DS5110 Homework 2

Parth Shah

2023-02-13

Part A

Problem 1

Importing Data sets and loading libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

matches <- read_csv("matches.csv", show_col_types = FALSE)
deliveries <- read_csv("deliveries.csv", show_col_types = FALSE)
```

Source: https://www.kaggle.com/datasets/nowke9/ipldata

The Indian Premier League (IPL) is a T-20 cricket tournament for men hosted in India and overseen by the Board of Cricket Control of India (BCCI). It consists of 8 teams that face off in a league format, followed by a knockout stage. Some previous teams, such as Deccan Chargers, Kochi Tuskers, Gujrat Lions, and Pune Warriors, are no longer active in the tournament. Although two teams were banned for two seasons, they have since returned to the tournament. This analysis uses data from 2008 to 2019, and all findings are based on that data. The dataset includes information about IPL matches and deliveries, with the "matches.csv" file providing details about each match such as the match ID, season, location, date, participating teams, toss winner, result, and winning team. The "deliveries.csv" file includes information about each delivery in a match, including the match ID, inning, batting and bowling teams, batsman and bowler, and the runs scored.

Variables:

matches.csv:

```
-id: Unique match ID
-season: Season of the IPL tournament
-city: City where the match was played
-date: Date of the match
-team1: Name of team 1
-team2: Name of team 2
-toss_winner: Team that won the toss
-toss_decision: Toss decision (bat or field)
-result: Result of the match
-dl_applied: Duckworth-Lewis method applied or not
-winner: Winning team
-win_by_runs: Number of runs by which the team won
-win_by_wickets: Number of wickets by which the team won
-player_of_match: Player of the match
-venue: Venue where the match was played
-umpire1: Name of umpire 1
-umpire2: Name of umpire 2
-umpire3: Name of umpire 3
```

deliveries.csv:

```
-match_id: Match ID
-inning: Inning of the match
-batting_team: Batting team
-bowling_team: Bowling team
-over: Over number
-ball: Ball number
-batsman: Batsman who faced
-non_striker: Non-striker batsman
-bowler: Bowler who bowled the delivery
-is_super_over: Whether the delivery was in a super over or not
-wide_runs: Number of wide runs
-bye_runs: Number of bye runs
-legbye_runs: Number of legbye runs
-noball_runs: Number of no ball runs
-penalty_runs: Number of penalty runs
-batsman_runs: Number of runs scored by the batsman
-extra_runs: Number of extra runs
-total_runs: Total number of runs scored in that delivery
-player_dismissed: Batsman who got dismissed
-dismissal_kind: Kind of dismissal
-fielder: Fielder who made the dismissal (if any)
```

Pre-process the data

