Norayr_Sukiasyan_HW4

Norayr Sukiasyan

2025-03-30

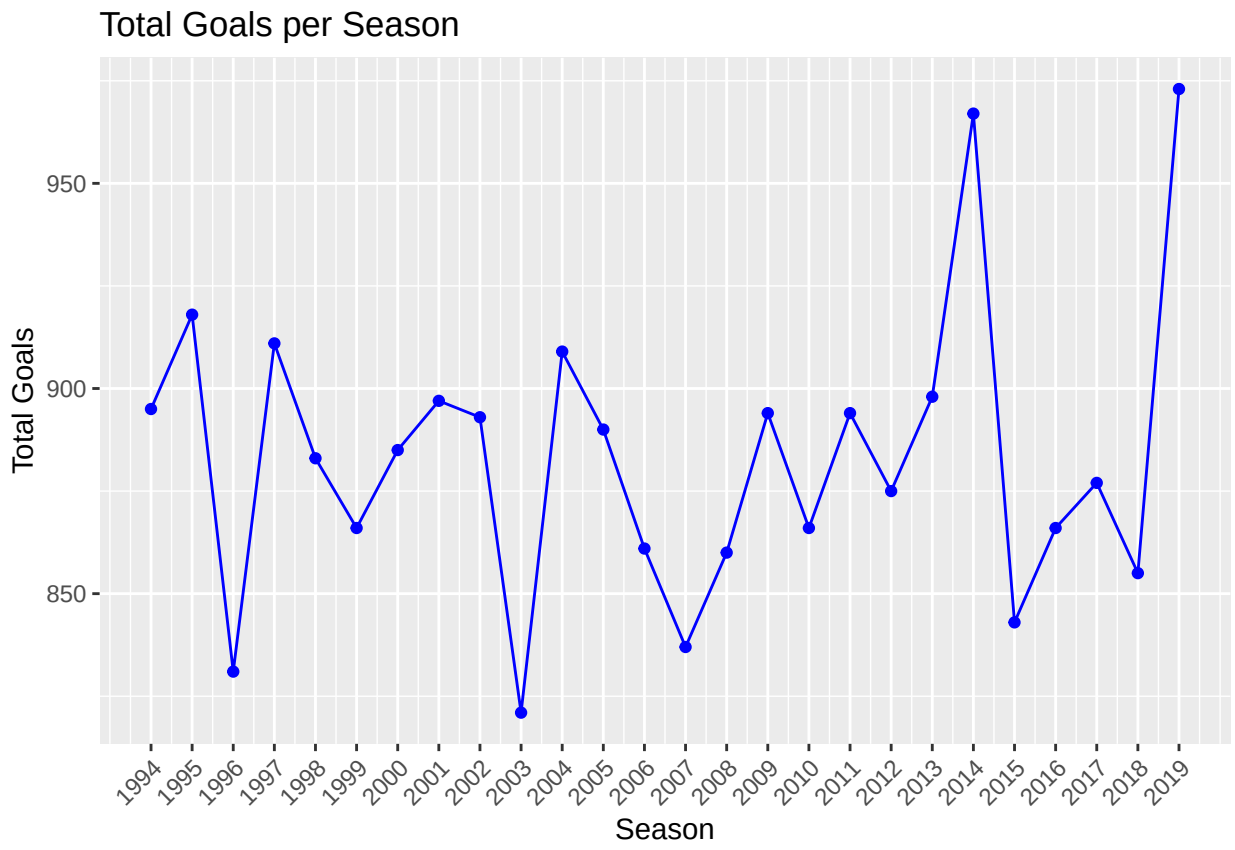
```
bundesliga <- read.csv("bundesliga.csv", stringsAsFactors = FALSE)
bundesliga2 <- read.csv("bundesliga2.csv", stringsAsFactors = FALSE)

df <- bind_rows(bundesliga, bundesliga2)

df$DATE <- as.Date(df$DATE, format = "%Y-%m-%d")

df <- df %>%
  mutate(TotalGoals = FTHG + FTAG)
```

