Assignment 2 - Tidying Data

Mansi Pravin Thanki

2023-02-07

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/r9/2cgj8871421bvklfhk05xfzc0000gn/T//RtmpHrNy0x/downloaded_packages
```

```
install.packages("lubridate", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/r9/2cgj8871421bvklfhk05xfzc0000gn/T//RtmpHrNy0x/downloaded_packages
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.7     v dplyr   1.1.0
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(readr)
library(dplyr)
```

About Netflix dataset:

Description:

- This dataset contains information about Netflix (a popular entertainment streaming service) TV Shows and Movies upto the beginning of Year 2021.
- The dataset contains 7787 observations of 12 variables
- Columns present in the dataset are as follows:

Show__ID - unique id of the netflix show

type - Netflix show can be 2 types -> Movie or TV show

title - Name or title of the Netflix show

director - Name of the director of the show

Cast - acting cast of the show

country - origin country of the Show

date__added - date on which the show was released on Netflix

release year - release year of the show

rating - rating of the show

duration - Length of the movie

genre - genre of the movie

Description - Summary of the movie

- Very limited movies from the year 2021 are present in the dataset
- Citing source of dataset: https://www.kaggle.com/datasets/senapatirajesh/netflix-tv-shows-and-movies?select=NetFlix.csv

Loading the dataset

```
#  Citing source of dataset: https://www.kaggle.com/datasets/senapatirajesh/netflix-tv-shows-and-movies
netflix_dataset <- read_csv("/Users/mansipravinthanki/Downloads/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.
```

```
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
## # ... with 7,777 more rows, and 3 more variables: duration <dbl>, genres <chr>,
## #   description <chr>
```

Tidying the dataset / Preprocessing steps

- **Resolved issues with mixed date format in date_added column**
- **Filtered out and retained the dates in date_added column that had date_added before current date**
- **Replaced Duration with 1-10 mins duration with mean value of the duration**
- **Replaced NA values in rating column with 'Unknown' rating**
- **Created new column to get the year in which the show was aired from date_added column**

```
# as you can see, certain dates are in character format "14-Aug-20" whereas a
# very few dates are in character  format "November 1, 2019"

# we can tidy this by using lubridate package and mutating the column date_added
netflix_dataset <- netflix_dataset%>%mutate(netflix_dataset, date_added = lubridate::ymd(date_added))

# one can see that some years in date_added belong to 2031, 2027, 2025 etc.
# The release dates for this movies is in the past years.
# So this does not make sense that date_added (date that the movie was added on Netflix)
# will ever have a date in future from today
# So we use filter and retain only the rows where date_added has a date before current date
netflix_dataset <- filter(netflix_dataset, netflix_dataset$date_added < Sys.Date())


# movies or TV shows cannot have 1-10 mins of duration. It seems like a bad data to me
# hence I have replaced those values with the mean of duration
netflix_dataset$duration[netflix_dataset$duration<10] <- as.integer(mean(netflix_dataset$duration))

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

# using str_sub, extracted the year and created a new column in dataframe -> aired_on_netflix_year
```

```
netflix_dataset$aired_on_netflix_year <- as.double(str_sub(netflix_dataset$date_added, start = 0, end =