```r
matches$date <- as.Date(matches$date, format = "%d/%m/%y")
```

```r
matches$team1 <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       matches$team1)
```

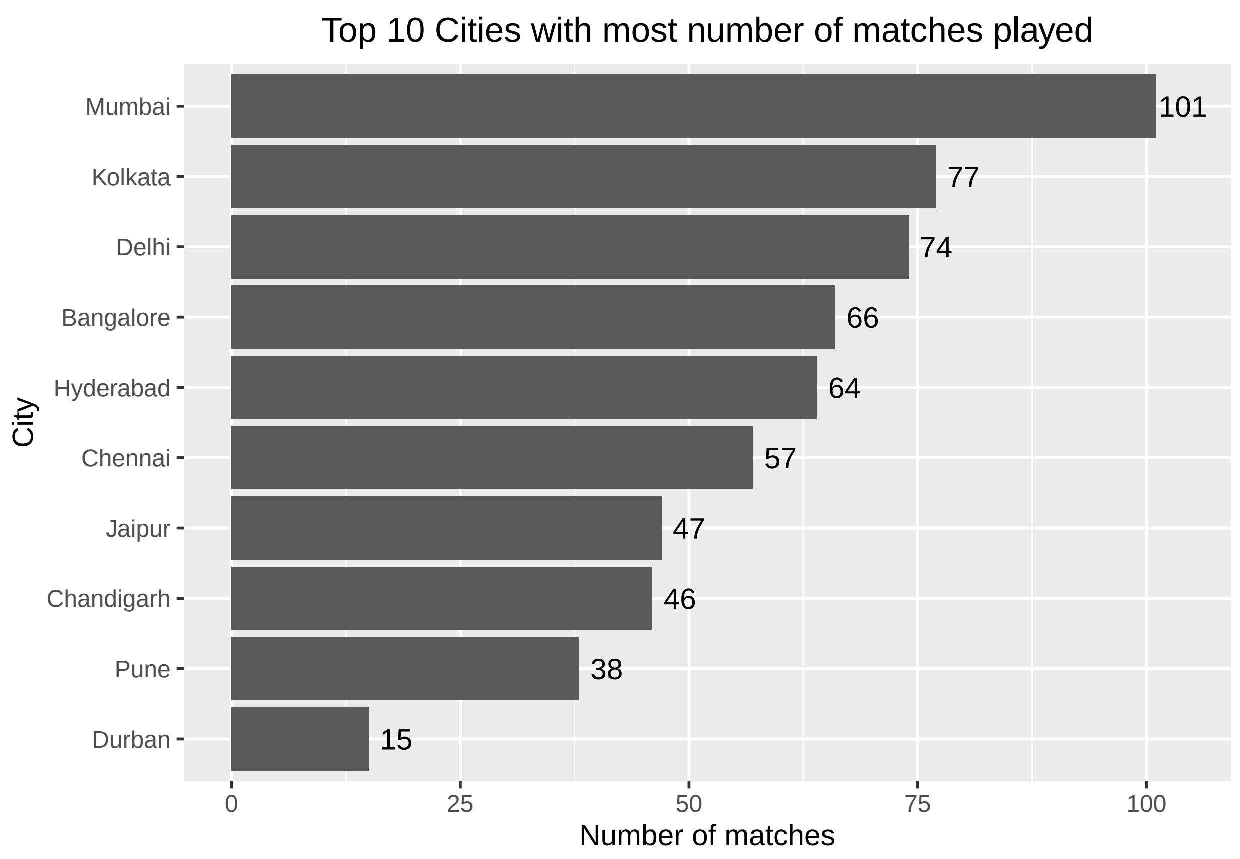
```r
matches$team2 <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       matches$team2)
matches$winner <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       matches$winner)
matches$toss_winner <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       matches$toss_winner)
deliveries$batting_team <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       deliveries$batting_team)
deliveries$bowling_team <-
  gsub("Rising Pune Supergiants",
       "Rising Pune Supergiant",
       deliveries$bowling_team)
matches$venue <-
  gsub("Feroz Shah Kotla Ground", "Feroz Shah Kotla", matches$venue)
matches$venue <-
  gsub("M Chinnaswamy Stadium", "M. Chinnaswamy Stadium", matches$venue)
matches$venue <-
  gsub("MA Chidambaram Stadium, Chepauk",
       "M.A. Chidambaram Stadium",
       matches$venue)
matches$venue <-
  gsub("M. A. Chidambaram Stadium",
       "M.A. Chidambaram Stadium",
       matches$venue)
matches$venue <-
  gsub(
    "Punjab Cricket Association IS Bindra Stadium, Mohali",
    "Punjab Cricket Association Stadium",
    matches$venue
  )
matches$venue <-
  gsub(
    "Punjab Cricket Association Stadium, Mohali",
    "Punjab Cricket Association Stadium",
    matches$venue
  )
matches$venue <-
  gsub("IS Bindra Stadium",
       "Punjab Cricket Association Stadium",
       matches$venue)
matches$venue <-
  gsub(
    "Rajiv Gandhi International Stadium, Uppal",
    "Rajiv Gandhi International Stadium",
    matches$venue
  )
matches$venue <-
  gsub(
    "Rajiv Gandhi Intl. Cricket Stadium",
    "Rajiv Gandhi International Stadium",
    matches$venue
  )
```

I have converted the date column in the matches.csv file from character to data format. Also, there were inconsistencies among the names of the same teams and venues, and I have tried to make the uniform as well. Further, there are many NA values in the data, but they have significance as it represents that those values don't exist, rather than being existent but unknown.

Problem 2

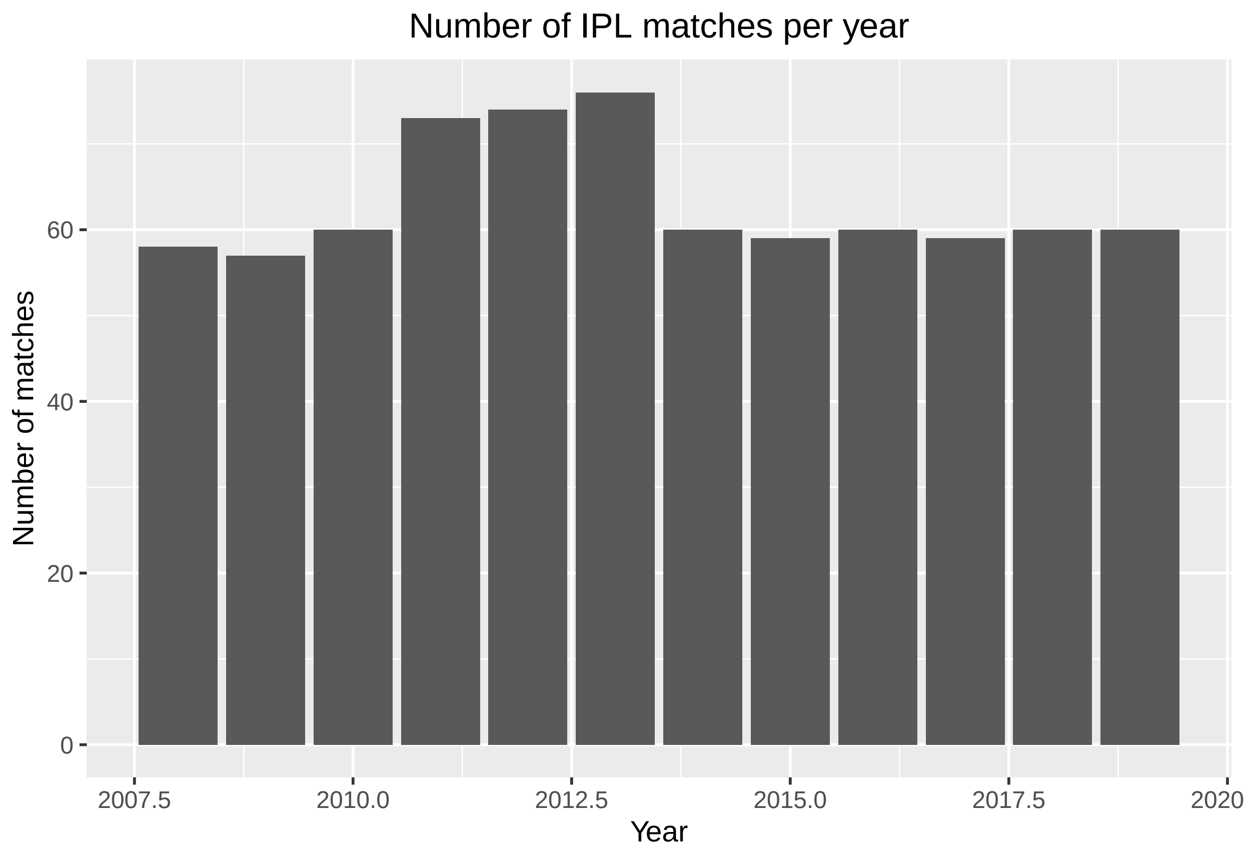
Visualizations

