataset %>%
  filter(date_added < Sys.Date())

# Replace extremely short durations (1-10 mins) with the mean duration
netflix_dataset$duration[netflix_dataset$duration < 10] <- as.integer(mean(netflix_dataset$duration))

# Replace NA values in the "rating" column with 'Unknown'
netflix_dataset$rating[is.na(netflix_dataset$rating)] <- 'Unknown'

# Create a new column named "aired_on_netflix_year" by extracting the year from the "date_added" column
netflix_dataset$aired_on_netflix_year <- as.double(str_sub(netflix_dataset$date_added, start = 0, end =

# Display the transformed dataframe, and observe the changes made in the "date_added" column
tibble(netflix_dataset)
```

```
## # A tibble: 6,317 x 13
##    show_id type    title   director cast  country date_added release_year rating
##    <chr>   <chr>   <chr>   <chr>    <chr> <chr>   <date>            <dbl> <chr>
##  1 s1      TV Show 3%      <NA>     João~ Brazil  2014-08-20         2020 TV-MA
##  2 s10     Movie   1920    Vikram ~ Rajn~ India   2015-12-17         2008 TV-MA
##  3 s100    Movie   3 Hero~ Iman Br~ Reza~ Indone~ 2005-01-19         2016 TV-PG
##  4 s1000   Movie   Blue M~ Lev L. ~ Alan~ United~ 2001-03-16         2016 R
##  5 s1001   TV Show Blue P~ <NA>     Davi~ United~ 2003-12-18         2017 TV-G
##  6 s1003   Movie   Blue S~ Les May~ Mart~ German~ 2001-01-21         1999 PG-13
##  7 s1004   Movie   Blue V~ Derek C~ Ryan~ United~ 2005-07-18         2010 R
##  8 s1005   Movie   BluffM~ Rohan S~ Abhi~ India   2008-01-21         2005 TV-14
##  9 s1008   Movie   BNK48:~ Nawapol~ <NA>  Thaila~ 2001-03-19         2018 TV-14
## 10 s1009   Movie   Bo Bur~ Bo Burn~ Bo B~ United~ 2003-06-16         2016 TV-MA
## # i 6,307 more rows
## # i 4 more variables: duration <dbl>, genres <chr>, description <chr>,
## #   aired_on_netflix_year <dbl>
```

Problem 2

Visualizations

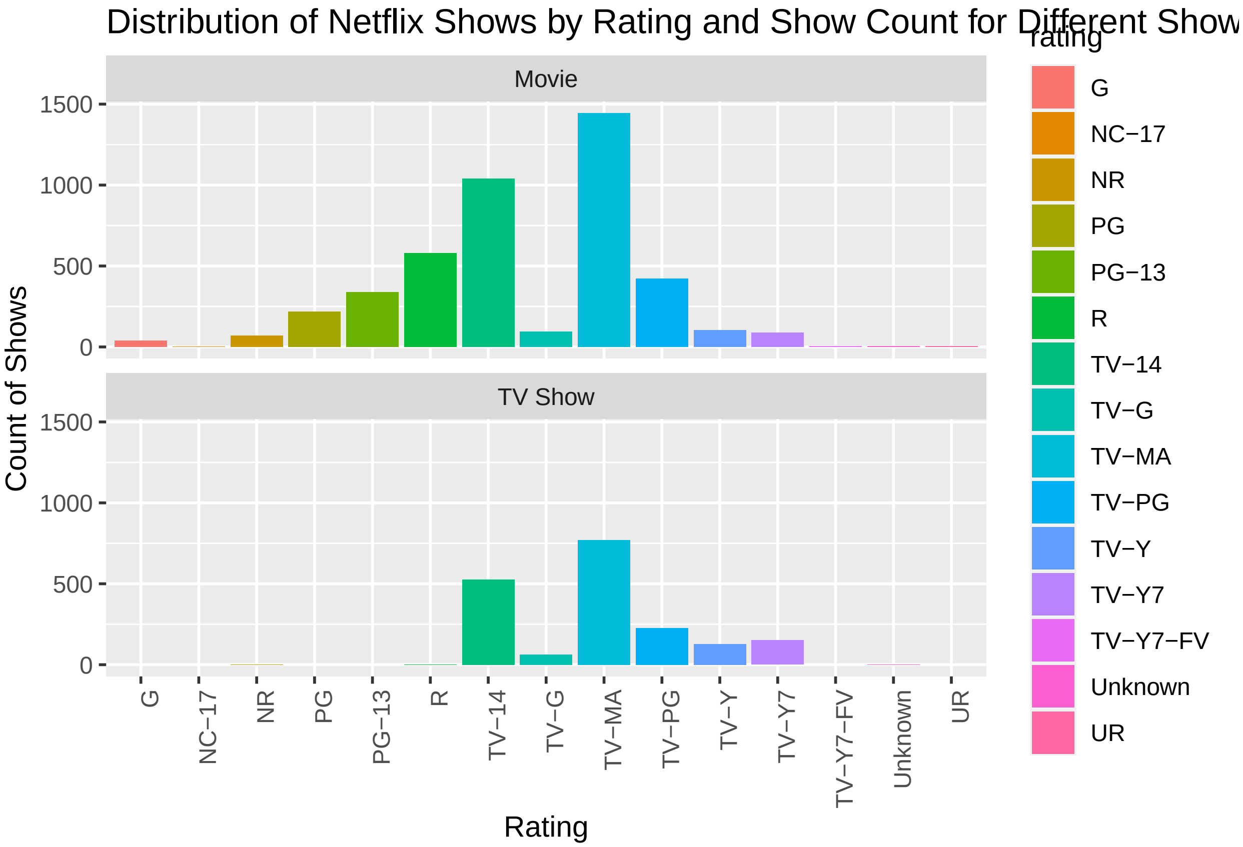
```r
library(ggplot2)
# Create a bar plot to display the distribution of Netflix shows by release year and type
netflix_plot <- ggplot(data = netflix_dataset[netflix_dataset$release_year > 2005, ],
                       aes(x = release_year, fill = type)) +
  geom_bar() +  # Add bars to represent the count
  facet_wrap(~type, ncol = 1) +  # Separate plots by show type
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # Rotate x-axis labels for readability
  labs(title = "Distribution of Netflix Shows by Type and Release Year",
       x = "Release Year",
       y = "Count",
       fill = "Type of Netflix Show")  # Set plot titles and labels

