HW4

Vanesa Avoyan

2025-03-30

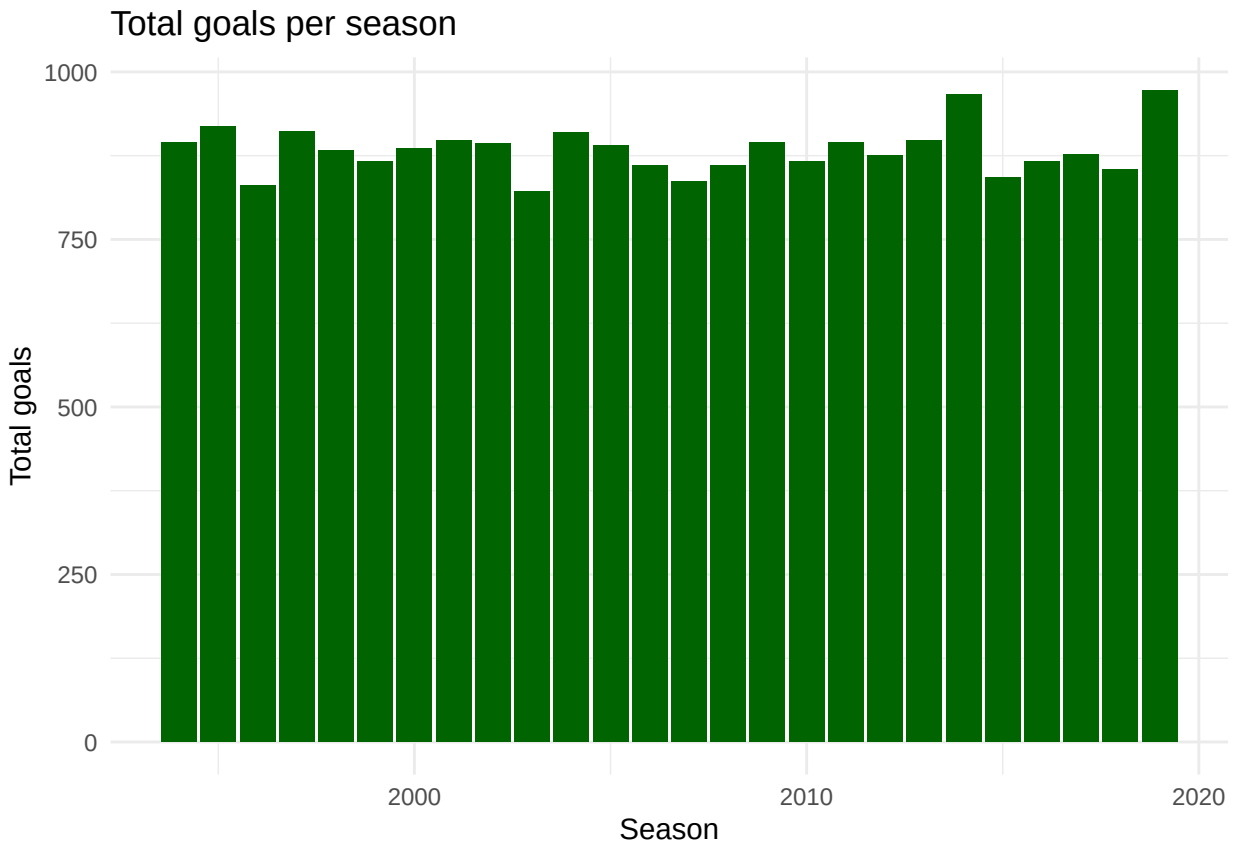
```
library(dplyr)
library(ggplot2)
library(stringr)
library(scales)
library(lubridate)
library(tidyr)
```

```
bundesliga <- read.csv("bundesliga.csv")
```

```
bundesliga_cleaned <- read.csv("bundesliga2.csv")
```