```r
city_matches <- matches %>%
  group_by(city) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

top_10_cities <- head(city_matches, 10)

ggplot(top_10_cities, aes(x = reorder(city, count, FUN = rev), y = count)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = count), nudge_y = 3) +
  labs(x = "City", y = "Number of matches") +
  coord_flip() +
  ggtitle("Top 10 Cities with most number of matches played") +
  theme(plot.title = element_text(hjust = 0.5))
```



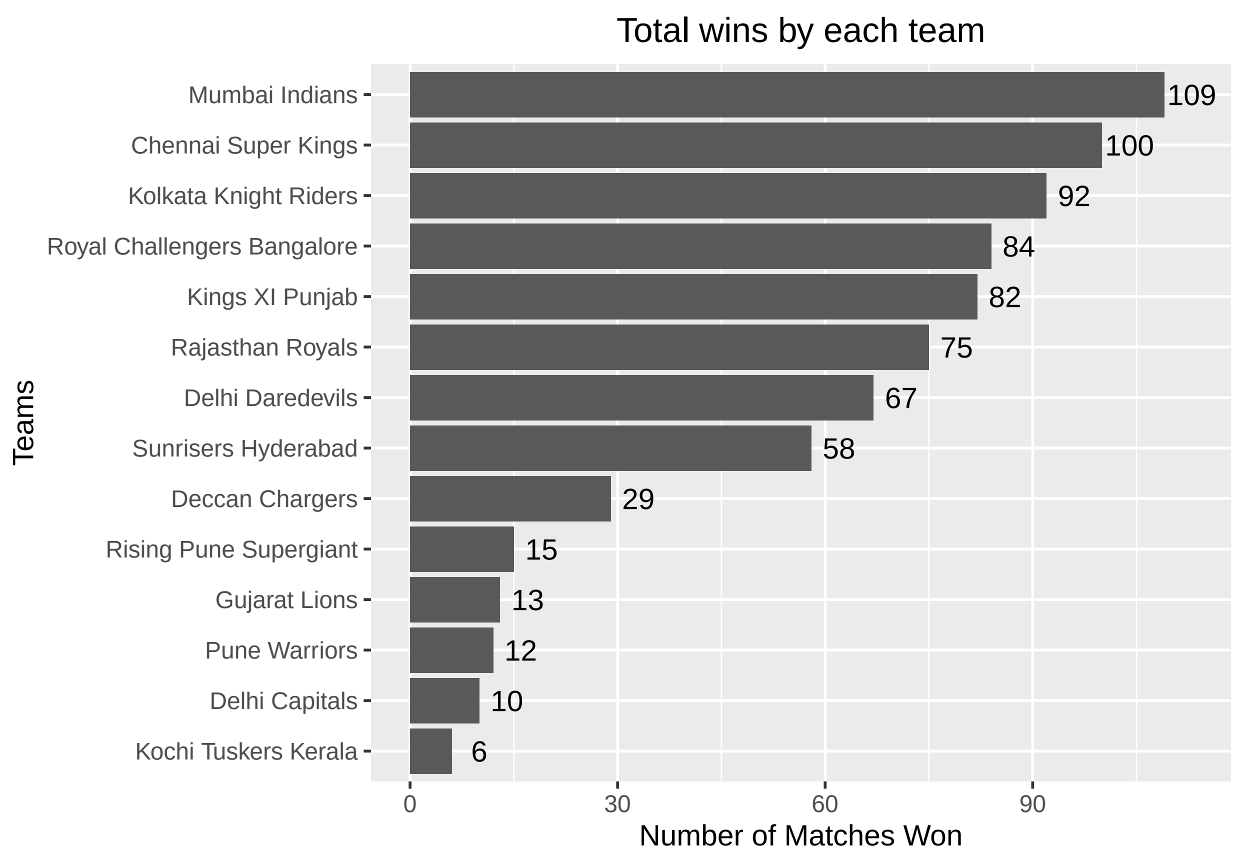
The plot displays the top 10 cities with the most matches played. The data suggest some cities host more matches than others, indicating their popularity and support for cricket. The higher number of matches played in some cities can reflect better infrastructure, a larger fan base, or popularity for hosting cricket events, as well as more resources or teams based in these cities.

```r
matches_per_year <- matches %>%
  group_by(season) %>%
  summarise(count = n())