# printing dataframe. Check date_added column to see the transformation
tibble(netflix_dataset)
```

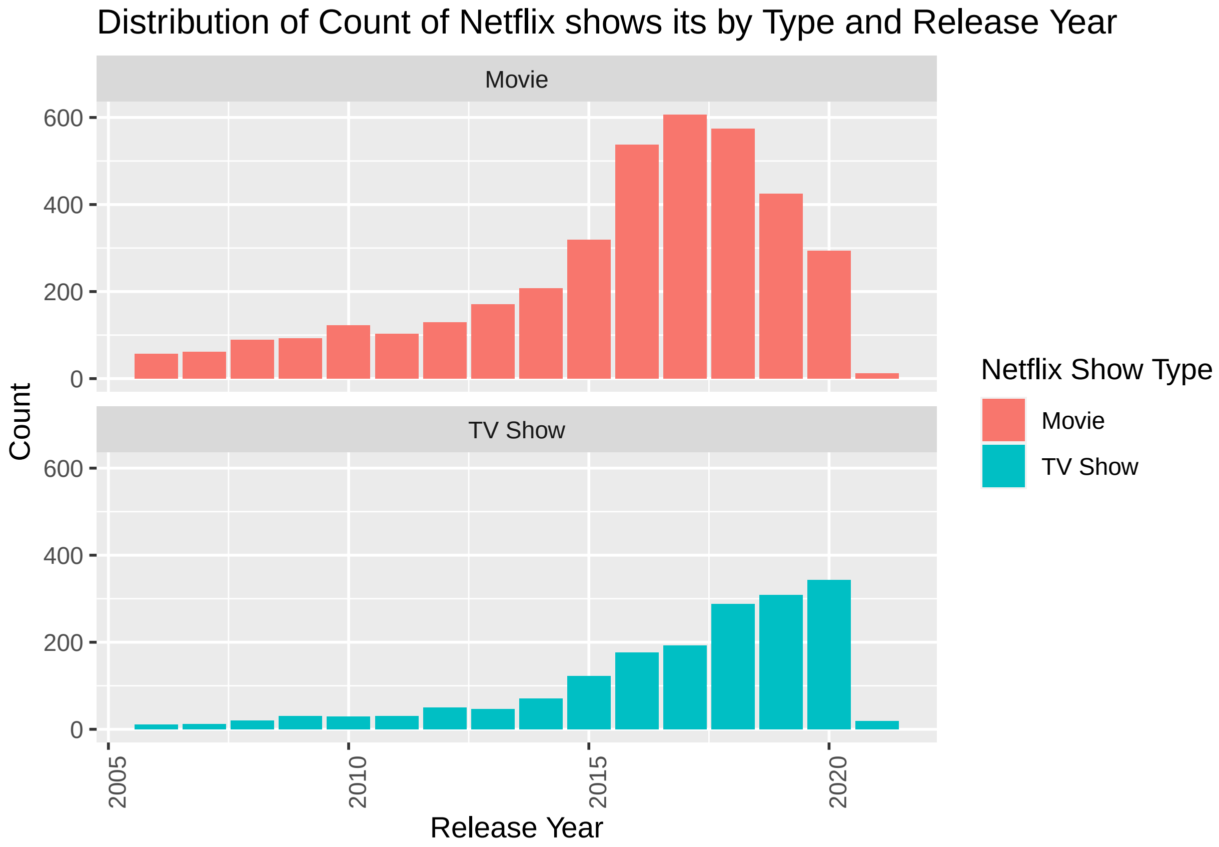
```
## # A tibble: 6,180 x 13
##    show_id type    title   director cast  country date_added release_year rating
##    <chr>   <chr>   <chr>   <chr>    <chr> <chr>   <date>            <dbl> <chr>
##  1 s1      TV Show 3%      <NA>     João~ Brazil  2014-08-20          2020 TV-MA
##  2 s10     Movie   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
## # ... with 6,170 more rows, and 4 more variables: duration <dbl>, genres <chr>,
## #   description <chr>, aired_on_netflix_year <dbl>
```

Problem 2

Visualization No: 1

Distribution of Count of Netflix shows its by Type and Release Year

```
library(ggplot2)