# Display the plot
print(netflix_plot)
```



Findings and Summary:

- The observations were made from the year 2005 onward.
- Netflix categorizes shows into two types: 1. Movies and 2. TV Shows.
- The provided bar plot illustrates the distribution of these two types of shows from 2005 to 2021.
- Notably, the count of TV Shows exhibits a consistent increase over the specified years.
- In contrast, the count of movies shows a rise until 2017, followed by a decline.
- Due to the dataset's limitation, encompassing data until the beginning of 2021, the count of movies for 2021 is relatively low for both Movies and TV Shows.
- The peak count for movies occurred in 2017, while for TV Shows, the highest count was in 2020.
- Across all years, except for 2020, the count of movies consistently exceeded the count of TV Shows.

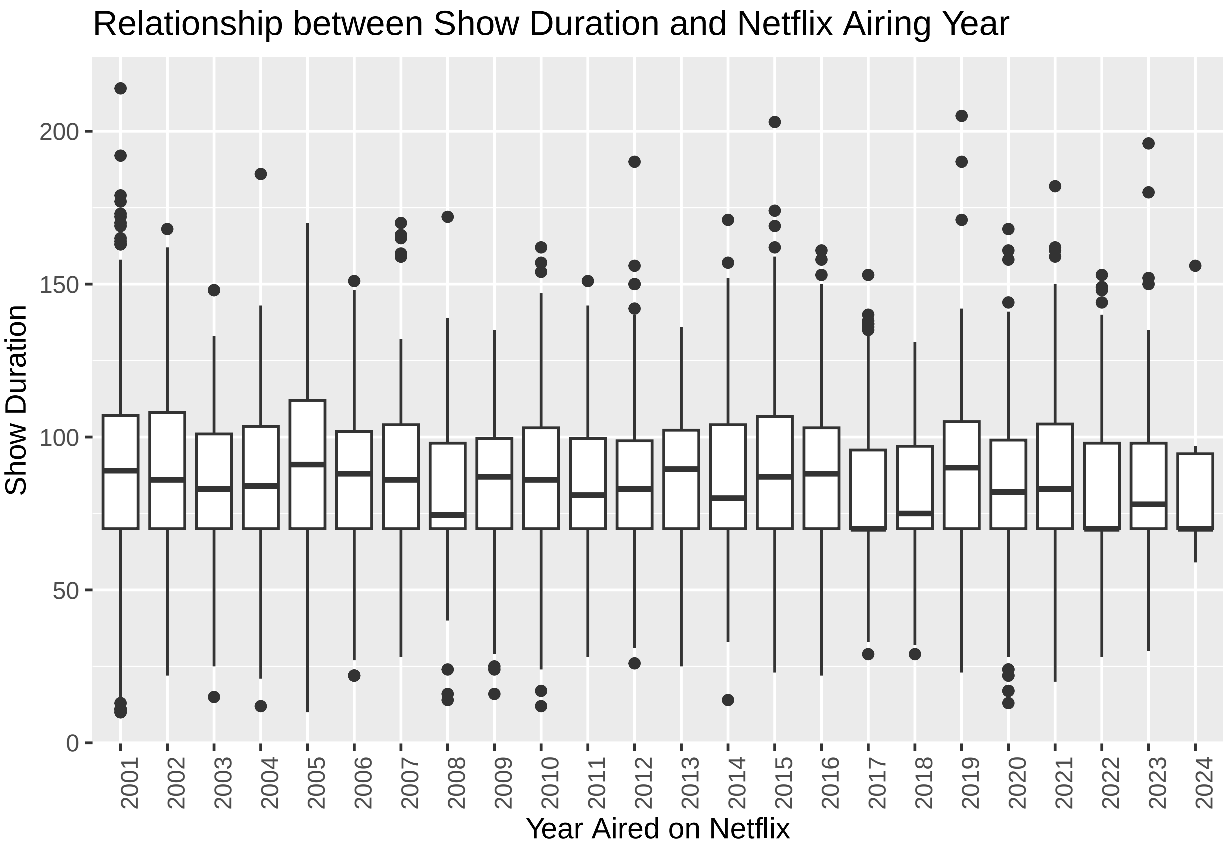
```
# Create a bar plot to visualize the distribution of Netflix shows based on their ratings and show coun
ggplot(data = netflix_dataset, aes(x = rating, fill = rating)) +
  geom_bar() +  # Add bars to the plot
  facet_wrap(~type, ncol = 1) +  # Separate plots by show type
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # Rotate x-axis labels
  labs(title = "Distribution of Netflix Shows by Rating and Show Count for Different Show Types",
       x = "Rating",
       y = "Count of Shows")  # Set plot titles and labels
```



Findings and Conclusions:

- The majority of shows in the dataset, encompassing both Movies and TV Shows, fall under the **TV-MA rating** category.
- Movies tend to have a higher count for each rating compared to TV Shows.
- Notably, there are more TV-Y7-rated TV shows on Netflix than TV-Y7-rated movies.