ggplot(matches_per_year, aes(x = season, y = count)) +
  geom_bar(stat = "identity") +
  labs(x = "Year", y = "Number of matches") +
  ggtitle("Number of IPL matches per year") +
  theme(plot.title = element_text(hjust = 0.5))
```



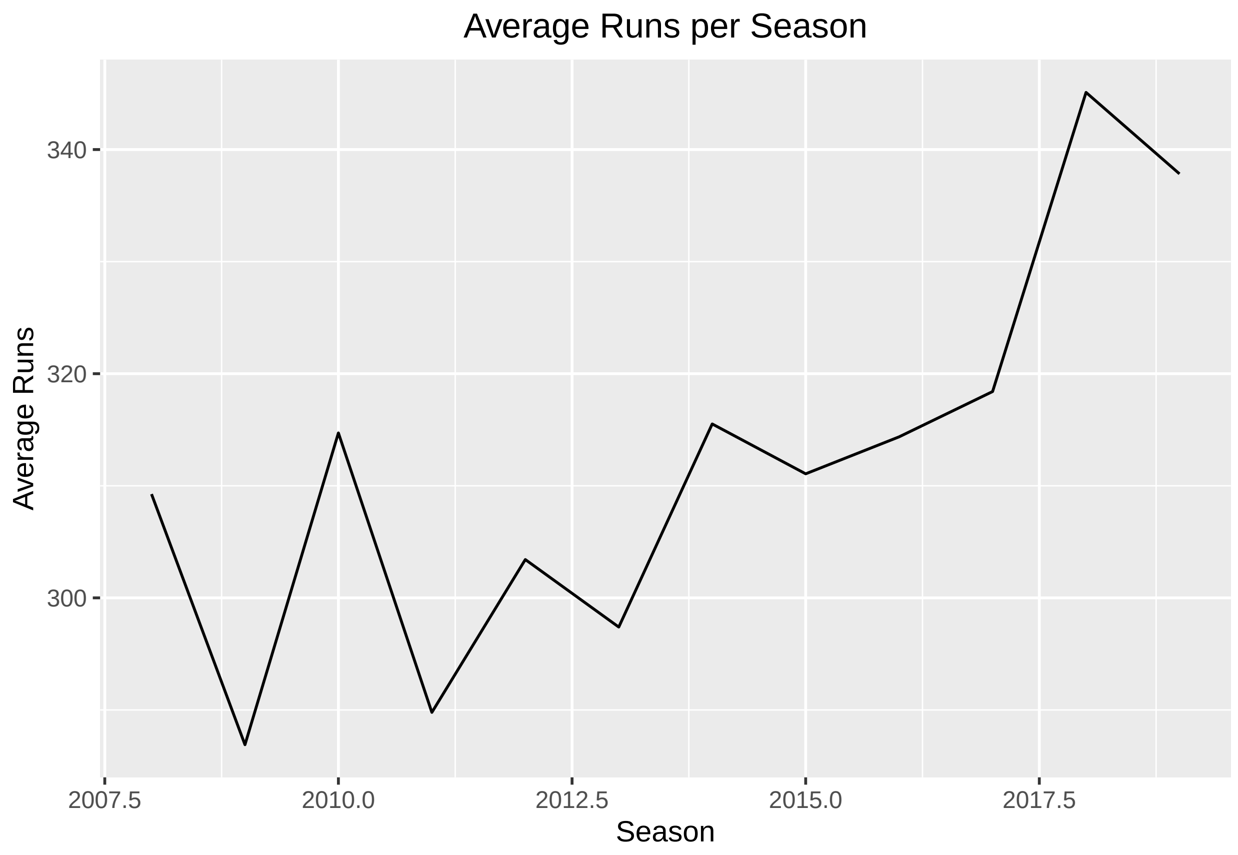
The bar plot displays the distribution of matches across different seasons. The higher number of matches in seasons 2011, 2012, and 2013 suggests an increase in the number of participating teams or a change in the league format.

```
wins <- matches %>%
  count(winner) %>%
  rename(name = winner)

ggplot(na.omit(wins), aes(x = reorder(name, n, FUN = rev), y = n)) +
  geom_bar(stat = "identity") +
  ggtitle("Total wins by each team") +
  xlab("Teams") +
  ylab("Number of Matches Won") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = n), nudge_y = 4) +
  coord_flip()
```



The bar plot presents us with valuable insights into the winnings of various teams between 2008-2019. It is apparent that some teams like Mumbai Indians, Chennai Super Kings, and Kolkata Knight Riders have been the most successful and dominant, as they have won the tournament 4, 3, and 2 times, respectively.

```r
batsmen <- merge(matches[c('id', 'season')], deliveries, by.x='id', by.y='match_id')
batsmen <- batsmen[, !(names(batsmen) %in% c('id'))]
season <- aggregate(total_runs ~ season, data=batsmen, FUN=sum)

avgruns_each_season <- aggregate(id ~ season, data=matches, FUN=length)
names(avgruns_each_season) <- c('season', 'matches')
avgruns_each_season$total_runs <- season$total_runs
avgruns_each_season$average_runs_per_match <- avgruns_each_season$total_runs / avgruns_each_season$matc

ggplot() +
  geom_line(data=avgruns_each_season, aes(x=season, y=average_runs_per_match)) +
  xlab("Season") + ylab("Average Runs") +
  ggtitle("Average Runs per Season")+
   theme(plot.title = element_text(hjust = 0.5))
```



This line plot demonstrates a significant increase in the average runs scored per season over the years. This upward trend in the average runs per season indicates that the performance of batsmen in the Indian Premier League has improved with time. The improvement could be attributed to various factors such as advanced training techniques, enhanced infrastructure, and a higher caliber of players.

```r
s_man_of_match <- matches %>%
  group_by(player_of_match) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) %>%
  head(15)

df_man_of_match <- data.frame(s_man_of_match) %>%
  rename(times = count)
  # reset_index()

# Then, create the century dataset
cen <- deliveries %>%
  group_by(batsman, match_id) %>%
  summarize(batsman_runs = sum(batsman_runs)) %>%
  filter(batsman_runs >= 100) %>%
  group_by(batsman) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

```
## ‘summarise()‘ has grouped output by ’batsman’. You can override using the
## ‘.groups‘ argument.
```

```r
  # reset_index()

# Half-century dataset
half_cen <- deliveries %>%
  group_by(batsman, match_id) %>%
  summarize(batsman_runs = sum(batsman_runs)) %>%
  filter(batsman_runs >= 50 & batsman_runs < 100) %>%
  group_by(batsman) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