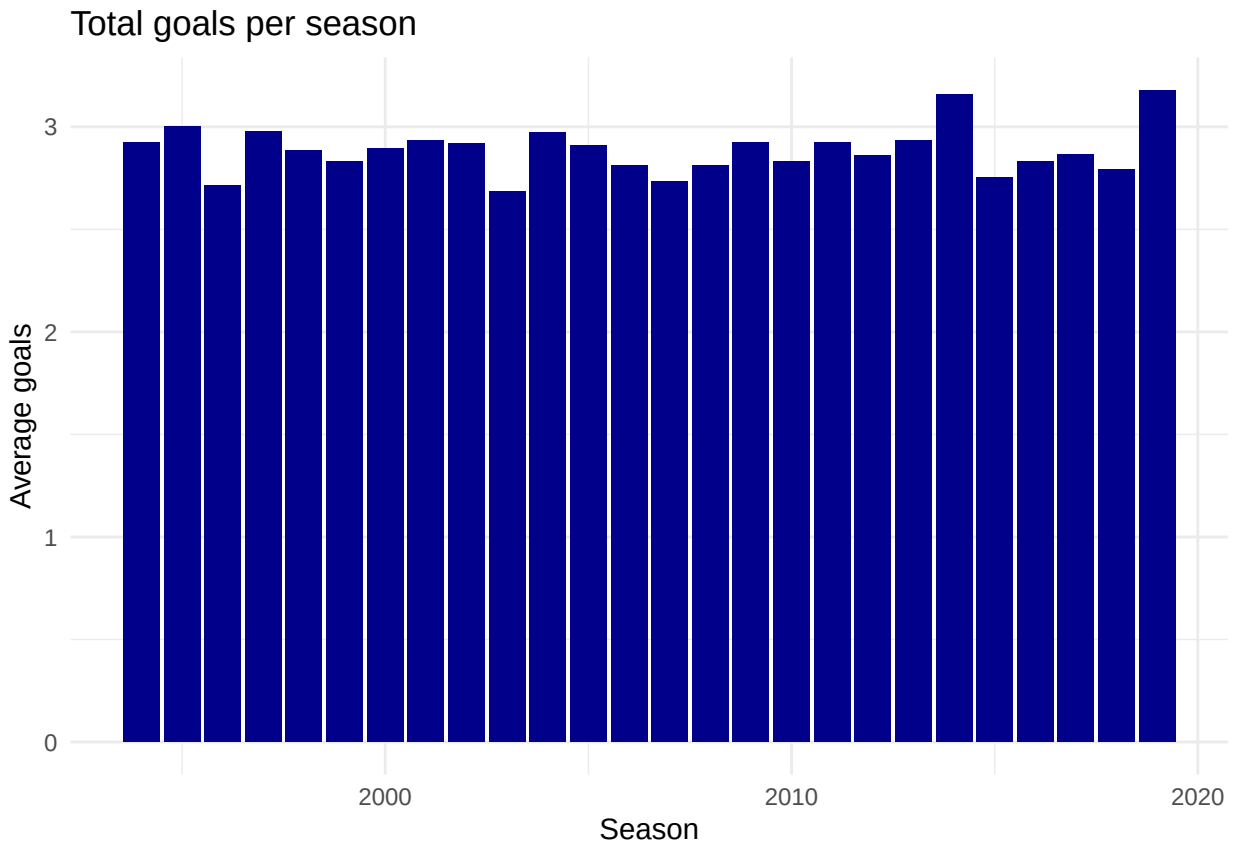
Part 1. 1) Total goals per match

```
bundes_Part1 <- bundesliga %>%
  group_by(SEASON) %>%
  summarise(Total_goals = sum(FTTG),
            Average_goals = mean(FTTG))

ggplot(data = bundes_Part1, aes(x = SEASON, y = Total_goals)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  labs(title = "Total goals per season", x = "Season", y = "Total goals") +
  theme_minimal()
```



Average goals per match.

```
ggplot(data = bundes_Part1, aes(x = SEASON, y = Average_goals)) +
  geom_bar(stat = "identity", fill = "darkblue") +
  labs(title = "Total goals per season", x = "Season", y = "Average goals") +
  theme_minimal()
```