ggplot(netflix_dataset[which(netflix_dataset$release_year>2005),], aes( x= release_year, fill = type))
  geom_bar() +
  facet_wrap(~type, ncol=1) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Distribution of Count of Netflix shows its by Type and Release Year",
       x = "Release Year",
       y = "Count",
       fill = "Netflix Show Type")
```



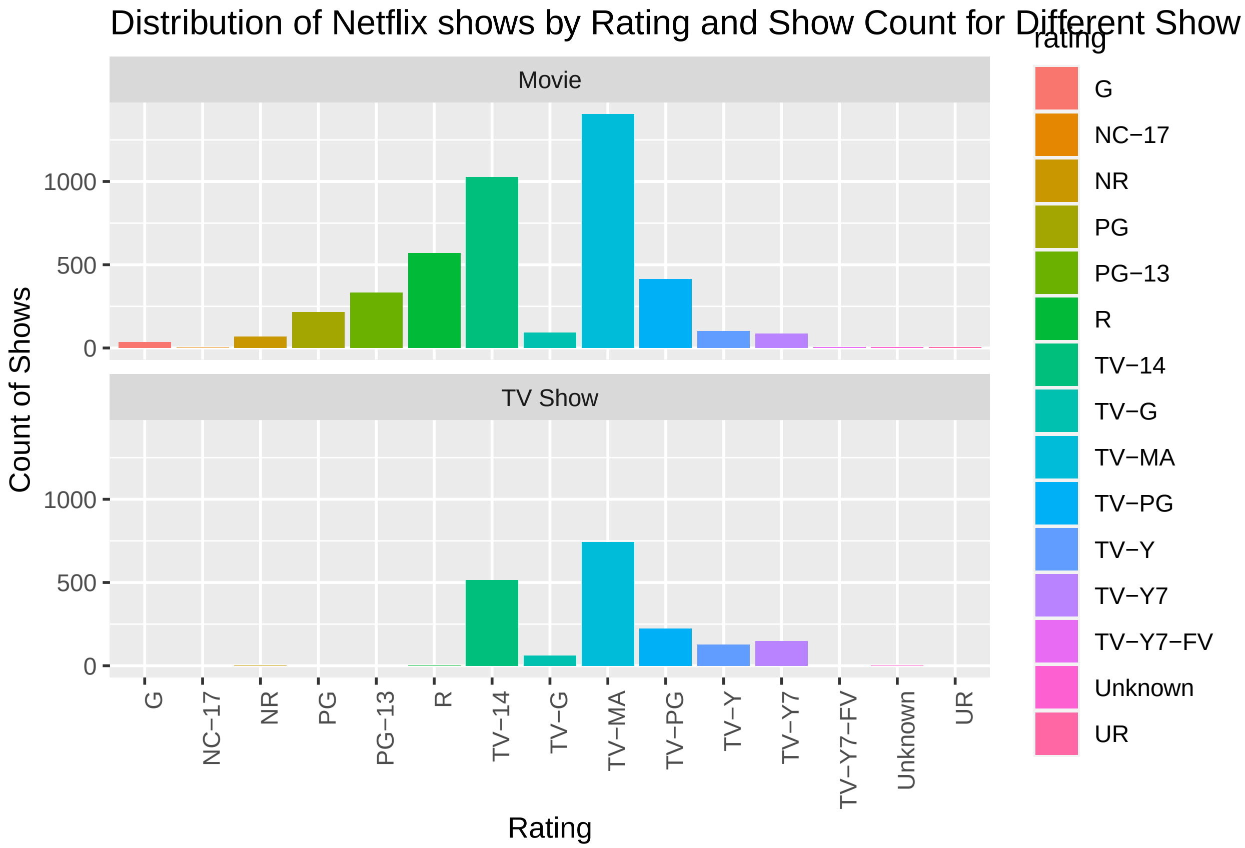
Observations and Conclusions:

- I have made the observations for the years 2005 onwards
- There are two types of Netflix shows -> 1. Movie 2. TV Show
- The above bar plot visualizes the count of the two types of shows across the years 2005-2021
- One can visualize that for TV Shows, the count increases over the years.
- For Movies, one can see that the count of movies is rising until the year 2017. After 2017, it sees a decline in the count.
- The dataset contains data about Netflix shows until beginning of 2021. And hence, it does not contain many movies of the year 2021, and hence the count for year 2021 is low for both Movies and TV Shows
- The highest count of Movies was in year 2017 and for TV Shows (~600 count), the highest count was in 2020 (~350 count)
- The count of Movies has always been higher than count of TV Shows across all the years except 2020.

Visualization No: 2

Distribution of Netflix shows by Rating and Show Count for Different Show Types

```
library(ggplot2)