```
## `summarise()` has grouped output by 'batsman'. You can override using the
## `.groups` argument.
```

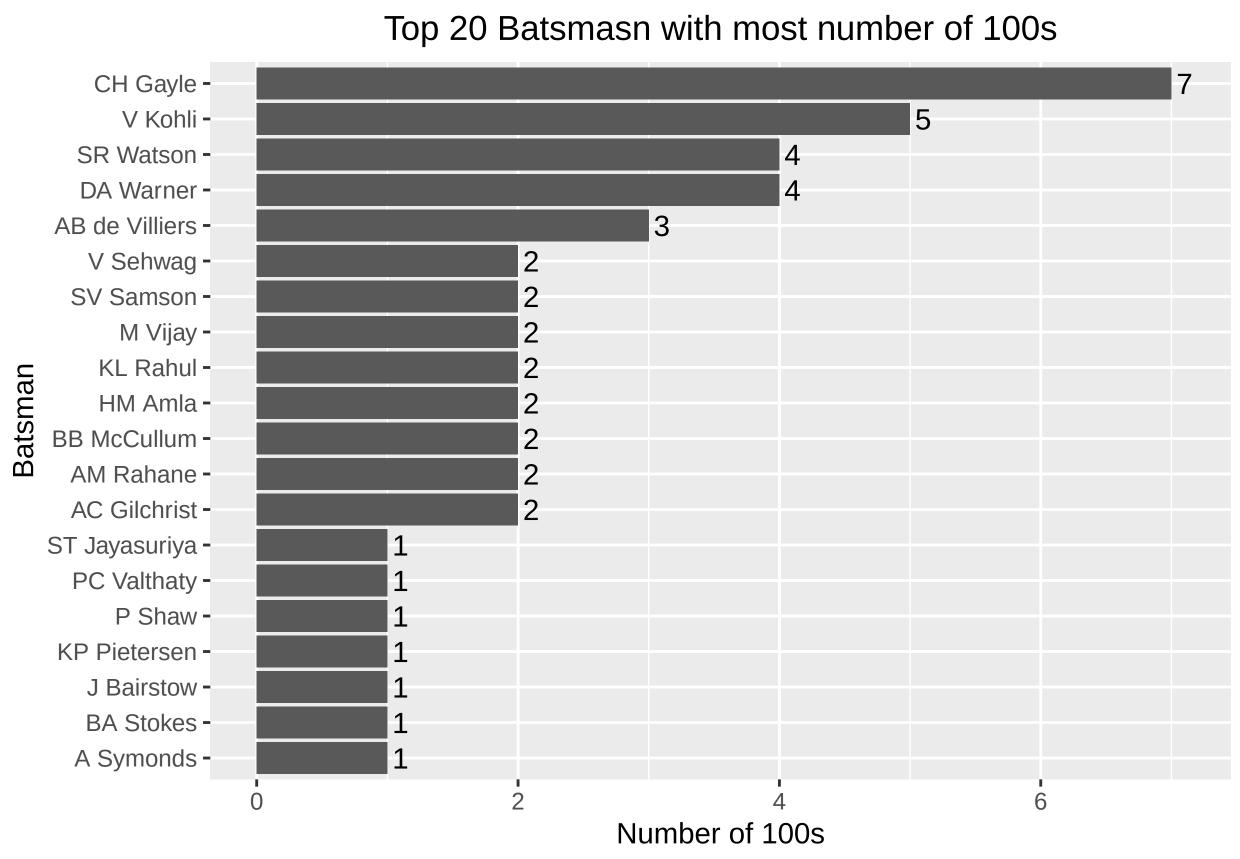
```r
  # reset_index()

# Merge the century and half-century datasets
df_big <- merge(cen, half_cen, by = "batsman", all = TRUE) %>%
  replace_na(list(count.x = 0, count.y = 0))

# Strike rate dataset
df_strike_rate <- deliveries %>%
  group_by(batsman) %>%
  summarize(Ball = n(), batsman_runs = mean(batsman_runs)) %>%
  arrange(desc(batsman_runs))
df_strike_rate <- rename(df_strike_rate, Strike.Rate = batsman_runs)
df_strike_rate$Strike.Rate <- df_strike_rate$Strike.Rate * 100

# Runs per match dataset
df_runs_per_match <- deliveries %>%
  group_by(batsman, match_id) %>%
  summarize(batsman_runs = sum(batsman_runs))
```