```
# Create a boxplot to explore the relationship between the duration of Netflix shows
# and the year they were aired on Netflix, focusing on shows aired after 2005.
ggplot(data = netflix_dataset[which(netflix_dataset$release_year > 2005), ],
       aes(x = as.factor(aired_on_netflix_year), y = duration)) +
  geom_boxplot() +  # Add boxplot for visualizing the distribution
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +  # Rotate x-axis labels
  labs(title = "Relationship between Show Duration and Netflix Airing Year",
       x = "Year Aired on Netflix",
       y = "Show Duration")  # Set plot titles and labels
```



Findings and Conclusions:

- There is no evident correlation between the years and the duration of shows.
- The average duration of shows remains consistent, ranging between 100 and 75 minutes across all years.
- The year 2019 stands out with the highest average duration.

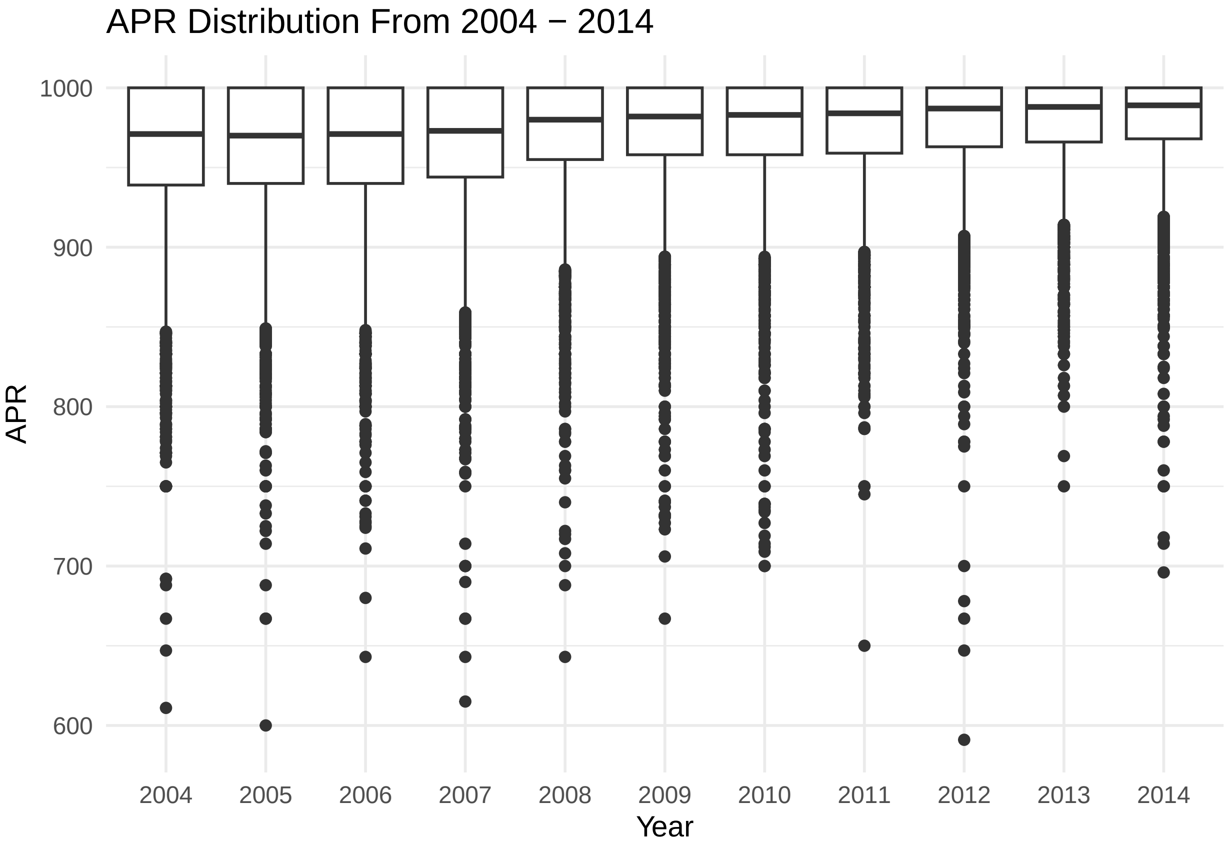
Part B

Problem 3