ggplot(netflix_dataset, aes(x = rating, fill=rating)) +
  geom_bar() +
  facet_wrap(~ type, ncol = 1) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Distribution of Netflix shows by Rating and Show Count for Different Show Types",
       x = "Rating",
       y = "Count of Shows")
```



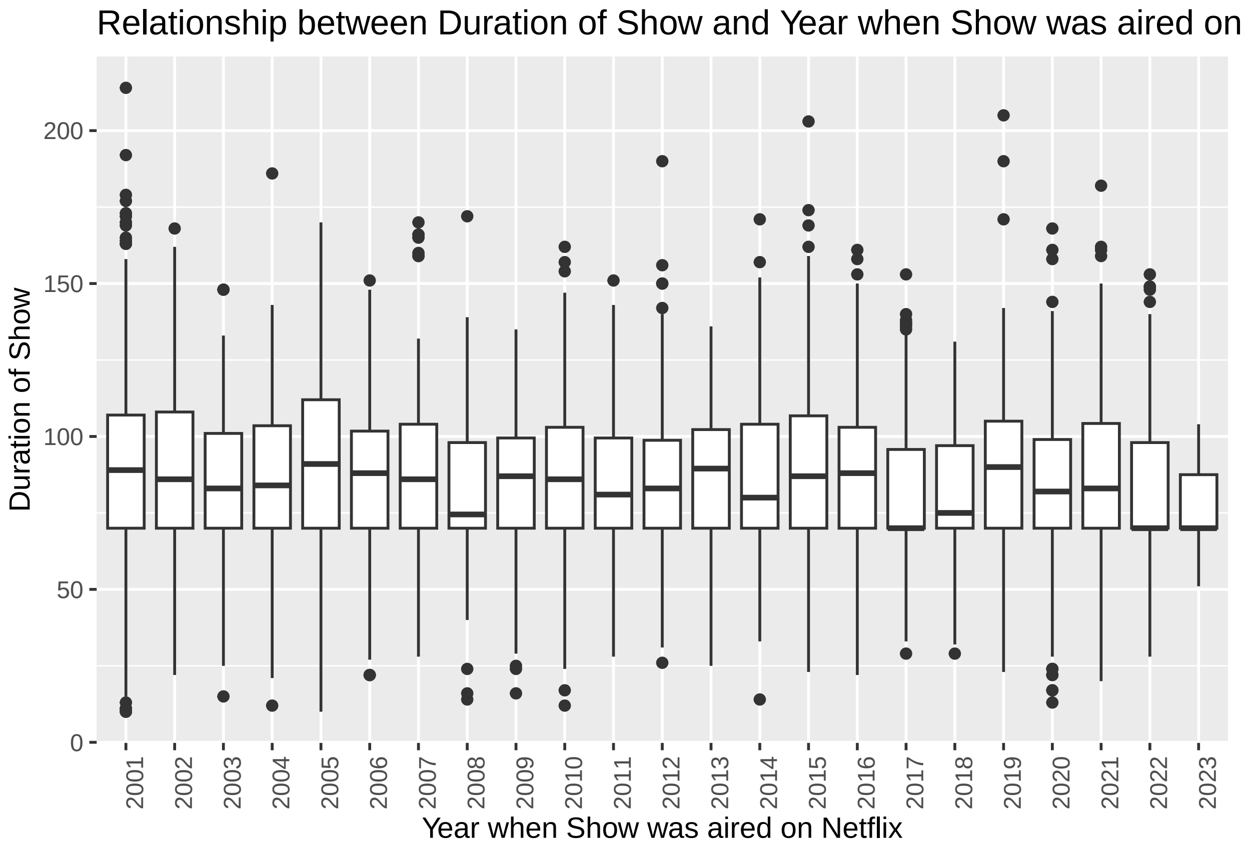
Observations and Conclusions:

- Maximum count of shows (both Movies and TV Shows) in the dataset belong to **TV-MA rating**
- The count of shows of each eating is higher for Movies than the TV Shows
- More TV-Y7 TV shows are on Netflix than TV-YZ movies

Visualization No: 3

Relationship between Duration of Show and Year when Show was aired on Netflix

```r
library(ggplot2)

ggplot(netflix_dataset[which(netflix_dataset$release_year>2005),], aes(x=as.factor(aired_on_netflix_yea
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title="Relationship between Duration of Show and Year when Show was aired on Netflix",
       x="Year when Show was aired on Netflix",
       y="Duration of Show")
```



Observations and Conclusions:

- It can be observed that there is no direct relationship between the Years and the Duration of the shows.
- The average duration of shows across all years range between 100-75
- Year 2019 has the highest average duration

# PART B:

```r
apr_dataset <- read_tsv("/Users/mansipravinthanki/Downloads/NCAA-D1-APR-2003-14/DS0001/26801-0001-Data.
```