```
## `summarise()` has grouped output by 'batsman'. You can override using the
## `.groups` argument.
```

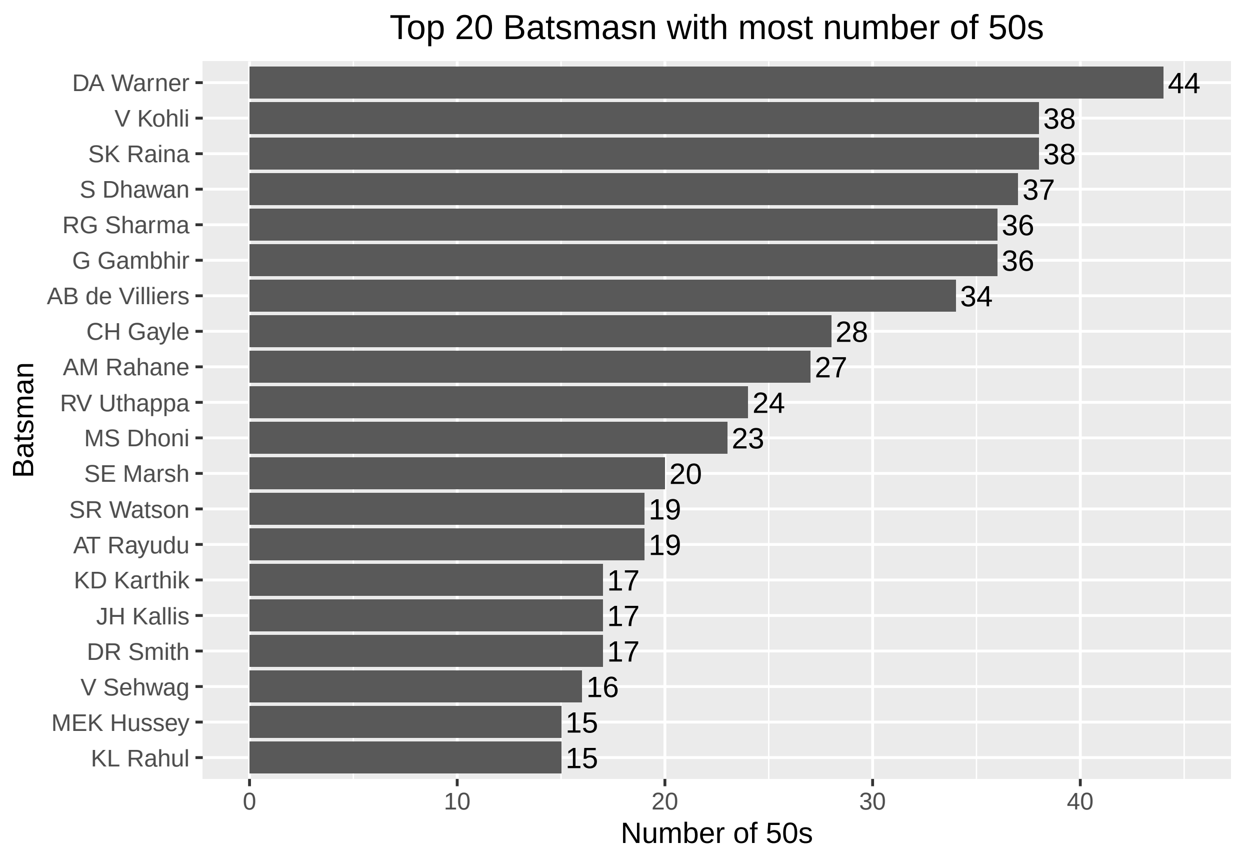
```r
df_total_runs <- df_runs_per_match %>%
  group_by(batsman) %>%
  summarize(Batsman.Run = sum(batsman_runs), Match.Count = n(), Average.score = mean(batsman_runs))

# Number of sixes and fours datasets
df_sixes <- deliveries %>%
  filter(batsman_runs == 6) %>%
  group_by(batsman) %>%
  summarize(Six = n())
df_four <- deliveries %>%
  filter(batsman_runs == 4) %>%
  group_by(batsman) %>%
  summarize(Four = n())

# Merge all the datasets
df_batsman_stat <- full_join(full_join(full_join(df_strike_rate, df_total_runs, by = "batsman"), df_six
colnames(df_batsman_stat) <- c("batsman", "Ball", "Strike_Rate", "Batsman_Run", "Match_Count", "Average_
# df_batsman_stat$Strike_Rate <- df_batsman_stat$Strike_Rate * 100
df_batsman_stat <- df_batsman_stat %>% arrange(desc(Batsman_Run))
  # reset_index(drop = TRUE)

batsman_stats <- full_join(df_batsman_stat, df_big, by = "batsman") %>% replace_na(list(`count_x` = 0, 
colnames(batsman_stats) <- c("batsman", "Ball", "Strike_Rate", "Batsman_Run", "Match_Count", "Average_s
```

```
centuries <- batsman_stats[order(batsman_stats$`100s`),]
half_centuries <- batsman_stats[order(batsman_stats$`50s`),]
centuries[is.na(centuries)] <- 0
half_centuries[is.na(half_centuries)] <- 0
centuries <- centuries[order(centuries$`100s`),]
half_centuries <- centuries[order(centuries$`50s`),]
centuries <- tail(centuries, 20)
half_centuries <- tail(half_centuries, 20)
```

```
ggplot(centuries, aes(x = reorder(batsman,`100s`, FUN = rev), y = `100s`)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = `100s`), nudge_y = 0.1) +
  labs(x = "Batsman", y = "Number of 100s") +
  ggtitle("Top 20 Batsmasn with most number of 100s") +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_flip()
```