Task 1

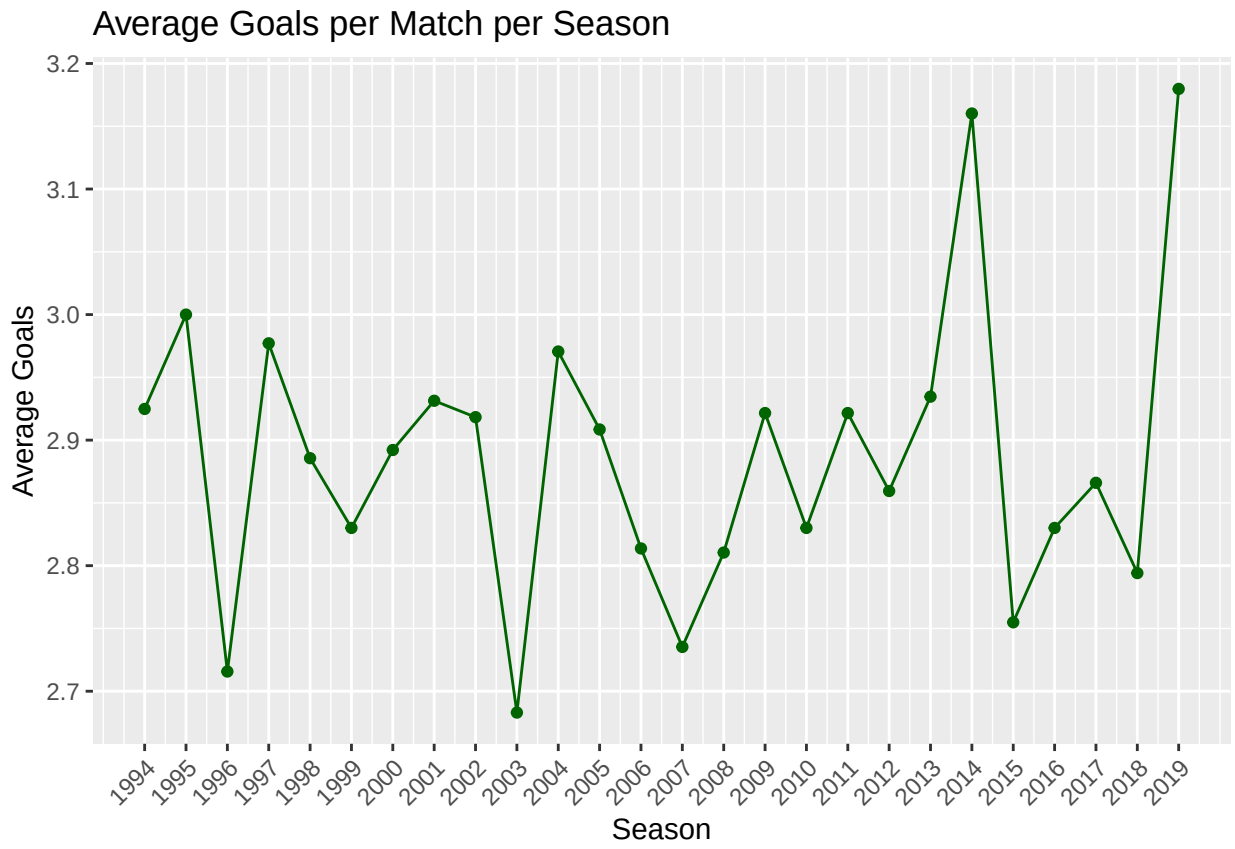
```
goals_by_season <- df %>%
  group_by(SEASON) %>%
  summarize(
    TotalGoalsSeason = sum(TotalGoals, na.rm = TRUE),
    Matches = n(),
    AvgGoalsPerMatch = mean(TotalGoals, na.rm = TRUE)
  ) %>%
  ungroup()

goals_by_season
```

```
## # A tibble: 26 x 4
##    SEASON TotalGoalsSeason Matches AvgGoalsPerMatch
##     <int>            <int>   <int>            <dbl>
##  1   1994              895     324             2.92
##  2   1995              918     324             3
##  3   1996              831     324             2.72
##  4   1997              911     324             2.98
##  5   1998              883     324             2.89
##  6   1999              866     324             2.83
##  7   2000              885     324             2.89
##  8   2001              897     324             2.93
##  9   2002              893     324             2.92
## 10   2003              821     324             2.68
## # i 16 more rows
```

```r
goals_by_season <- goals_by_season %>%
  mutate(SeasonOrder = as.numeric(str_sub(SEASON, 1, 4))) %>%
  arrange(SeasonOrder)

ggplot(goals_by_season, aes(x = SeasonOrder, y = TotalGoalsSeason)) +
  geom_line(color = "blue") +
  geom_point(color = "blue") +
  scale_x_continuous(breaks = goals_by_season$SeasonOrder,
                     labels = goals_by_season$SEASON) +
  labs(title = "Total Goals per Season",
       x = "Season",
       y = "Total Goals") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

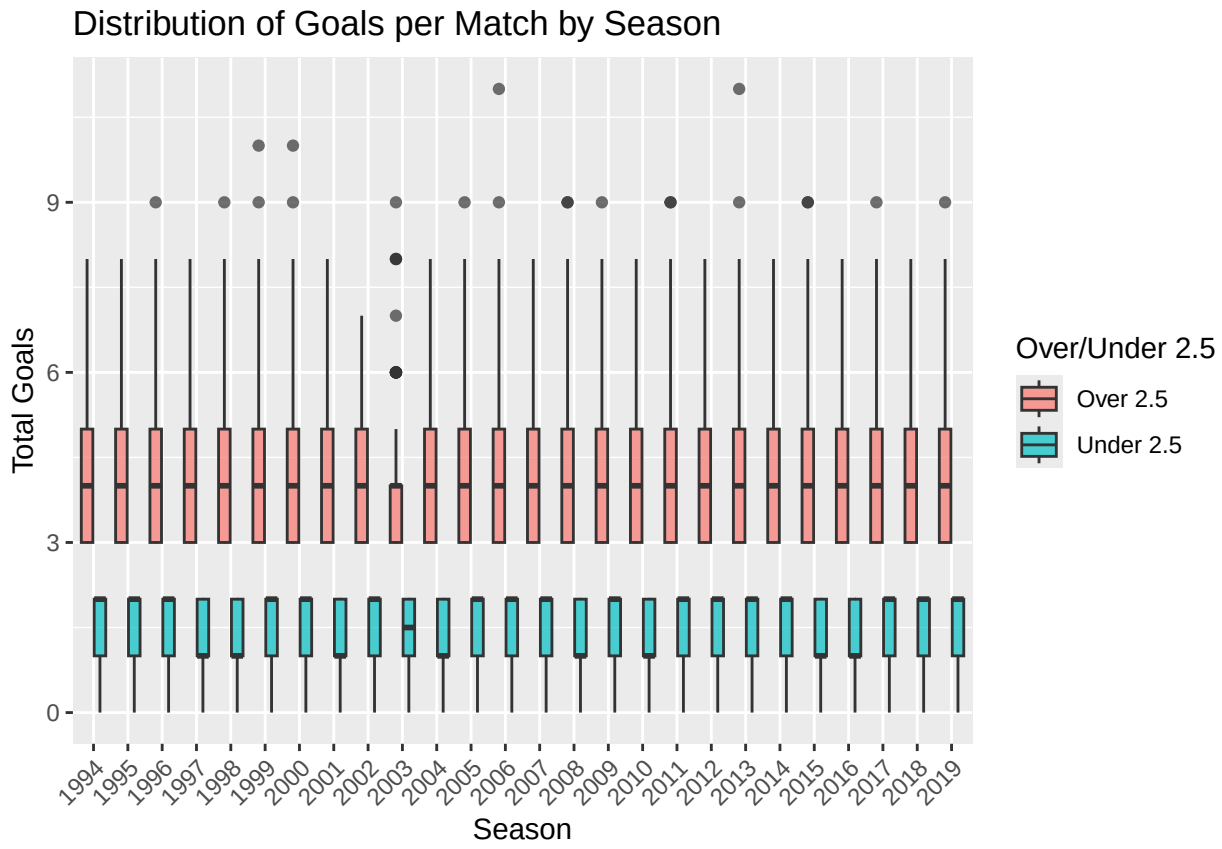



```r
ggplot(goals_by_season, aes(x = SeasonOrder, y = AvgGoalsPerMatch)) +
  geom_line(color = "darkgreen") +
  geom_point(color = "darkgreen") +
  scale_x_continuous(breaks = goals_by_season$SeasonOrder,
                     labels = goals_by_season$SEASON) +
  labs(title = "Average Goals per Match per Season",
       x = "Season",
       y = "Average Goals") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



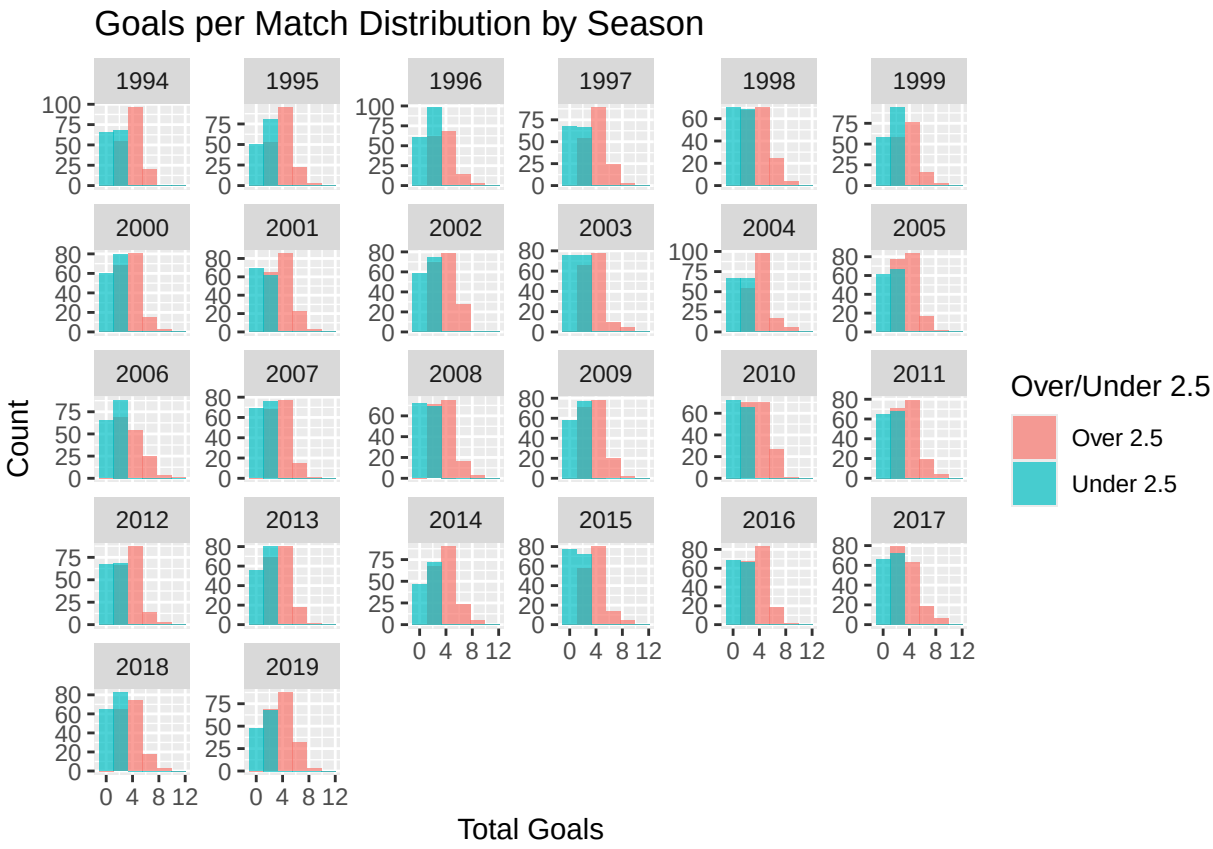

```
df <- df %>%
  mutate(OverUnder2.5 = if_else(TotalGoals > 2.5, "Over 2.5", "Under 2.5"),
         SeasonOrder = as.numeric(str_sub(SEASON, 1, 4)))

df$SEASON <- reorder(df$SEASON, df$SeasonOrder)

ggplot(df, aes(x = SEASON, y = TotalGoals, fill = OverUnder2.5)) +
  geom_boxplot(alpha = 0.7) +
  labs(title = "Distribution of Goals per Match by Season",
       x = "Season",
       y = "Total Goals",
       fill = "Over/Under 2.5") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 468 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
ggplot(df, aes(x = TotalGoals, fill = OverUnder2.5)) +
  geom_histogram(bins = 6, alpha = 0.7, position = "identity") +
  facet_wrap(~ SEASON, scales = "free_y") +
  labs(title = "Goals per Match Distribution by Season",
       x = "Total Goals",
       y = "Count",
       fill = "Over/Under 2.5")
```

```
## Warning: Removed 468 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Task 1.3