```
## Rows: 6511 Columns: 76
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (4): SCL_NAME, SPORT_NAME, CONFNAME_14, D1_FB_CONF_14
## dbl (69): SCL_UNITID, SPORT_CODE, ACADEMIC_YEAR, SCL_DIV_14, SCL_SUB_14, SCL...
## lgl  (3): DATA_TAB_GENERALINFO, DATA_TAB_MULTIYRRATE, DATA_TAB_ANNUALRATE
##
## 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.
```

#Q3:

Create a tidy data frame that includes columns for: • School ID • School name • Sport code • Sport name • Year • APR All other columns can be discarded. Use your tidied dataset to visualize the distributions of APRs over time. How does the distribution of APRs change year-to-year from 2004 to 2014?

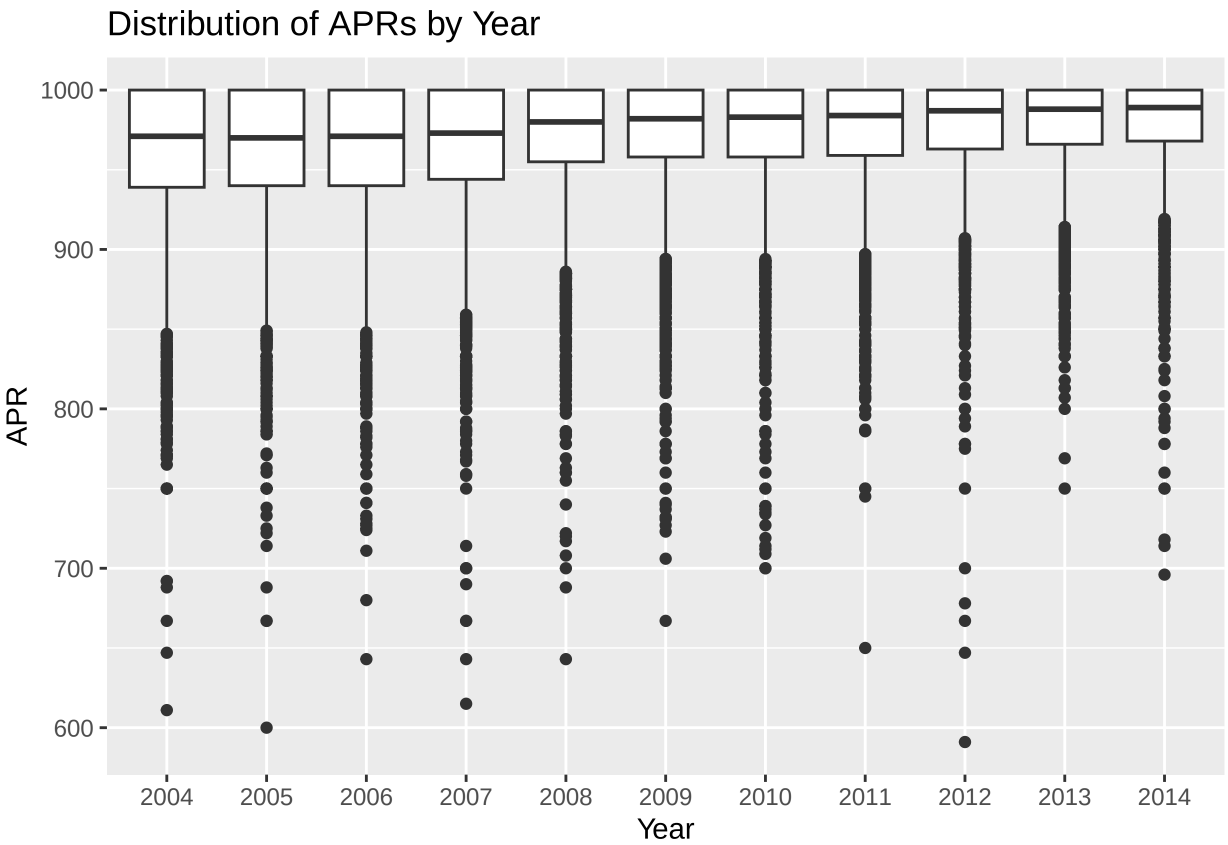
Solution:

```
# first  get all the columns that start with "APR_RATE_"
untidy_apr_columns_df <- apr_dataset %>% select(all_of(starts_with("APR_RATE_")))

# get the column names from the untidy_apr_columns_df dataframe
apr_columnNames <- sort(colnames(untidy_apr_columns_df))

# as seen, the observation of Years are scattered across the columns
#hence one can use pivot_longer to address this.
apr_dataset <-pivot_longer(apr_dataset, cols=apr_columnNames, names_to = "YEAR",
                           values_to = "APR")

# using str_sub to extract year from "APR_RATE_XXXX_1000"
apr_dataset$Year <- str_sub(apr_dataset$YEAR, start = -9, end = -6)

# filtering values to exclude values that are negative
apr_dataset <- filter(apr_dataset,APR>0)

# selecting the columns out of tidied dataset
apr_dataset <- select(apr_dataset, SCL_UNITID, SCL_NAME, SPORT_CODE, SPORT_NAME,
                      Year, APR)