```r
ggplot(half_centuries, aes(x = reorder(batsman,`50s`, FUN = rev), y = `50s`)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = `50s`), nudge_y = 1) +
  labs(x = "Batsman", y = "Number of 50s") +
  ggtitle("Top 20 Batsmasn with most number of 50s") +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_flip()
```



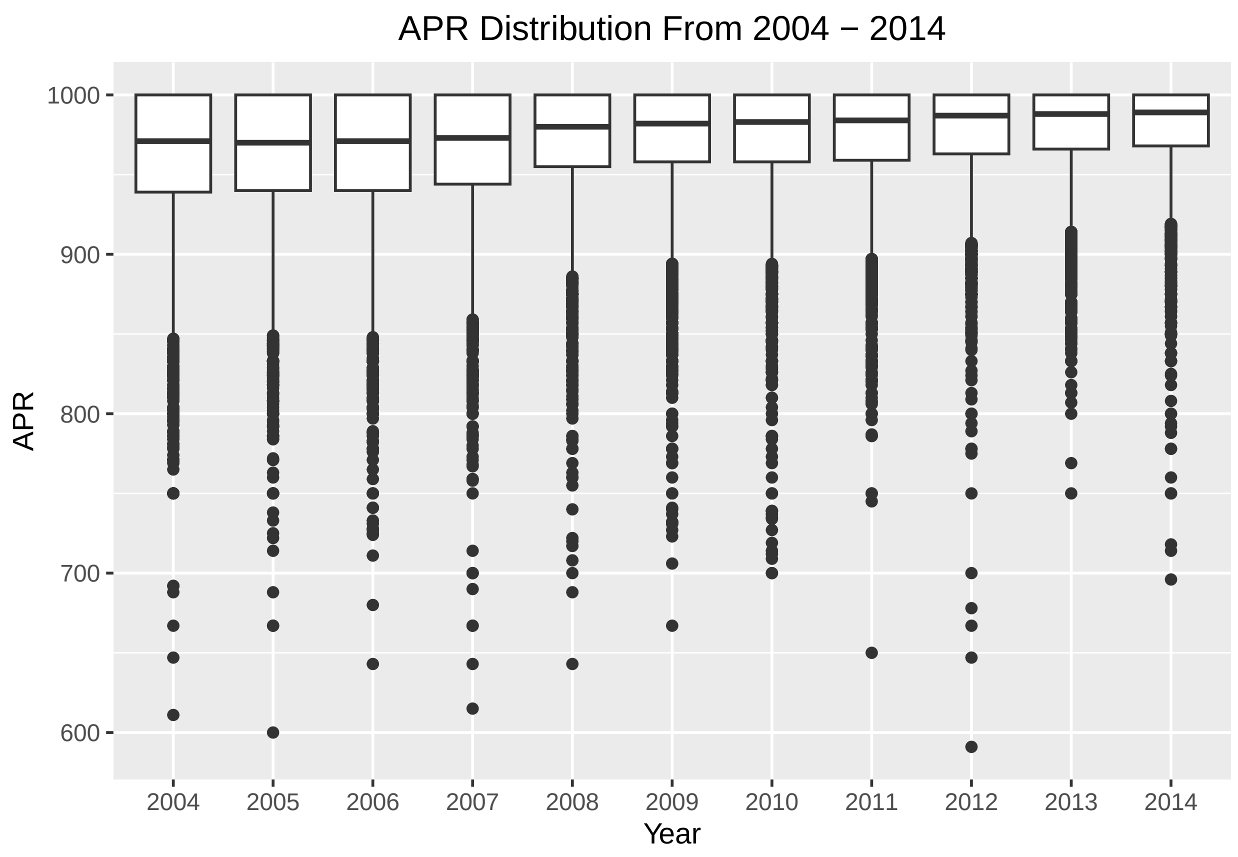
Part B

Problem 3