```r
# install.packages(c("readr", "tidyverse"))
library(tidyverse)
library(readr)
library(dplyr)
library(stringr)

data <- read.csv(file = "26801-0001-Data.tsv", sep = "\t", na = c(-99))

data %>%
  pivot_longer(cols = starts_with("APR_RATE_"), names_to = "YEAR", values_to = "APR") %>%
  select(SCL_UNITID, SCL_NAME, SPORT_CODE, SPORT_NAME, YEAR, APR) %>%
  mutate(YEAR = str_sub(YEAR, 10, 13)) -> data

print(data)
```

```
## # A tibble: 71,621 x 6
##    SCL_UNITID SCL_NAME               SPORT_CODE SPORT_NAME      YEAR    APR
##          <int> <chr>                        <int> <chr>          <chr> <int>
##  1      100654 Alabama A&M University          20 Women's Bowling 2014   1000
##  2      100654 Alabama A&M University          20 Women's Bowling 2013   1000
##  3      100654 Alabama A&M University          20 Women's Bowling 2012   1000
##  4      100654 Alabama A&M University          20 Women's Bowling 2011   1000
##  5      100654 Alabama A&M University          20 Women's Bowling 2010    950
##  6      100654 Alabama A&M University          20 Women's Bowling 2009   1000
##  7      100654 Alabama A&M University          20 Women's Bowling 2008   1000
##  8      100654 Alabama A&M University          20 Women's Bowling 2007    958
##  9      100654 Alabama A&M University          20 Women's Bowling 2006    875
## 10      100654 Alabama A&M University          20 Women's Bowling 2005   1000
## # i 71,611 more rows
```

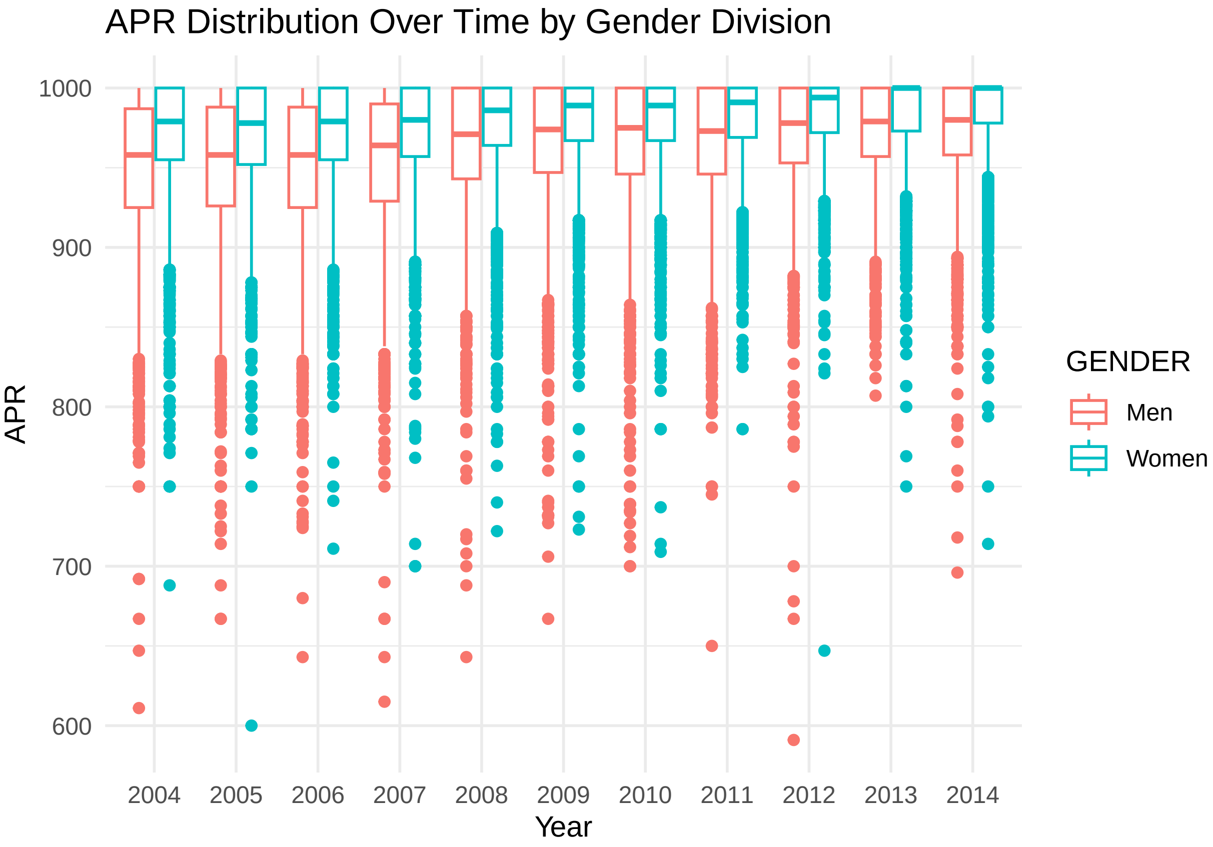
```
ggplot(data, aes(x = YEAR, y = APR)) +
  geom_boxplot(na.rm = TRUE) +
  ggtitle("APR Distribution From 2004 - 2014") +
  xlab("Year") +
  ylab("APR") +
  theme_minimal()
```



The average annual percentage rates (APRs) have continuously risen over time, as indicated by the plot. A direct positive correlation exists between APRs and the years, signifying an improvement in the academic progress of teams from 2004 to 2014.

Problem 4