```r
df_home <- df %>%
  select(SEASON, DATE, HOMETEAM, FTHG) %>%
  rename(Team = HOMETEAM, Goals = FTHG)

df_away <- df %>%
  select(SEASON, DATE, AWAYTEAM, FTAG) %>%
  rename(Team = AWAYTEAM, Goals = FTAG)

df_long <- bind_rows(df_home, df_away)

df_long <- df_long %>%
  group_by(SEASON) %>%
  arrange(DATE, .by_group = TRUE) %>%
  mutate(MatchNumber = row_number()) %>%
  ungroup()

df_long <- df_long %>%
  group_by(SEASON, Team) %>%
  arrange(MatchNumber, .by_group = TRUE) %>%
  mutate(CumulativeGoals = cumsum(Goals)) %>%
  ungroup()
```

```r
all_seasons <- unique(df_long$SEASON)

pdf("Task_1.3_Line_Charts_By_Season.pdf", width = 10, height = 6)

for(s in all_seasons){

  season_data <- df_long %>% filter(SEASON == s)

  total_goals_season <- df %>%
    filter(SEASON == s) %>%
    summarize(TotalGoalsSeason = sum(TotalGoals, na.rm = TRUE)) %>%
    pull(TotalGoalsSeason)

  bayern_goals <- season_data %>%
    filter(Team == "Bayern Munich") %>%
    summarize(BayernTotal = sum(Goals, na.rm = TRUE)) %>%
    pull(BayernTotal)

  season_data <- season_data %>%
    mutate(TeamColor = if_else(Team == "Bayern Munich", "Bayern", "Other"))

  p <- ggplot(season_data, aes(x = MatchNumber, y = CumulativeGoals, group = Team, color = TeamColor)) +
    geom_line(size = 1) +
    scale_color_manual(values = c("Bayern" = "red", "Other" = "grey60")) +
    labs(title = paste0("Season: ", s, " | Total Goals: ", total_goals_season),
         x = "Match Number",
         y = "Cumulative Goals",
         color = "Team") +
    labs(caption = paste("Bayern Munich total goals this season:", bayern_goals)) +
    theme_minimal()

  print(p)

}
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 36 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```

```
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 36 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
dev.off()
```

```
## pdf
##   2
```

Task 2.1

```
df_home_wins <- df %>%
  filter(FTHG > FTAG) %>%
  group_by(SEASON, HOMETEAM) %>%
```

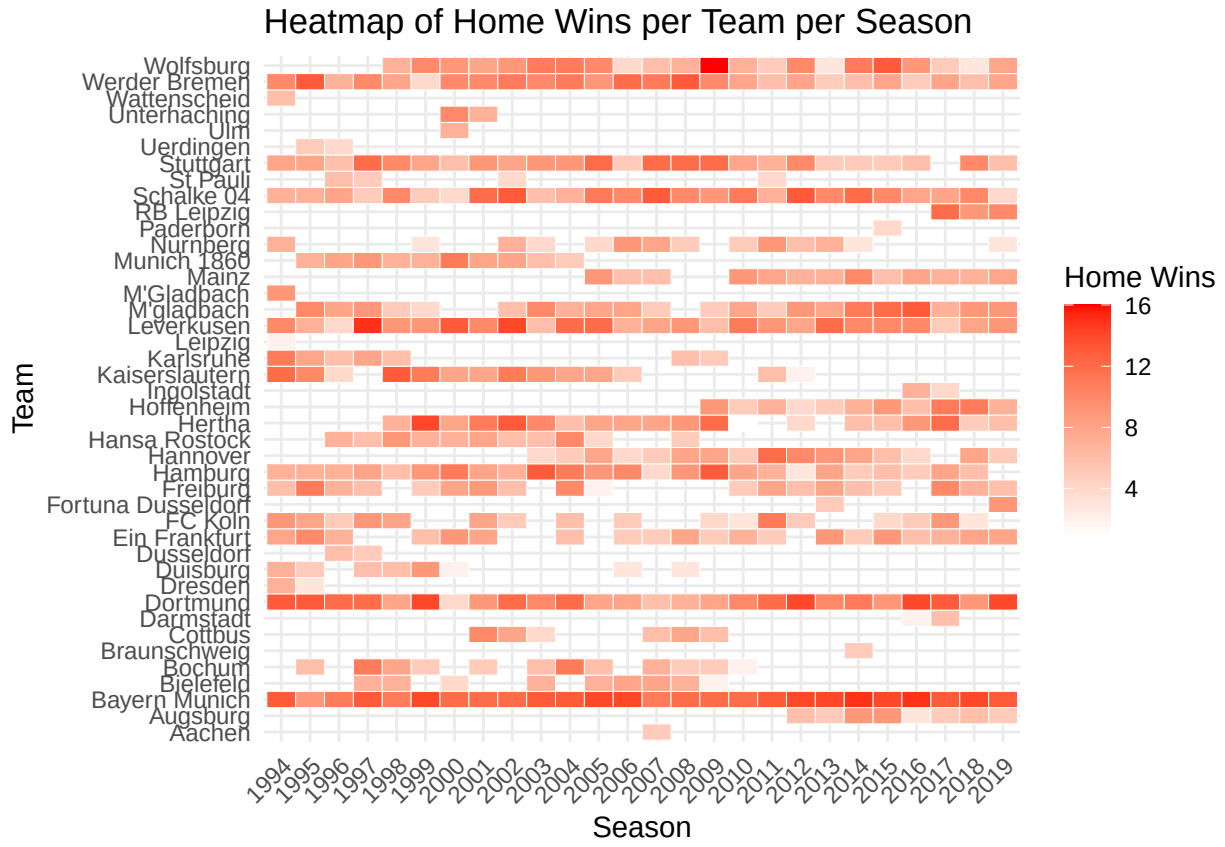
```
  summarise(HomeWins = n(), .groups = "drop") %>%
  rename(Team = HOMETEAM)

df_away_wins <- df %>%
  filter(FTAG > FTHG) %>%
  group_by(SEASON, AWAYTEAM) %>%
  summarise(AwayWins = n(), .groups = "drop") %>%
  rename(Team = AWAYTEAM)

head(df_home_wins)
```

```
## # A tibble: 6 x 3
##   SEASON Team          HomeWins
##   <fct>  <chr>            <int>
## 1 1994   Bayern Munich       13
## 2 1994   Dortmund            13
## 3 1994   Dresden              7
## 4 1994   Duisburg             7
## 5 1994   Ein Frankfurt        8
## 6 1994   FC Koln              9
```