2)******************************

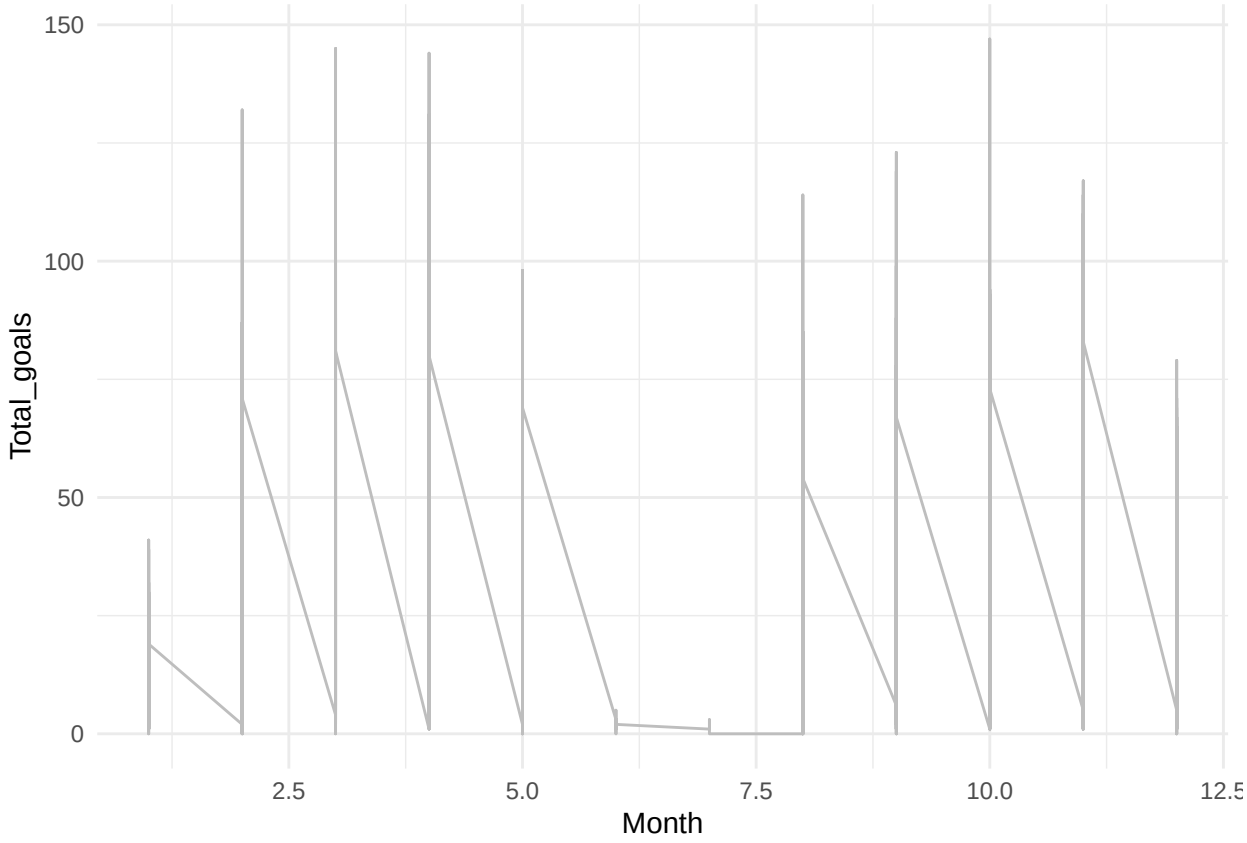
```
summary(bundesliga)
```

```
##      SEASON        LEAGUE              DATE             HOMETEAM
##  Min.   :1994   Length:7956        Length:7956        Length:7956
##  1st Qu.:2000   Class :character   Class :character   Class :character
##  Median :2006   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2006
##  3rd Qu.:2013
##  Max.   :2019
##    AWAYTEAM             FTSC                FTHG            FTAG
##  Length:7956        Length:7956        Min.   :0.000   Min.   :0.000
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.000
##  Mode  :character   Mode  :character   Median :1.000   Median :1.000
##                                        Mean   :1.665   Mean   :1.221
##                                        3rd Qu.:2.000   3rd Qu.:2.000
##                                        Max.   :9.000   Max.   :9.000
##       FTTG
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 3.000
##  Mean   : 2.887
##  3rd Qu.: 4.000
##  Max.   :11.000
```

3)

```r
bundesliga <- bundesliga %>%
  mutate(Month = month(DATE))

bundes1.3 <- bundesliga %>%
  group_by(Month, HOMETEAM) %>%
  summarise(Total_goals = sum(FTHG), .groups = "drop")
```