```r
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
## # ... with 71,611 more rows
```

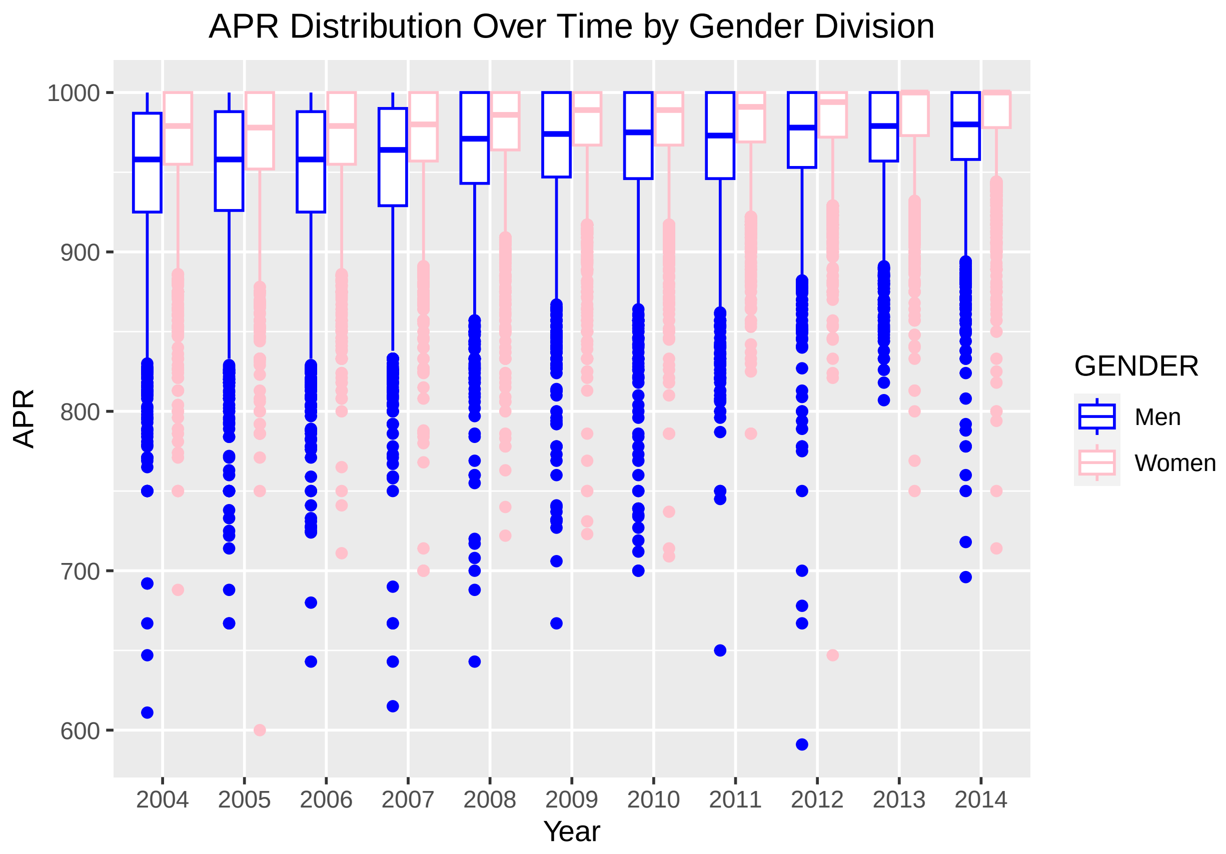
```
ggplot(data, aes(x = YEAR, y = APR)) +
  geom_boxplot(na.rm = TRUE) +
  ggtitle("APR Distribution From 2004 - 2014") +
  xlab("Year") +
  ylab("APR") +
  theme(plot.title = element_text(hjust = 0.5))
```



The average annual percentage rates (APRs) have been consistently increasing over time, as evidenced by the upward trend in both the first quantile and median values.

Problem 4

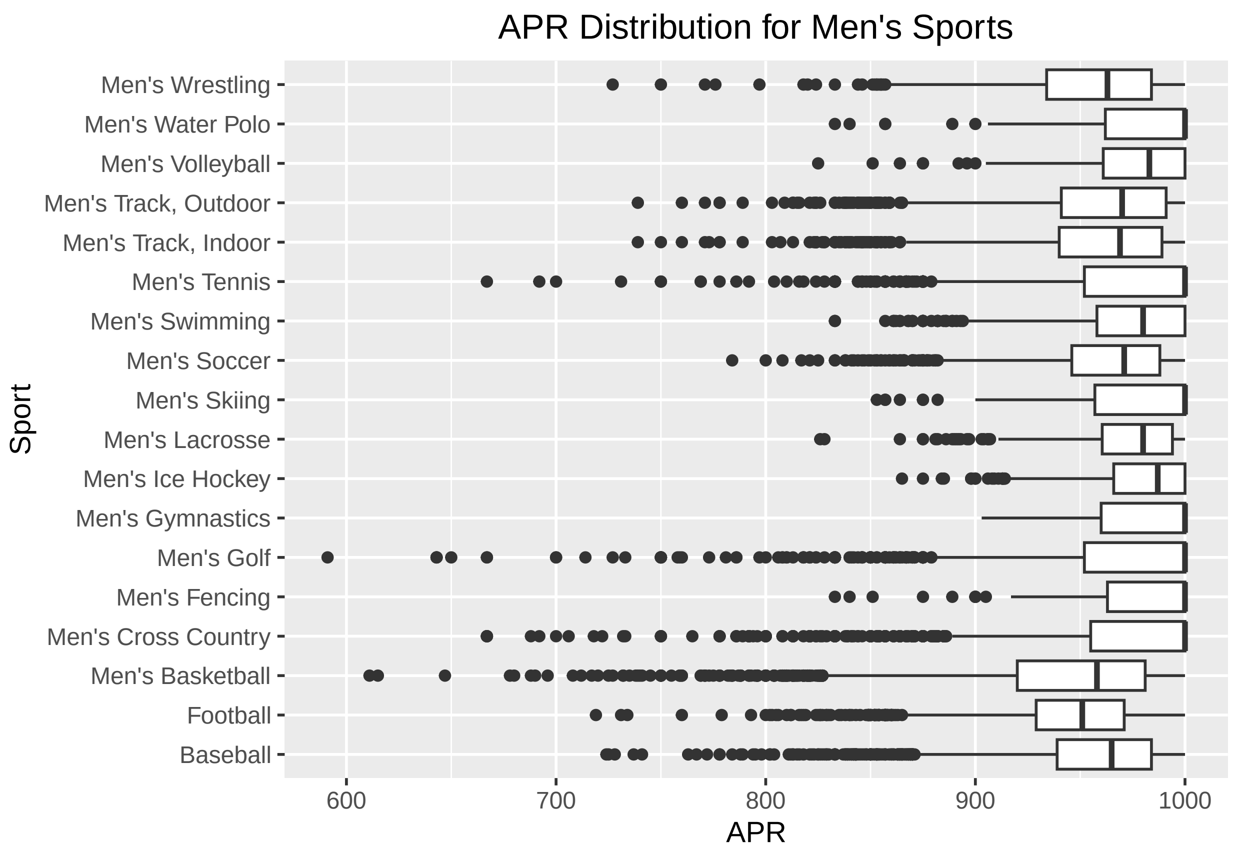
```r
data %>%
  filter(SPORT_CODE >= 1 & SPORT_CODE <= 37) %>%
  mutate(GENDER = ifelse(SPORT_CODE >= 1 &
                         SPORT_CODE <= 18, "Men", "Women")) -> data

ggplot(data, aes(x = YEAR, y = APR, color = GENDER)) +
  geom_boxplot(na.rm = TRUE) +
  scale_color_manual(values = c("Blue", "Pink")) +
  ggtitle("APR Distribution Over Time by Gender Division") +
  xlab("Year") +
  ylab("APR") +
  theme(plot.title = element_text(hjust = 0.5))
```



The average APR for women has consistently been higher than for men, both for individual and team performance, from 2004 to 2014.

Problem 5

```r
data %>%
  filter(GENDER == "Men") %>%
  ggplot(aes(x = SPORT_NAME, y = APR)) +
    geom_boxplot(na.rm = TRUE) +
    ggtitle("APR Distribution for Men's Sports") +
    xlab("Sport") +
    ylab("APR") +
    theme(plot.title = element_text(hjust = 0.5)) +
    coord_flip()
```



On average, men's sports teams such as football, basketball, and baseball tend to have lower APRs, while fencing, skiing, and water polo teams tend to have higher APRs. However, men's gymnastics has a consistently higher APR compared to other sports without any outliers, whereas men's football and basketball have a consistently lower APR.