```
data %>%
  filter(SPORT_CODE >= 1 & SPORT_CODE <= 37) %>%
  mutate(GENDER= ifelse(SPORT_CODE >= 1 & SPORT_CODE <= 18, "Men", "Women")) -> data

ggplot(data, aes(x = YEAR, y = APR, color=GENDER)) +
  geom_boxplot(na.rm=TRUE) +
  ggtitle("APR Distribution Over Time by Gender Division") +
  xlab("Year") +
  ylab("APR") +
  theme_minimal()
```



The mean APR for women consistently surpasses that of men, both at the individual and team levels, spanning the years from 2004 to 2014.

The visual representation indicates a recurring pattern where women's sports exhibit higher average APRs compared to men's sports.

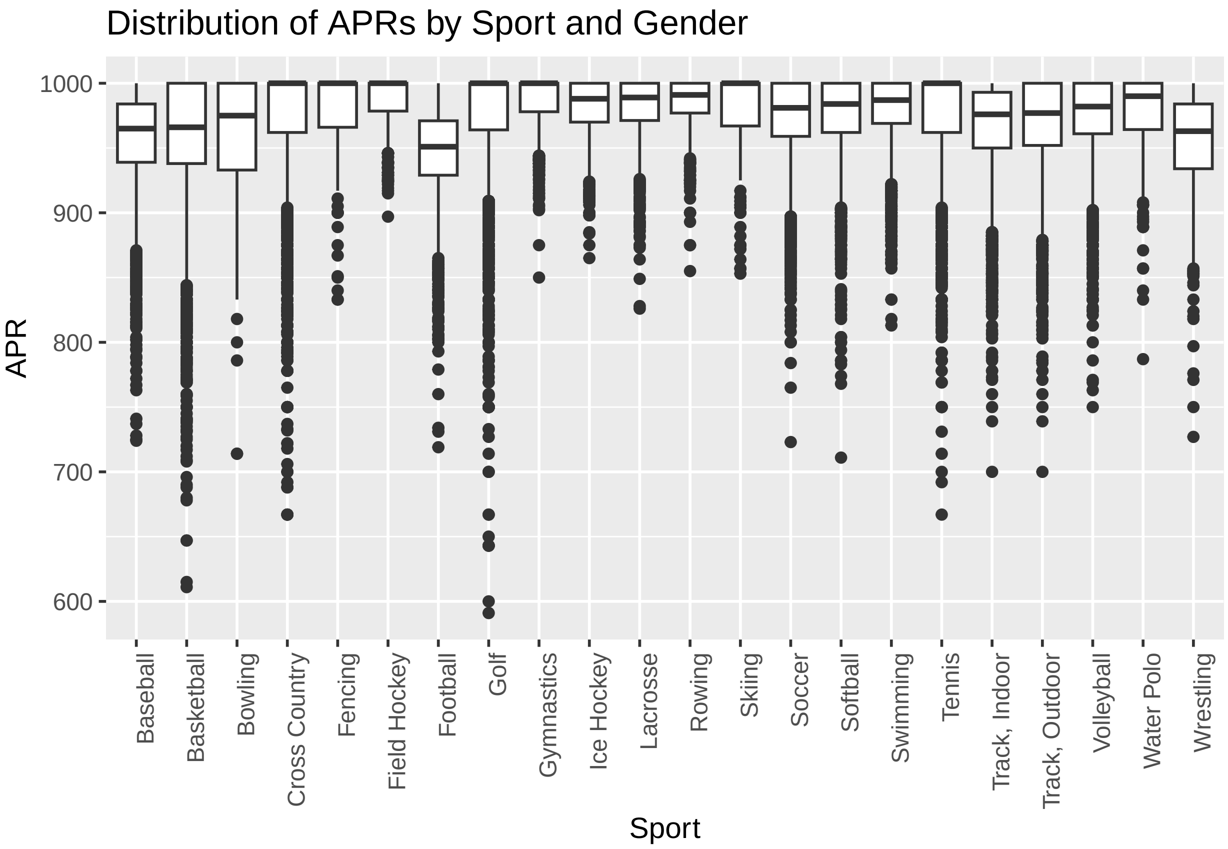
Throughout the period from 2004 to 2014, the average APR for women's sports consistently exceeds that of men's sports.

This analysis suggests that women's sports demonstrate superior academic progress compared to men's sports.

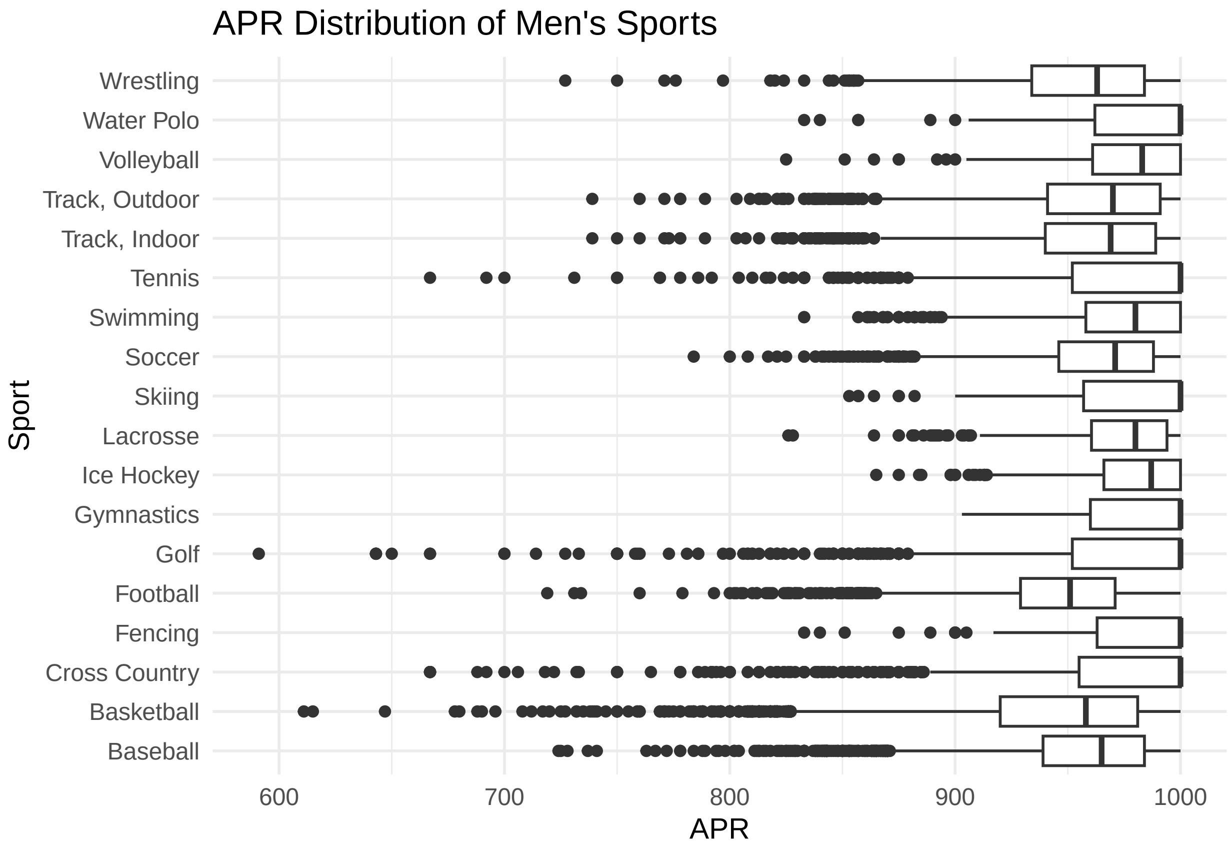