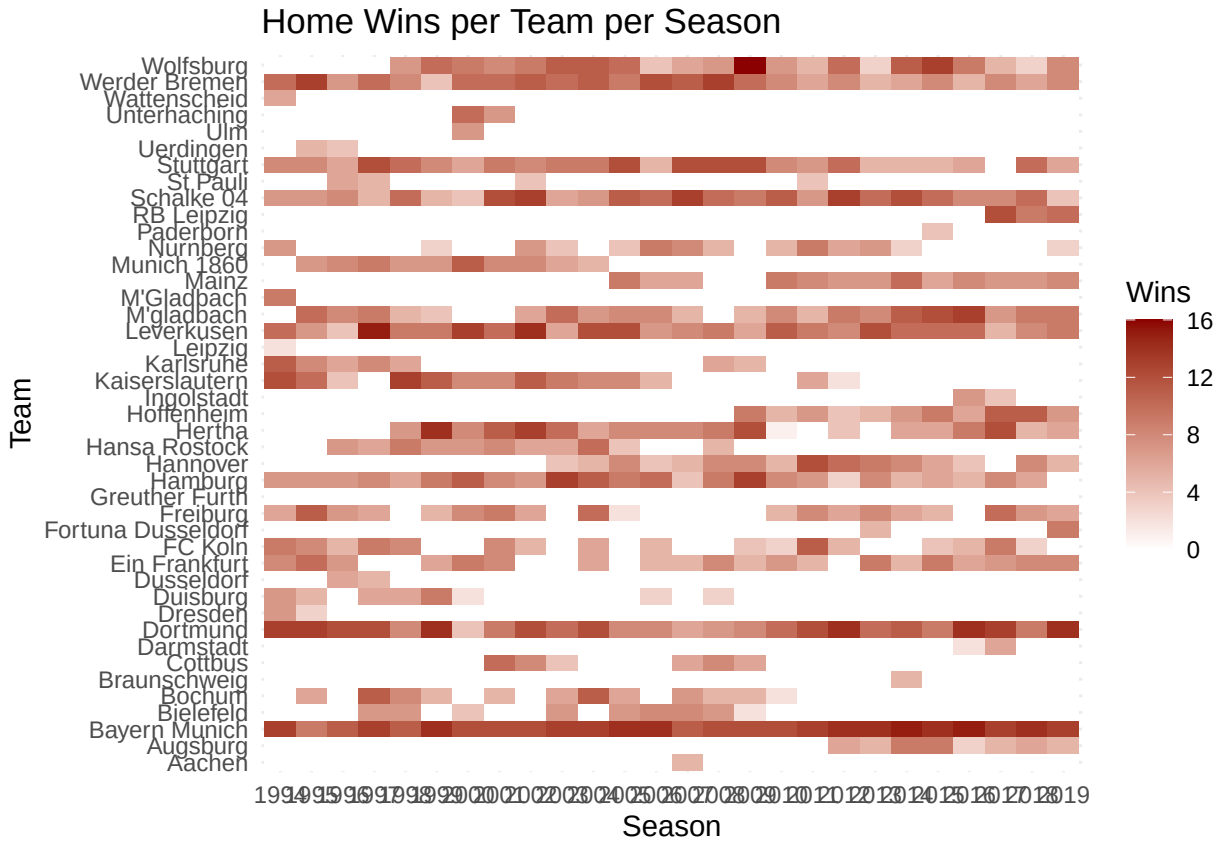
```r
ggplot(data = bundes1.3, aes(x = Month, y = Total_goals)) +
  geom_line(colour = "grey") +
  theme_minimal()
```



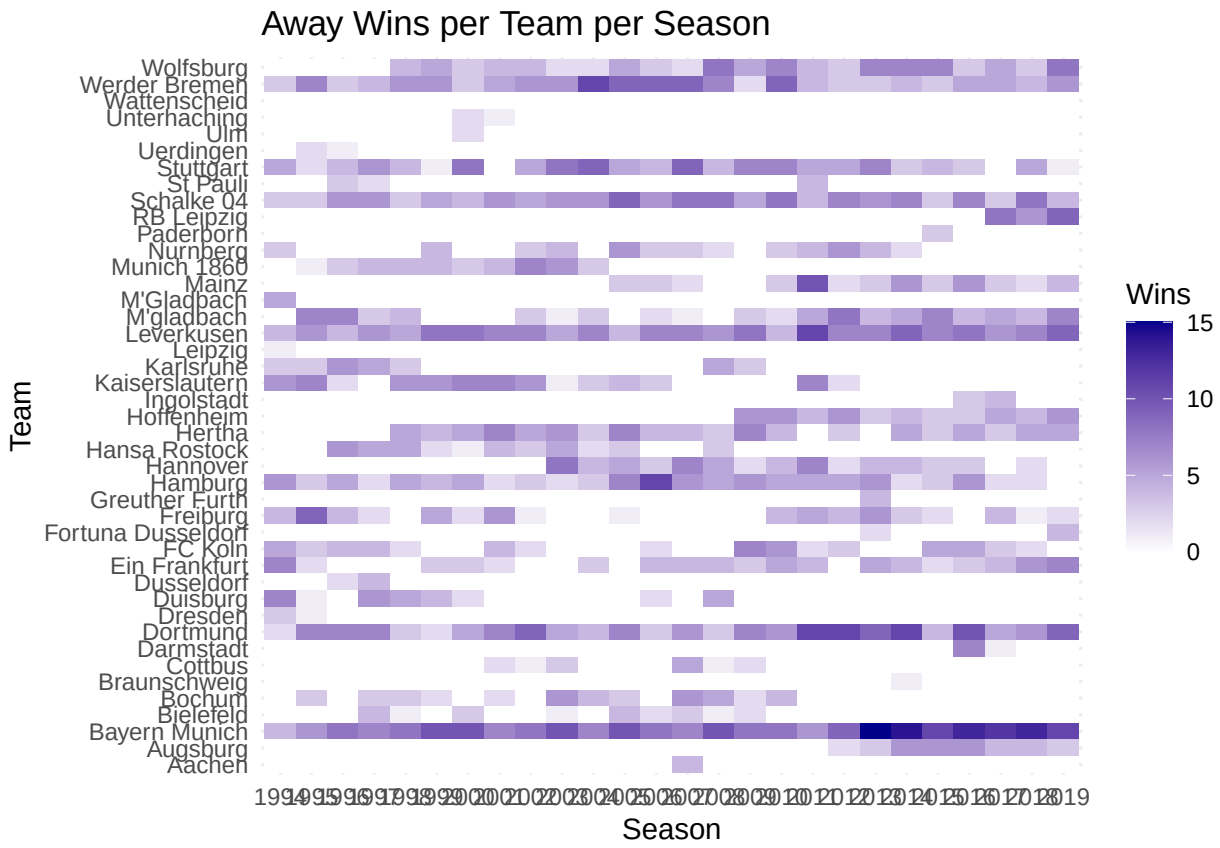
Part 2. 1)

```r
home_wins <- bundesliga %>%
  filter(FTHG > FTAG) %>%
  group_by(SEASON, HOMETEAM) %>%
  summarise(HomeWins = n(), .groups = "drop") %>%
  rename(Team = HOMETEAM)

# Calculate Away Wins
away_wins <- bundesliga %>%
  filter(FTHG < FTAG) %>%
  group_by(SEASON, AWAYTEAM) %>%
  summarise(AwayWins = n(), .groups = "drop") %>%
  rename(Team = AWAYTEAM)

# Merge Home and Away Wins
wins <- full_join(home_wins, away_wins, by = c("SEASON", "Team")) %>%
```

```
  mutate(HomeWins = replace_na(HomeWins, 0), AwayWins = replace_na(AwayWins, 0))

# Reshape for Heatmap (Home Wins)
home_wins_heatmap <- wins %>%
  select(SEASON, Team, HomeWins) %>%
  pivot_wider(names_from = SEASON, values_from = HomeWins, values_fill = 0)

# Reshape for Heatmap (Away Wins)
away_wins_heatmap <- wins %>%
  select(SEASON, Team, AwayWins) %>%
  pivot_wider(names_from = SEASON, values_from = AwayWins, values_fill = 0)

# Create Heatmap (Home Wins)
ggplot(pivot_longer(home_wins_heatmap, cols = -Team, names_to = "SEASON", values_to = "Wins"),
       aes(x = SEASON, y = Team, fill = Wins)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "darkred") +
  labs(title = "Home Wins per Team per Season", x = "Season", y = "Team", fill = "Wins") +
  theme_minimal()
```