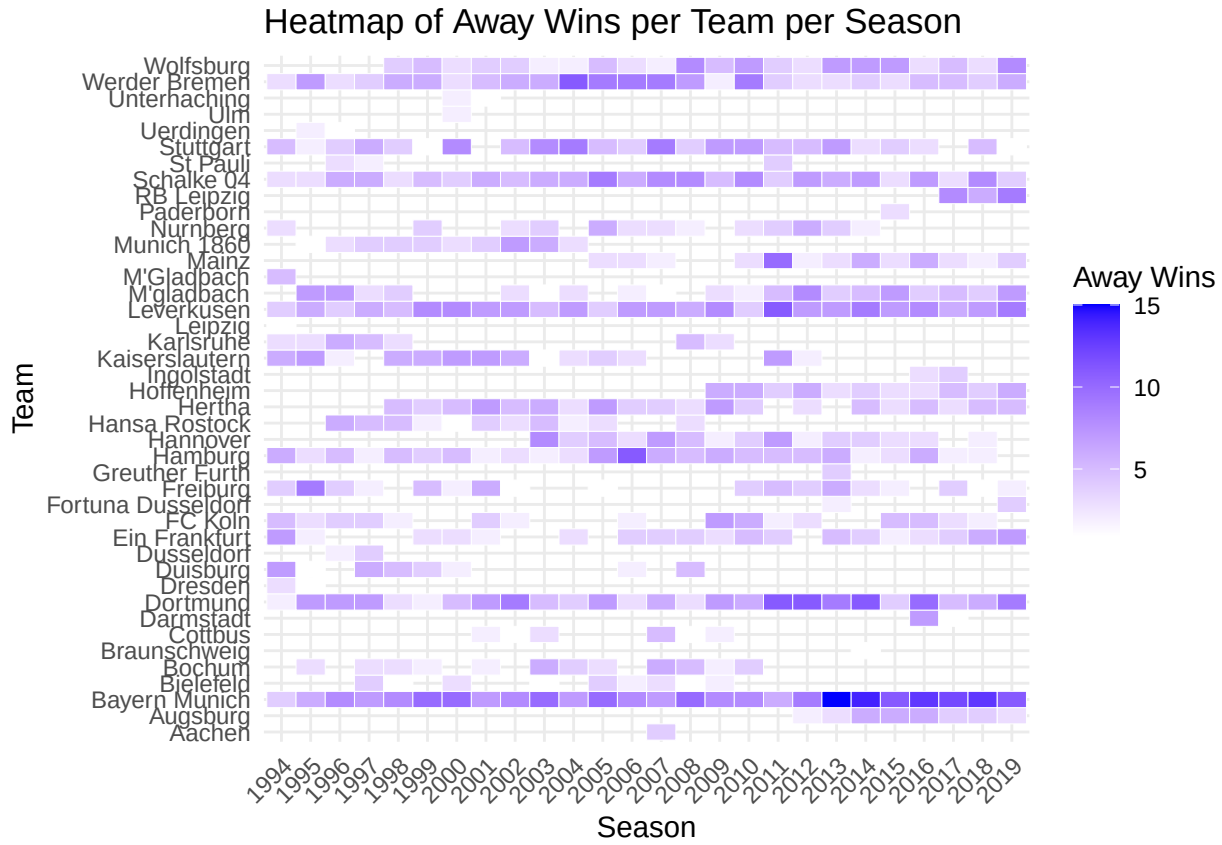
```
head(df_away_wins)
```

```
## # A tibble: 6 x 3
##   SEASON Team          AwayWins
##   <fct>  <chr>            <int>
## 1 1994   Bayern Munich        4
## 2 1994   Dortmund             2
## 3 1994   Dresden              3
## 4 1994   Duisburg             7
## 5 1994   Ein Frankfurt        7
## 6 1994   FC Koln              5
```

```
ggplot(df_home_wins, aes(x = SEASON, y = Team, fill = HomeWins)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Heatmap of Home Wins per Team per Season",
       x = "Season",
       y = "Team",
       fill = "Home Wins") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
ggplot(df_away_wins, aes(x = SEASON, y = Team, fill = AwayWins)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Heatmap of Away Wins per Team per Season",
       x = "Season",
       y = "Team",
       fill = "Away Wins") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Task 2.2

```
df_win_margins <- df %>%
  filter(FTHG != FTAG) %>%
  mutate(
    Team = if_else(FTHG > FTAG, HOMETEAM, AWAYTEAM),
    Location = if_else(FTHG > FTAG, "Home", "Away"),
    MarginOfVictory = if_else(FTHG > FTAG, FTHG - FTAG, FTAG - FTHG)
  ) %>%
  select(SEASON, DATE, Team, Location, MarginOfVictory)

head(df_win_margins)
```

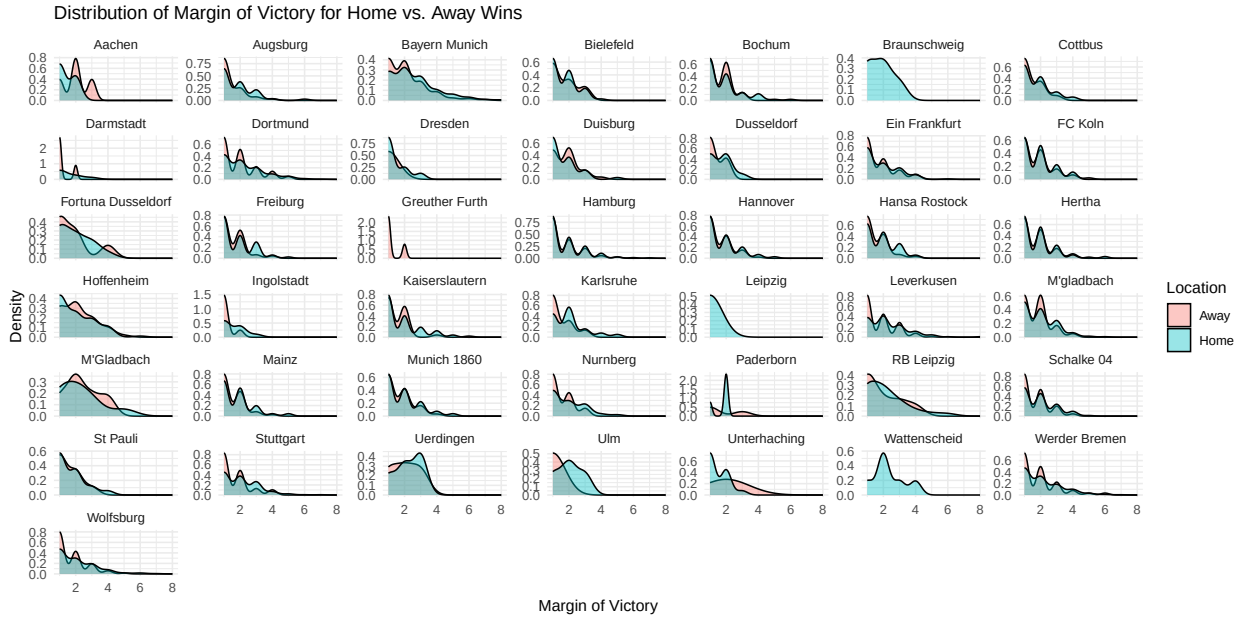
```
##   SEASON       DATE           Team Location MarginOfVictory
## 1   1994 1993-08-07  Bayern Munich     Home               2
## 2   1994 1993-08-07       Dortmund     Home               1
## 3   1994 1993-08-07 Kaiserslautern     Away               2
## 4   1994 1993-08-07        Hamburg     Home               3
## 5   1994 1993-08-07  Ein Frankfurt     Away               4
## 6   1994 1993-08-07   Wattenscheid     Home               3
```