The trend reveals a continual increase in the average APR for women's sports over the years, while there is some irregular growth in the average APR for men's sports.

Problem 5

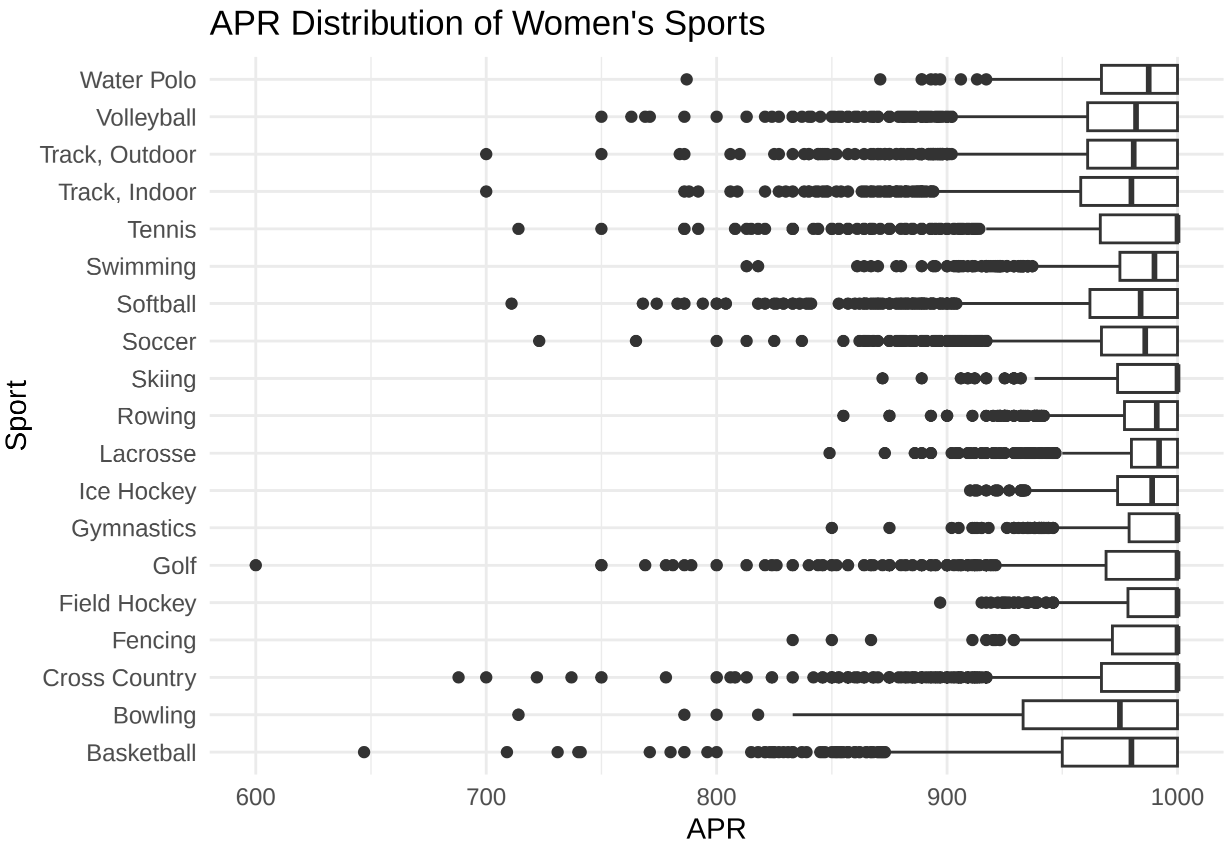
```
library(stringr)
data <- mutate(data,SPORT_NAME = str_remove(SPORT_NAME, "^Men's |^Women's "))
ggplot(data, aes(x = SPORT_NAME, y = APR)) +
  geom_boxplot(na.rm = TRUE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Distribution of APRs by Sport and Gender",
       x = "Sport",
       y = "APR",
       fill = "Gender")
```