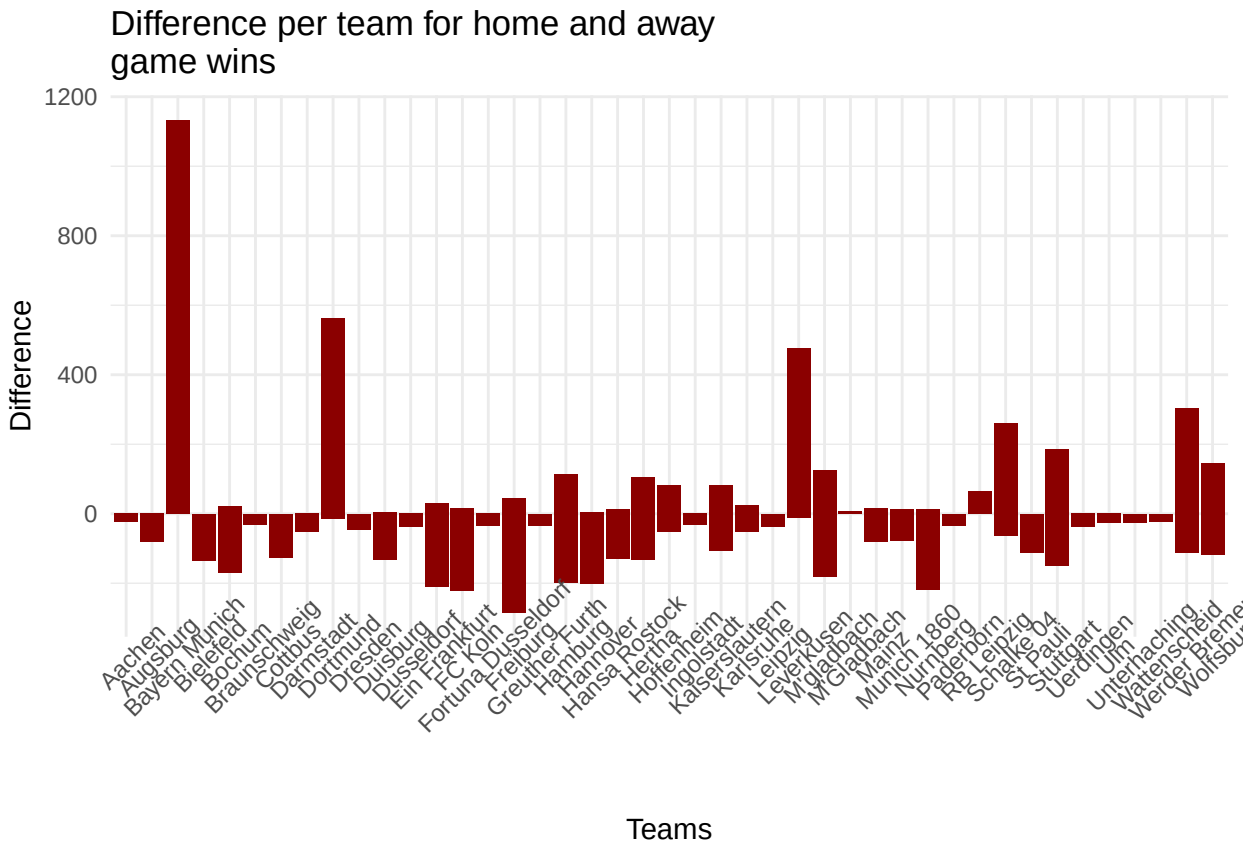



```
# Create Heatmap (Away Wins)
ggplot(pivot_longer(away_wins_heatmap, cols = -Team, names_to = "SEASON", values_to = "Wins"),
       aes(x = SEASON, y = Team, fill = Wins)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "darkblue") +
  labs(title = "Away Wins per Team per Season", x = "Season", y = "Team", fill = "Wins") +
  theme_minimal()
```