# printing the dataframe
tibble(apr_dataset)
```

```
## # A tibble: 66,889 x 6
##    SCL_UNITID SCL_NAME               SPORT_CODE SPORT_NAME      Year    APR
##         <dbl> <chr>                       <dbl> <chr>           <chr> <dbl>
##  1     100654 Alabama A&M University         20 Women's Bowling 2004   1000
##  2     100654 Alabama A&M University         20 Women's Bowling 2005   1000
##  3     100654 Alabama A&M University         20 Women's Bowling 2006    875
##  4     100654 Alabama A&M University         20 Women's Bowling 2007    958
##  5     100654 Alabama A&M University         20 Women's Bowling 2008   1000
##  6     100654 Alabama A&M University         20 Women's Bowling 2009   1000
##  7     100654 Alabama A&M University         20 Women's Bowling 2010    950
##  8     100654 Alabama A&M University         20 Women's Bowling 2011   1000
##  9     100654 Alabama A&M University         20 Women's Bowling 2012   1000
## 10     100654 Alabama A&M University         20 Women's Bowling 2013   1000
## # ... with 66,879 more rows
```

```
library(ggplot2)
```

```
ggplot(apr_dataset, aes(x = Year, y = APR)) +
  geom_boxplot() +
  labs(title = "Distribution of APRs by Year",
       x = "Year",
       y = "APR")
```



- As seen in the boxplot above, the APRs are consistently increasing over the years

- There is a direct positively increasing relationship between APRs and the Years
- This indicates that the academic progress of the teams have increased over the span of years from 2004-2014
- The highest average APR is seen in 2014, and the lowest average is seen in 2004.

#Q4:

We would like to compare APRs between men's and women's sports. Transform your tidied dataset to remove mixed sports, and create a column indicating the gender division of each sport. (You may assume sport codes 1-18 are men's, and 19-37 are women's.) Visualize the distributions of APRs over time again, but broken down by gender division. How do the average APRs compare between men's and women's sports? Does this relationship hold true across each year from 2004 to 2014?

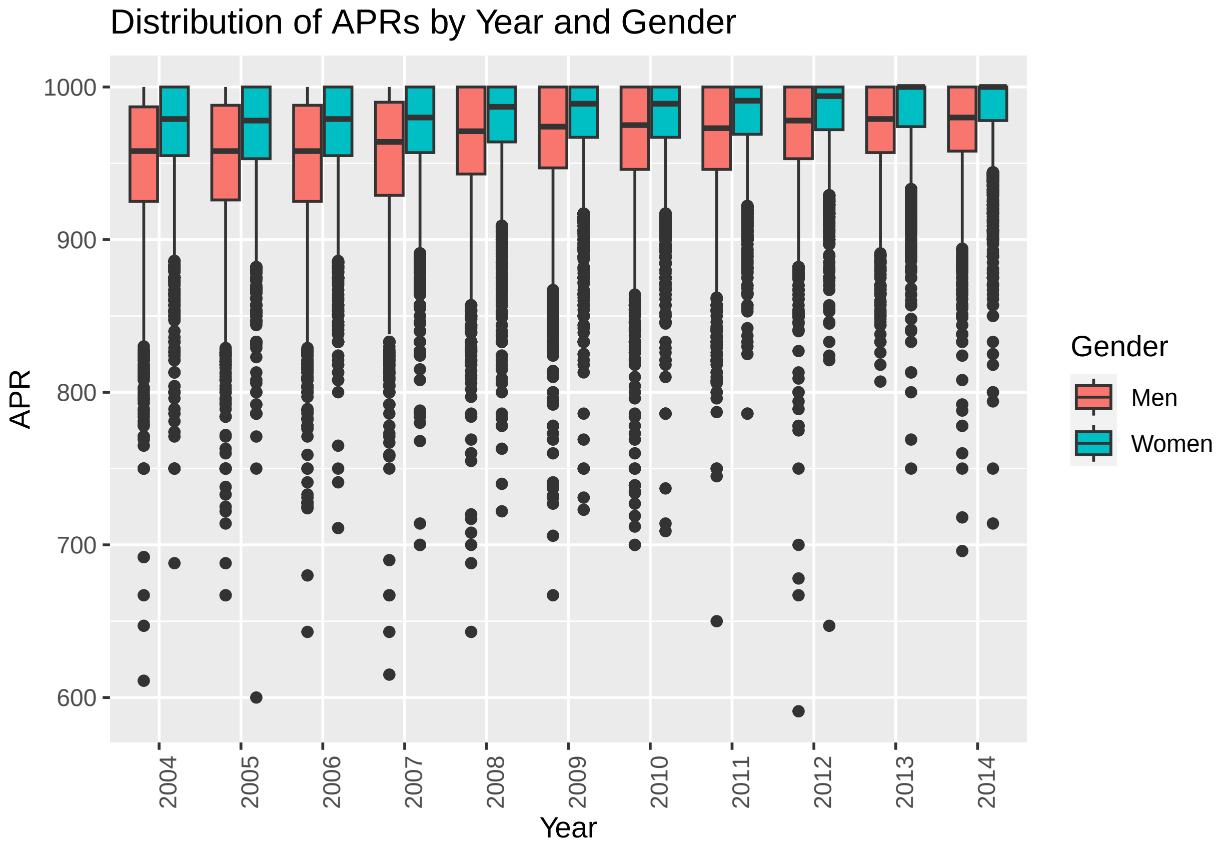
Solution:

```
# creating a new dataframe from existing dataframe to work on
apr_dataset2 <- apr_dataset
```

```r
# using ifelse to fill the new column gender based on the Sports Code Men 1-18 and Women 19-37
apr_dataset2 <- mutate(apr_dataset2, Gender = ifelse(SPORT_CODE <= 18, "Men", "Women"))

# printing dataframe
tibble(apr_dataset2)
```

```
## # A tibble: 66,889 x 7
##    SCL_UNITID SCL_NAME               SPORT_CODE SPORT_NAME    Year    APR Gender
##         <dbl> <chr>                       <dbl> <chr>         <chr> <dbl> <chr>
##  1     100654 Alabama A&M University         20 Women’s Bowl~ 2004   1000 Women
##  2     100654 Alabama A&M University         20 Women’s Bowl~ 2005   1000 Women
##  3     100654 Alabama A&M University         20 Women’s Bowl~ 2006    875 Women
##  4     100654 Alabama A&M University         20 Women’s Bowl~ 2007    958 Women
##  5     100654 Alabama A&M University         20 Women’s Bowl~ 2008   1000 Women
##  6     100654 Alabama A&M University         20 Women’s Bowl~ 2009   1000 Women
##  7     100654 Alabama A&M University         20 Women’s Bowl~ 2010    950 Women
##  8     100654 Alabama A&M University         20 Women’s Bowl~ 2011   1000 Women
##  9     100654 Alabama A&M University         20 Women’s Bowl~ 2012   1000 Women
## 10     100654 Alabama A&M University         20 Women’s Bowl~ 2013   1000 Women
## # ... with 66,879 more rows
```

```r
library(ggplot2)