```
ggplot(df_win_margins, aes(x = MarginOfVictory, fill = Location)) +
  geom_density(alpha = 0.4) +
```

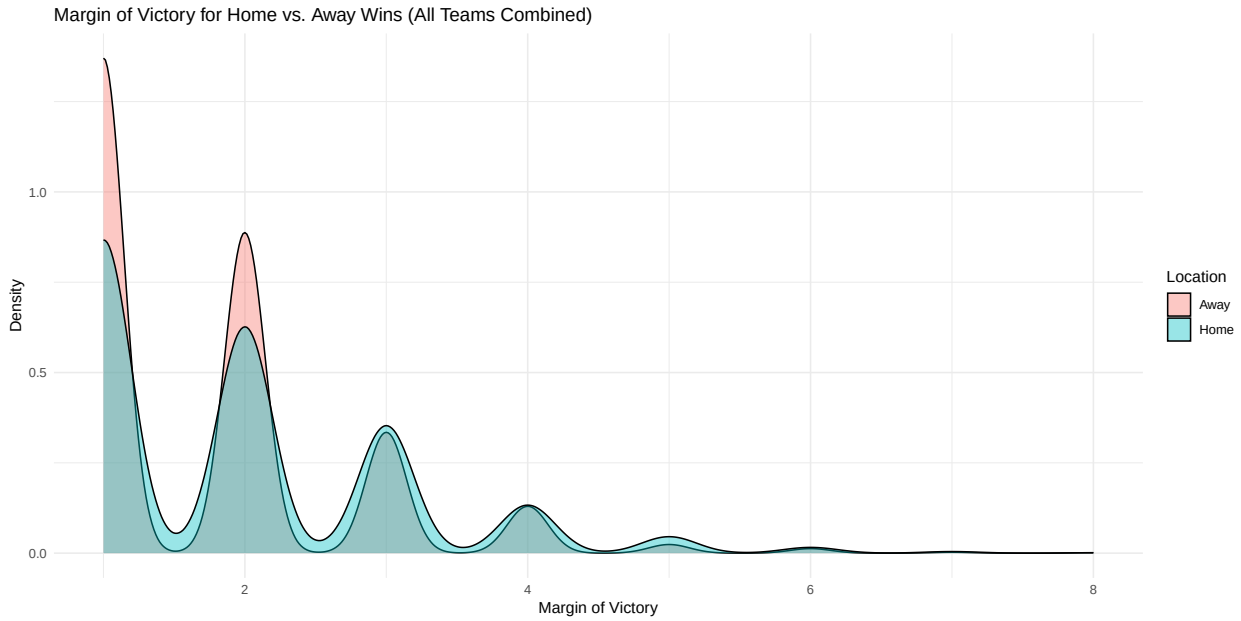
```
  facet_wrap(~ Team, scales = "free_y") +
  labs(title = "Distribution of Margin of Victory for Home vs. Away Wins",
       x = "Margin of Victory",
       y = "Density",
       fill = "Location") +
  theme_minimal()
```

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```




```
ggplot(df_win_margins, aes(x = MarginOfVictory, fill = Location)) +
  geom_density(alpha = 0.4) +
  labs(title = "Margin of Victory for Home vs. Away Wins (All Teams Combined)",
       x = "Margin of Victory",
       y = "Density",
       fill = "Location") +
  theme_minimal()
```



Task 3.1

```
df_points <- df %>%
  mutate(
    HomePoints = case_when(FTHG > FTAG ~ 3,
                           FTHG == FTAG ~ 1,
                           TRUE ~ 0),
    AwayPoints = case_when(FTAG > FTHG ~ 3,
                           FTAG == FTHG ~ 1,
                           TRUE ~ 0),
    HomeGF = FTHG,
    HomeGA = FTAG,
    AwayGF = FTAG,
    AwayGA = FTHG
  )

home_summary <- df_points %>%
  group_by(SEASON, HOMETEAM) %>%
  summarize(
    Points = sum(HomePoints),
    GF = sum(HomeGF),
    GA = sum(HomeGA),
    .groups = "drop"
  ) %>%
  rename(Team = HOMETEAM)

away_summary <- df_points %>%
  group_by(SEASON, AWAYTEAM) %>%
  summarize(
    Points = sum(AwayPoints),
    GF = sum(AwayGF),
```

```
    GA = sum(AwayGA),
    .groups = "drop"
  ) %>%
  rename(Team = AWAYTEAM)

season_table <- bind_rows(home_summary, away_summary) %>%
  group_by(SEASON, Team) %>%
  summarize(
    Points = sum(Points),
    GF = sum(GF),
    GA = sum(GA),
    .groups = "drop"
  ) %>%
  mutate(GD = GF - GA)

season_table <- season_table %>%
  group_by(SEASON) %>%
  arrange(desc(Points), desc(GD), desc(GF), .by_group = TRUE) %>%
  mutate(FinalRank = min_rank(-Points) + 0) %>%
  ungroup()

head(season_table)
```

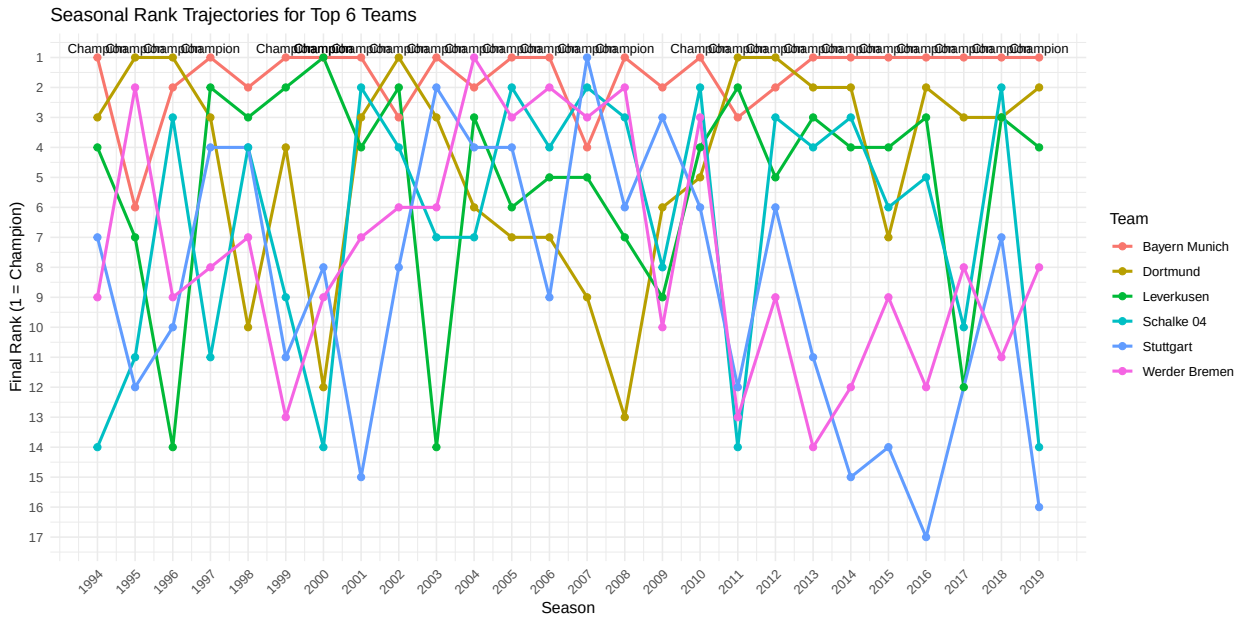
```
## # A tibble: 6 x 7
##   SEASON Team           Points    GF    GA    GD FinalRank
##   <fct>  <chr>           <dbl> <int> <int> <int>     <dbl>
## 1 1994   Bayern Munich      61    68    37    31         1
## 2 1994   Kaiserslautern     61    64    36    28         1
## 3 1994   Dortmund           54    49    45     4         3
## 4 1994   Ein Frankfurt      53    57    41    16         4
## 5 1994   Leverkusen         53    60    47    13         4
## 6 1994   Karlsruhe          52    46    43     3         6
```