2)

```
ggplot(data = bundesliga_cleaned, aes(x = TEAM,
                                      y = DIFF)) +
  geom_bar(stat = "identity", fill = "darkred") +
  labs(title = "Difference per team for home and away
game wins", x = "Teams", y = "Difference") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```

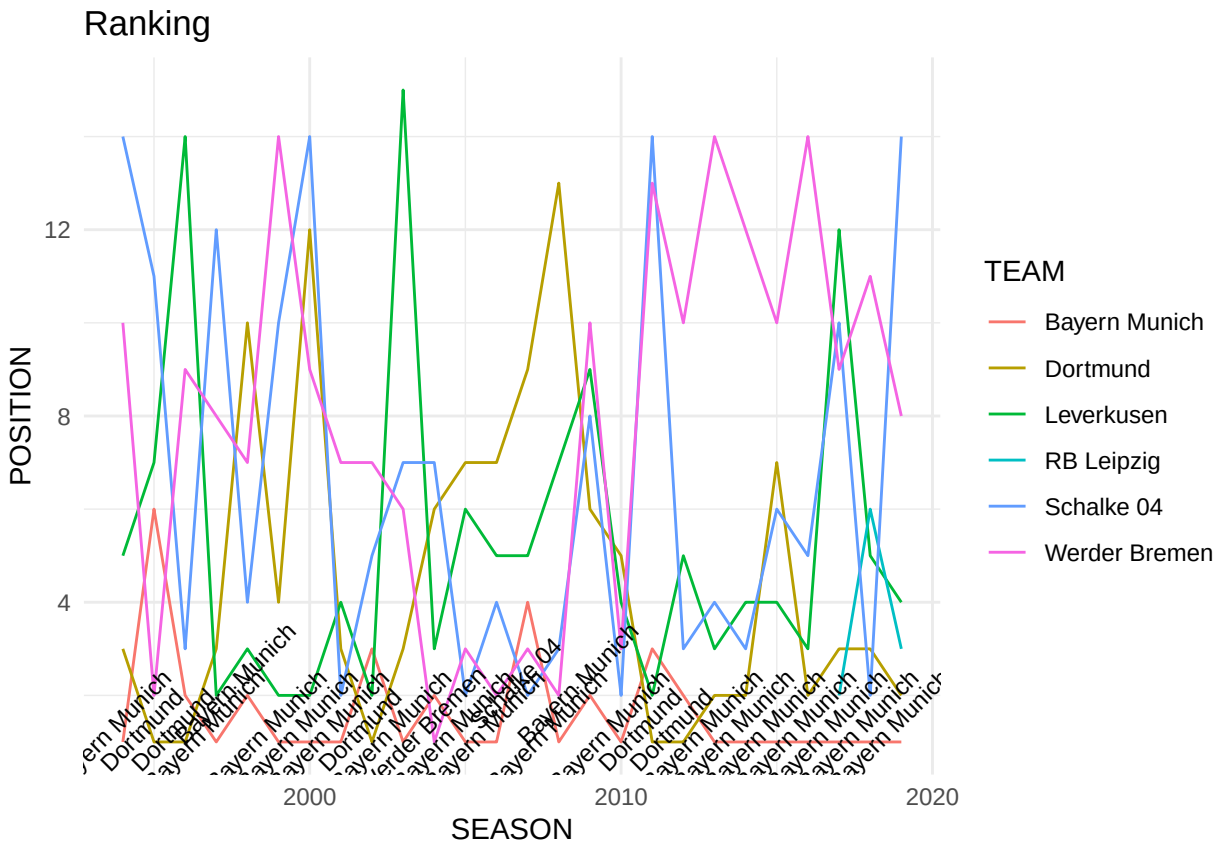


Part 3. 1)