ggplot(apr_dataset2, aes(x = Year, y = APR, fill = Gender)) +
  geom_boxplot() +
  # facet_wrap(~ Gender) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Distribution of APRs by Year and Gender",
       x = "Year",
       y = "APR")
```



- It can be observed from the above plots that Women sports have higher average APRs than Men's sports
- Also, it can be observed that across the span of 2004-2014, the Women sports average APR has always been greater than Men's sports average APR. So the above relationship holds true for the years 2004-2014.
- This comparison indicates that women's sports progress academically better than men's sports.
- One can see that women sports average APR has been consistently increasing over the years. However, there is some inconsistent growth in the men's sports average APR over the years.

#Q5

We would like to further visualize APR by both gender and specific sports. Process the the sport names to remove the "Men's" and "Women's" prefixes so that we can compare men's and women's teams within each sport. Then visualize the distribution of APR for both men's and women's teams for each sport. Are there sports where men's and women's teams have similar APRs?

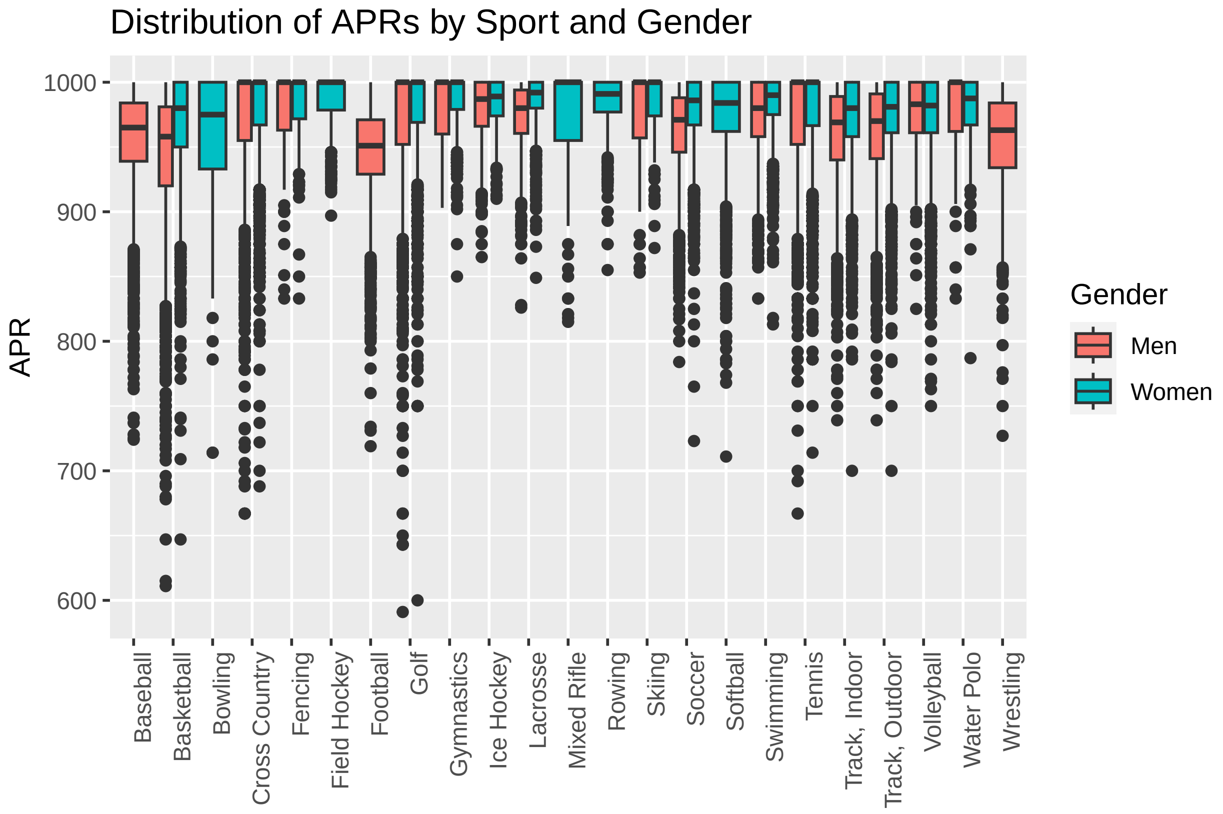
Solution:

```
library(stringr)

# creating a new dataframe from existing dataframe to work on
apr_dataset3 <- apr_dataset2

# used str_remove() to remove to the "Men's" and "Women's" prefixes
apr_dataset3 <- mutate(apr_dataset3,
                                  SPORT_NAME = str_remove(SPORT_NAME, "^Men's |^Women's "))
```

```
# printing dataframe
tibble(apr_dataset3)
```

```
## # A tibble: 66,889 x 7
##    SCL_UNITID SCL_NAME                SPORT_CODE SPORT_NAME Year    APR Gender
##         <dbl> <chr>                        <dbl> <chr>      <chr> <dbl> <chr>
##  1     100654 Alabama A&M University          20 Bowling    2004   1000 Women
##  2     100654 Alabama A&M University          20 Bowling    2005   1000 Women
##  3     100654 Alabama A&M University          20 Bowling    2006    875 Women
##  4     100654 Alabama A&M University          20 Bowling    2007    958 Women
##  5     100654 Alabama A&M University          20 Bowling    2008   1000 Women
##  6     100654 Alabama A&M University          20 Bowling    2009   1000 Women
##  7     100654 Alabama A&M University          20 Bowling    2010    950 Women
##  8     100654 Alabama A&M University          20 Bowling    2011   1000 Women
##  9     100654 Alabama A&M University          20 Bowling    2012   1000 Women
## 10     100654 Alabama A&M University          20 Bowling    2013   1000 Women
## # ... with 66,879 more rows
```

```
ggplot(apr_dataset3, aes(x = SPORT_NAME, y = APR, fill = Gender)) +
  geom_boxplot() +
  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")
```



- As seen in the above visualization, men's and women's teams have similar APRs in:

1. Volleyball
2. Fencing
3. Golf
4. Gymnastics
5. Skiing
6. Tennis
7. Cross Country

- The average APRs of the above sports for both men and women are very similar.
- For rest of the sports, there is quite a distinct difference between the APRs of both genders
- As viewed in the visualization of Q4, the Women APRs are greater than Men APR for maximum sports.