```
team_totals <- season_table %>%
  group_by(Team) %>%
  summarize(TotalPointsAllSeasons = sum(Points), .groups = "drop") %>%
  arrange(desc(TotalPointsAllSeasons))

top_6 <- team_totals$Team[1:6]
top_6
```

```
## [1] "Bayern Munich" "Dortmund"      "Leverkusen"    "Schalke 04"
## [5] "Werder Bremen" "Stuttgart"
```

```
season_table_top6 <- season_table %>%
  filter(Team %in% top_6)

season_table_top6 <- season_table_top6 %>%
  mutate(Champion = if_else(FinalRank == 1, "Champion", NA_character_))

season_table_top6 <- season_table_top6 %>%
  mutate(SeasonNum = as.numeric(str_sub(SEASON, 1, 4)))  # e.g. "2018" from "2018/2019"
```

```r
ggplot(season_table_top6, aes(x = SeasonNum, y = FinalRank, color = Team, group = Team)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  scale_x_continuous(
    breaks = sort(unique(season_table_top6$SeasonNum)),
    labels = sort(unique(season_table_top6$SEASON))
  ) +
  scale_y_reverse(breaks = 1:max(season_table_top6$FinalRank)) +  # Reverse rank
  geom_text(
    data = subset(season_table_top6, !is.na(Champion)),
    aes(label = Champion),
    vjust = -0.5, size = 3, color = "black"
  ) +
  labs(title = "Seasonal Rank Trajectories for Top 6 Teams",
       x = "Season",
       y = "Final Rank (1 = Champion)",
       color = "Team") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



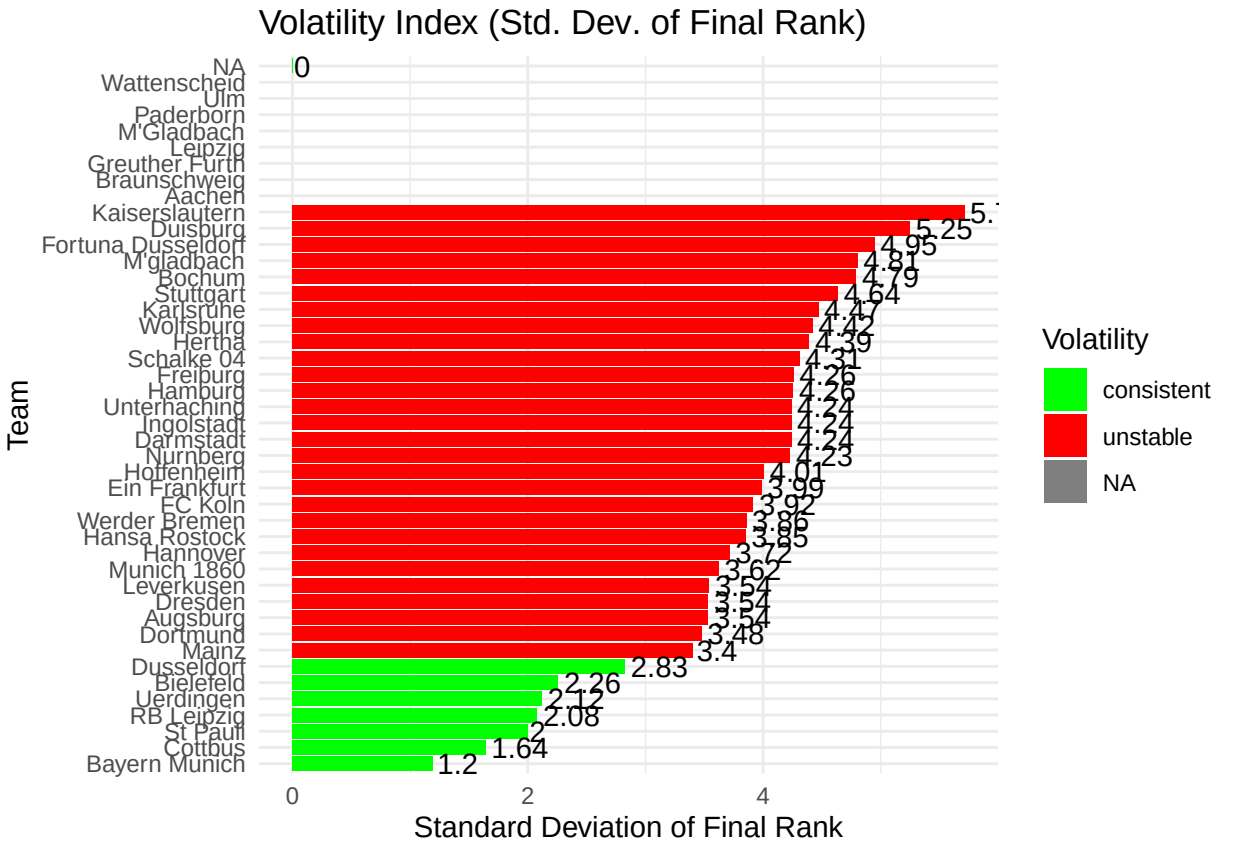
Task 3.2

```r
volatility <- season_table %>%
  group_by(Team) %>%
  summarize(
    SD_Rank = sd(FinalRank, na.rm = TRUE),
    .groups = "drop"
  )

volatility <- volatility %>%
```

```
  mutate(VolatilityType = if_else(SD_Rank > 3, "unstable", "consistent"))

ggplot(volatility, aes(x = reorder(Team, SD_Rank), y = SD_Rank, fill = VolatilityType)) +
  geom_col() +
  coord_flip() +  # Flip for easier reading (teams on the y-axis)
  scale_fill_manual(values = c("unstable" = "red", "consistent" = "green")) +
  geom_text(aes(label = round(SD_Rank, 2)), hjust = -0.1) +  # label slightly to the right
  labs(title = "Volatility Index (Std. Dev. of Final Rank)",
       x = "Team",
       y = "Standard Deviation of Final Rank",
       fill = "Volatility") +
  theme_minimal()
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_col()`).
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_text()`).
```