```
top_6 <- bundesliga_cleaned %>%
  group_by(TEAM) %>%
  summarise(Rank = mean(POSITION)) %>%
  arrange(Rank)
top_6 <- head(top_6, 6)
```

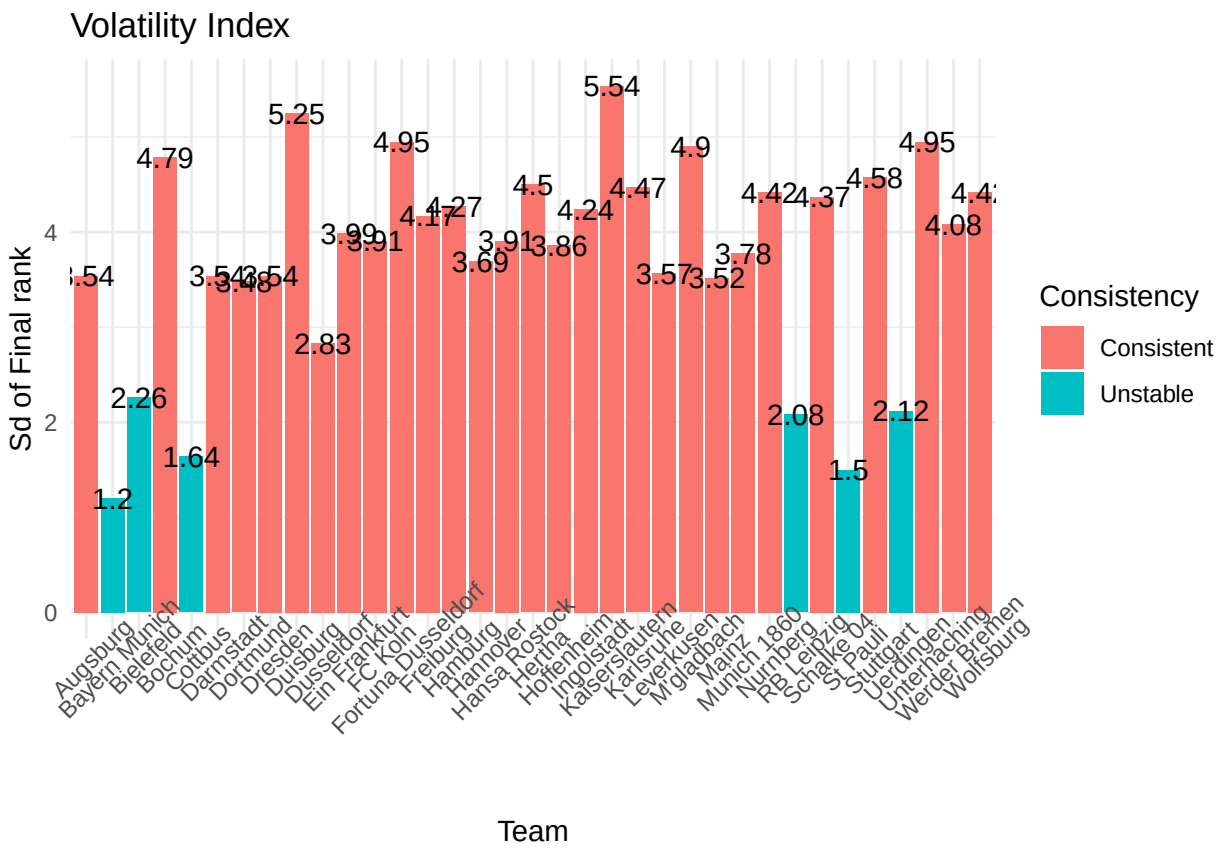
```
Trajectories <- bundesliga_cleaned %>%
  filter(TEAM %in% top_6$TEAM)

winners <- Trajectories %>%
  group_by(SEASON) %>%
  filter(POSITION == min(POSITION)) %>%
  ungroup()
```

```
ggplot(data = Trajectories, aes(x = SEASON, y = POSITION, colour = TEAM)) +
  geom_line() +
  geom_text(data = winners, aes(label = TEAM), vjust = -0.5, hjust = 0.5, size = 3, color = "black", ang
  theme_minimal() +
  labs(title = "Ranking")
```



2)