```
data %>%
filter(GENDER == "Men") %>%
ggplot(aes(x = SPORT_NAME, y = APR)) +
geom_boxplot(na.rm = TRUE) +
ggtitle("APR Distribution of Men's Sports") +
xlab("Sport") +
ylab("APR") +
theme_minimal() +
coord_flip()
```



```
data %>%
filter(GENDER == "Women") %>%
ggplot(aes(x = SPORT_NAME, y = APR)) +
geom_boxplot(na.rm = TRUE) +
ggtitle("APR Distribution of Women's Sports") +
xlab("Sport") +
ylab("APR") +
theme_minimal() +
coord_flip()
```



Generally, men’s sports teams, including football, basketball, and baseball, exhibit lower APRs on average. In contrast, sports like fencing, skiing, and water polo tend to have higher APRs. Notably, men’s gymnastics consistently maintains a higher APR than other sports, without any outliers, while men’s football and basketball consistently show lower APRs.

As depicted in the visual representation above, men’s and women’s teams exhibit comparable APRs in the following sports: 1. Volleyball 2. Fencing 3. Golf 4. Gymnastics 5. Skiing 6. Tennis 7. Cross Country

The average APRs for these sports are closely aligned for both men and women. In contrast, there is a noticeable divergence in APRs between genders for the remaining sports. As illustrated in the visualization from Q4, women consistently have higher APRs than men across the majority of sports.