Task 4.1

```r
df_pairs <- df %>%
  mutate(
    Team1 = if_else(HOMETEAM < AWAYTEAM, HOMETEAM, AWAYTEAM),
    Team2 = if_else(HOMETEAM < AWAYTEAM, AWAYTEAM, HOMETEAM),
    PairLabel = paste(Team1, "vs.", Team2)
  )

pair_counts <- df_pairs %>%
  group_by(PairLabel) %>%
  summarise(MatchesPlayed = n(), .groups = "drop") %>%
  arrange(desc(MatchesPlayed))

top_5_pairs <- pair_counts$PairLabel[1:5]
top_5_pairs
```

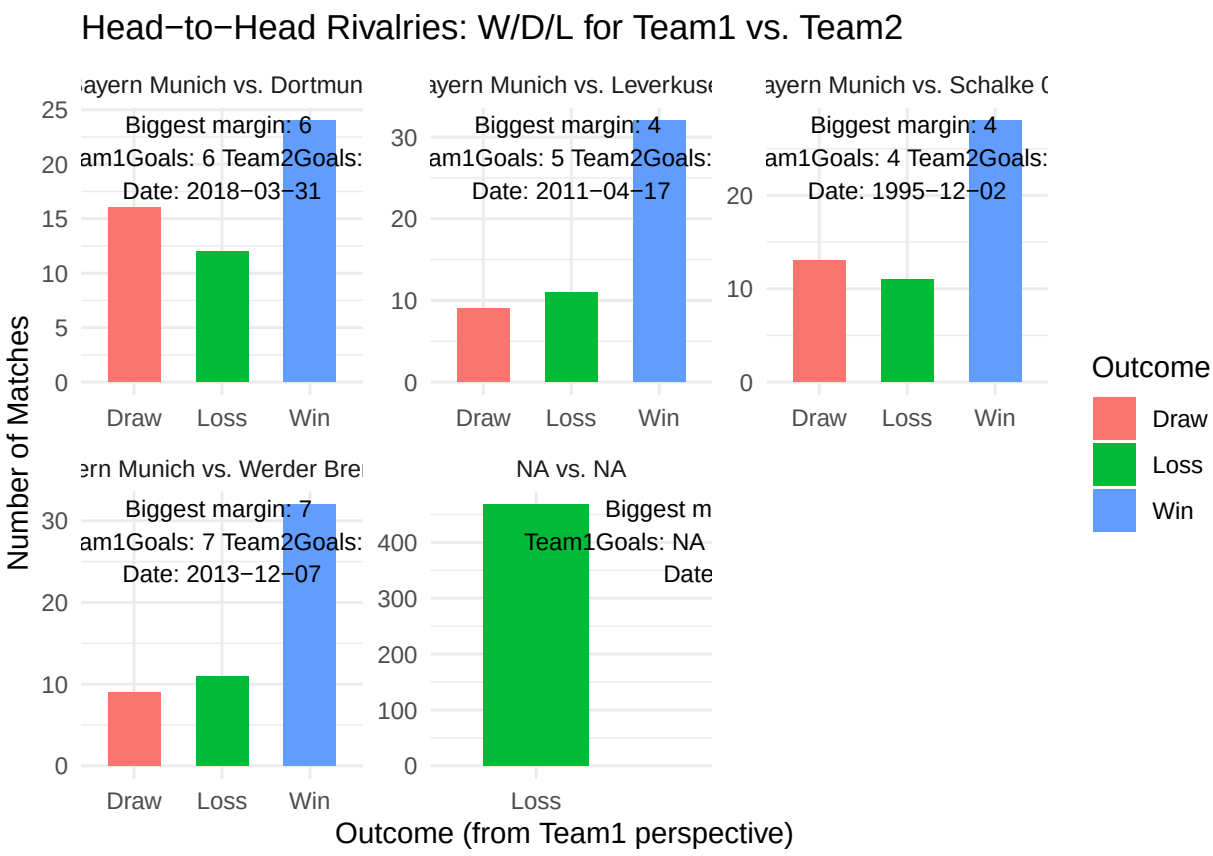
```
## [1] "NA vs. NA"                       "Bayern Munich vs. Dortmund"
## [3] "Bayern Munich vs. Leverkusen"    "Bayern Munich vs. Schalke 04"
## [5] "Bayern Munich vs. Werder Bremen"
```

```r
df_rivalries <- df_pairs %>%
  filter(PairLabel %in% top_5_pairs)

df_rivalries <- df_rivalries %>%
  mutate(
    IsTeam1Home = (Team1 == HOMETEAM),
    Team1Goals = if_else(IsTeam1Home, FTHG, FTAG),
    Team2Goals = if_else(IsTeam1Home, FTAG, FTHG),
    Outcome = case_when(
      Team1Goals > Team2Goals ~ "Win",
      Team1Goals == Team2Goals ~ "Draw",
      TRUE ~ "Loss"
    ),
    Margin = abs(Team1Goals - Team2Goals)
  )

rivalry_summary <- df_rivalries %>%
  group_by(PairLabel, Outcome) %>%
  summarise(Count = n(), .groups = "drop")

rivalry_summary
```

```
## # A tibble: 13 x 3
##    PairLabel                      Outcome Count
##    <chr>                          <chr>   <int>
##  1 Bayern Munich vs. Dortmund     Draw       16
##  2 Bayern Munich vs. Dortmund     Loss       12
##  3 Bayern Munich vs. Dortmund     Win        24
##  4 Bayern Munich vs. Leverkusen   Draw        9
##  5 Bayern Munich vs. Leverkusen   Loss       11
##  6 Bayern Munich vs. Leverkusen   Win        32
##  7 Bayern Munich vs. Schalke 04   Draw       13
##  8 Bayern Munich vs. Schalke 04   Loss       11
```

```
##  9 Bayern Munich vs. Schalke 04    Win        28
## 10 Bayern Munich vs. Werder Bremen Draw        9
## 11 Bayern Munich vs. Werder Bremen Loss       11
## 12 Bayern Munich vs. Werder Bremen Win        32
## 13 NA vs. NA                       Loss      468
```

```r
biggest_margin <- df_rivalries %>%
  group_by(PairLabel) %>%
  slice_max(Margin, n = 1, with_ties = FALSE) %>%
  ungroup()

ggplot(rivalry_summary, aes(x = Outcome, y = Count, fill = Outcome)) +
  geom_bar(stat = "identity", width = 0.6) +
  facet_wrap(~ PairLabel, scales = "free") +
  labs(title = "Head-to-Head Rivalries: W/D/L for Team1 vs. Team2",
       x = "Outcome (from Team1 perspective)",
       y = "Number of Matches") +
  theme_minimal() +
  geom_text(
    data = biggest_margin,
    aes(
      x = 2,
      y = Inf,
      label = paste("Biggest margin:", Margin,
                    "\nTeam1Goals:", Team1Goals,
                    "Team2Goals:", Team2Goals,
                    "\nDate:", DATE)
    ),
    vjust = 1.1,
    color = "black",
    size = 3
  )
```



Task 4.2

```
team_totals <- season_table %>%
  group_by(Team) %>%
  summarize(TotalPointsAllSeasons = sum(Points), .groups = "drop") %>%
  arrange(desc(TotalPointsAllSeasons))

top_5_teams <- team_totals$Team[1:5]
top_5_teams
```

```
## [1] "Bayern Munich" "Dortmund"      "Leverkusen"    "Schalke 04"
## [5] "Werder Bremen"
```