```
final_ranks <- bundesliga_cleaned %>%
  group_by(TEAM) %>%
  summarise(Final_rank = round(sd(POSITION),2)) %>%
  filter(!is.na(Final_rank)) %>%
  arrange(desc(Final_rank)) %>%
  mutate(Consistency = case_when(
    Final_rank < 2.5 ~ "Unstable",
    Final_rank >= 2.5 ~ "Consistent"
  ))

ggplot(data = final_ranks, aes(x = TEAM, y = Final_rank, fill = Consistency)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Final_rank)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title = "Volatility Index", x = "Team", y = "Sd of Final rank")
```

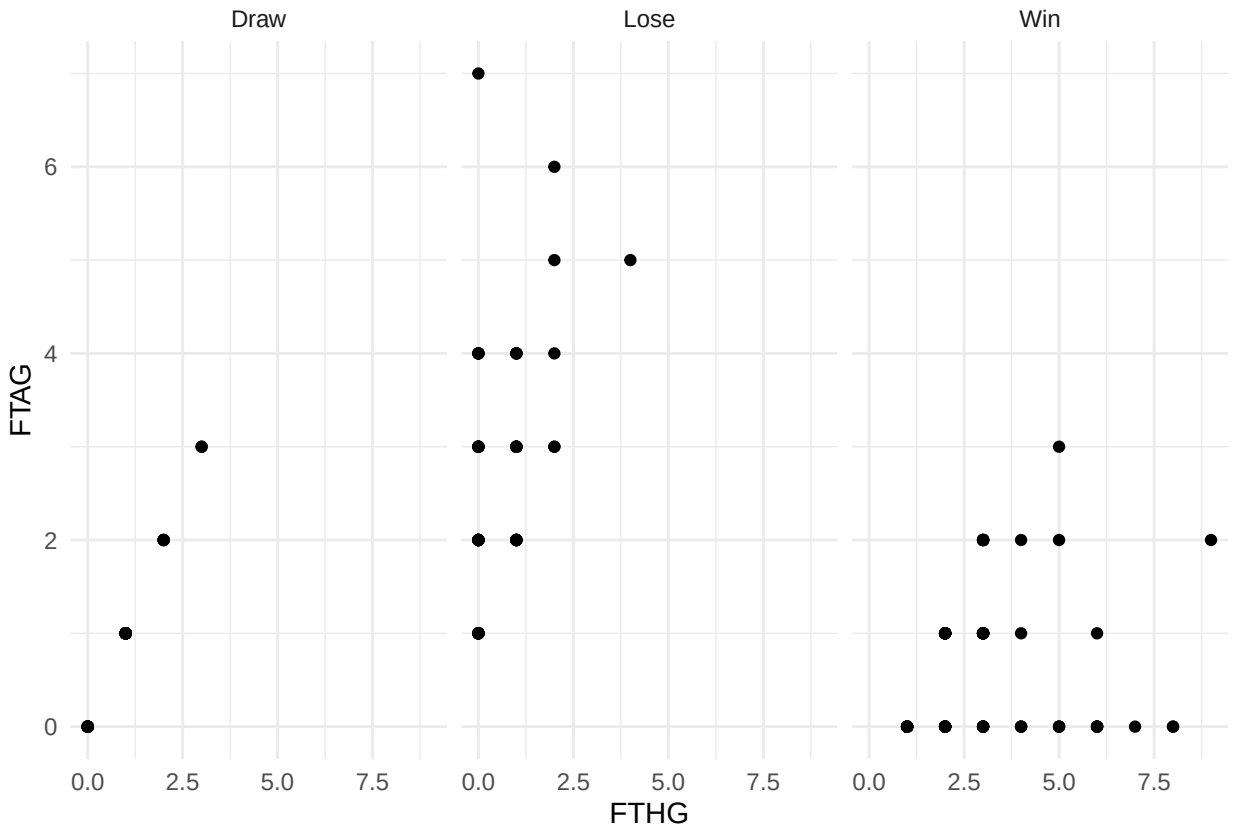


Part 4. 1) Hamburg and Werder Bremen Cologne and Borussia Mönchengladbach Bayern Munich and Borussia Mönchengladbach Borussia Dortmund and Schalke Bayern Munich and Borussia Dortmund

```
filt <- c("Bayern Munich", "Borussia Dortmund", "Borussia", "Schalke",
          "Borussia Mönchengladbach", "Cologne", "Hamburg",
          "Werder Bremen", "Bayern", "RB Leipzig", "Hertha Berlin",
          "Union Berlin", "St. Pauli", "VfB Stuttgart", "Karlsruhe")
```

```
part4 <- bundesliga %>%
  filter(HOMETEAM %in% filt) %>%
  filter(AWAYTEAM %in% filt) %>%
  mutate(Result = case_when(
    FTHG > FTAG ~ "Win",
    FTHG < FTAG ~ "Lose",
    FTHG == FTAG ~ "Draw"
  ))
```

```
ggplot(data = part4, aes(x = FTHG, y = FTAG)) +
  geom_point() +
  facet_grid(~Result) +
  theme_minimal()
```


Draw
Lose
Win
FTAG
FTHG
0
2
4
6
0.0
2.5
5.0
7.5