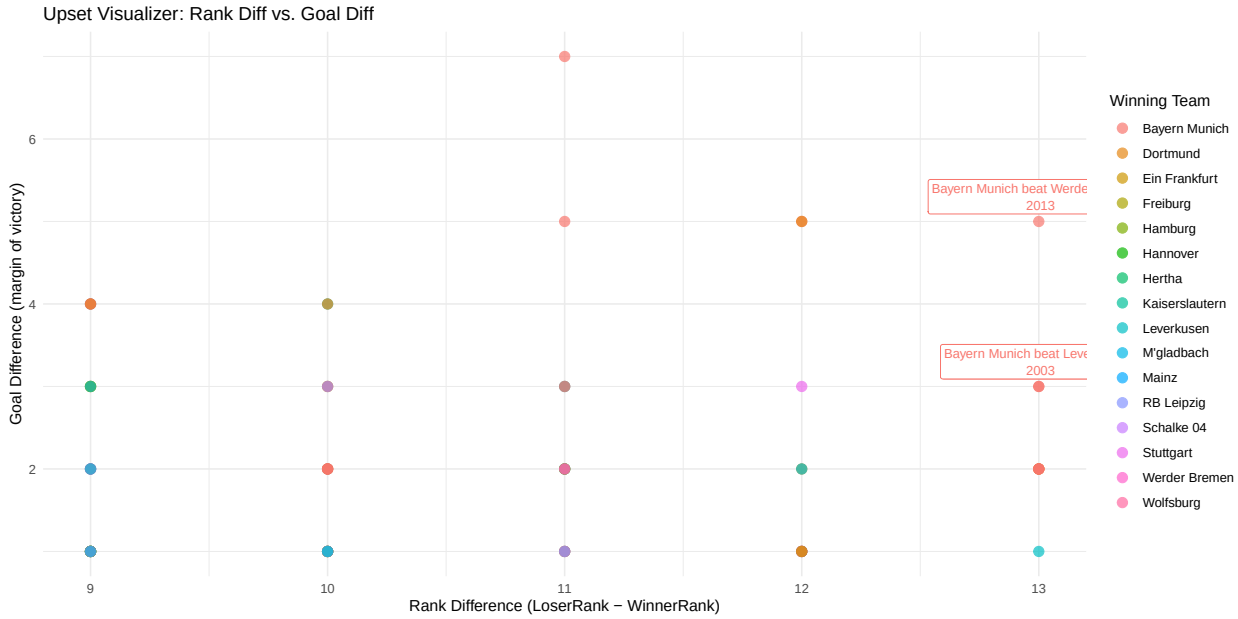
```
df_outcomes <- df %>%
  mutate(
    IsDraw = (FTHG == FTAG),
    Winner = case_when(
      FTHG > FTAG ~ HOMETEAM,
      FTAG > FTHG ~ AWAYTEAM,
      TRUE ~ NA_character_
    ),
    Loser = case_when(
      FTHG > FTAG ~ AWAYTEAM,
```

```
      FTAG > FTHG ~ HOMETEAM,
      TRUE ~ NA_character_
    ),
    GoalDiff = abs(FTHG - FTAG)
  ) %>%
  filter(!IsDraw)

df_outcomes <- df_outcomes %>%
  left_join(
    season_table %>% select(SEASON, Team, FinalRank),
    by = c("SEASON" = "SEASON", "Winner" = "Team")
  ) %>%
  rename(WinnerRank = FinalRank)

df_outcomes <- df_outcomes %>%
  left_join(
    season_table %>% select(SEASON, Team, FinalRank),
    by = c("SEASON" = "SEASON", "Loser" = "Team")
  ) %>%
  rename(LoserRank = FinalRank)

df_outcomes <- df_outcomes %>%
  mutate(
    RankDifference = LoserRank - WinnerRank,
    IsUpset = (RankDifference >= 9 & Loser %in% top_5_teams)
  )

head(df_outcomes)
```

```
##   SEASON       LEAGUE       DATE      HOMETEAM       AWAYTEAM FTSC FTHG FTAG
## 1   1994 Bundesliga 1 1993-08-07 Bayern Munich       Freiburg  3-1    3    1
## 2   1994 Bundesliga 1 1993-08-07      Dortmund      Karlsruhe  2-1    2    1
## 3   1994 Bundesliga 1 1993-08-07       FC Koln Kaiserslautern  0-2    0    2
## 4   1994 Bundesliga 1 1993-08-07       Hamburg       Nurnberg  5-2    5    2
## 5   1994 Bundesliga 1 1993-08-07    M'Gladbach  Ein Frankfurt  0-4    0    4
## 6   1994 Bundesliga 1 1993-08-07  Wattenscheid     Schalke 04  3-0    3    0
##   FTTG TEAM  M  W  D  L GF GA DIFF POINTS POSITION TotalGoals OverUnder2.5
## 1    4 <NA> NA NA NA NA NA NA   NA     NA       NA          4     Over 2.5
## 2    3 <NA> NA NA NA NA NA NA   NA     NA       NA          3     Over 2.5
## 3    2 <NA> NA NA NA NA NA NA   NA     NA       NA          2    Under 2.5
## 4    7 <NA> NA NA NA NA NA NA   NA     NA       NA          7     Over 2.5
## 5    4 <NA> NA NA NA NA NA NA   NA     NA       NA          4     Over 2.5
## 6    3 <NA> NA NA NA NA NA NA   NA     NA       NA          3     Over 2.5
##   SeasonOrder IsDraw         Winner      Loser GoalDiff WinnerRank LoserRank
## 1        1994  FALSE  Bayern Munich   Freiburg        2          1        15
## 2        1994  FALSE       Dortmund  Karlsruhe        1          3         6
## 3        1994  FALSE Kaiserslautern    FC Koln        2          1        11
## 4        1994  FALSE        Hamburg   Nurnberg        3         12        15
## 5        1994  FALSE  Ein Frankfurt M'Gladbach        4          4         9
## 6        1994  FALSE   Wattenscheid Schalke 04        3         17        14
##   RankDifference IsUpset
## 1             14   FALSE
## 2              3   FALSE
```

```
## 3             10   FALSE
## 4              3   FALSE
## 5              5   FALSE
## 6             -3   FALSE
```

```r
df_upsets <- df_outcomes %>% filter(IsUpset)

highlight_upsets <- df_upsets %>%
  arrange(desc(RankDifference), desc(GoalDiff)) %>%
  slice_head(n = 3)

ggplot(df_upsets, aes(x = RankDifference, y = GoalDiff, color = Winner)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Upset Visualizer: Rank Diff vs. Goal Diff",
       x = "Rank Difference (LoserRank - WinnerRank)",
       y = "Goal Difference (margin of victory)",
       color = "Winning Team") +
  theme_minimal() +
  geom_label(
    data = highlight_upsets,
    aes(label = paste(Winner, "beat", Loser, "\n", SEASON)),
    size = 3, nudge_y = 0.3, show.legend = FALSE
  )
```



Task 5

```r
all_seasons <- unique(season_table$SEASON)

season_colors <- list()

for (s in all_seasons) {
```

```
  season_data <- subset(season_table, SEASON == s)

  teams_in_season <- unique(season_data$Team)
  n_teams <- length(teams_in_season)

  palette_for_season <- brewer.pal(max(3, min(n_teams, 12)), "Set3")
  if (n_teams > 12) {
    palette_for_season <- colorRampPalette(palette_for_season)(n_teams)
  }

  teams_in_season_sorted <- sort(teams_in_season)

  color_vector <- setNames(palette_for_season[seq_along(teams_in_season_sorted)],
                           teams_in_season_sorted)

  season_colors[[s]] <- color_vector
}
```

```
pdf("Task_5_Season_Points_Bars.pdf", width = 8, height = 6)

for(s in all_seasons){

  season_data <- subset(season_table, SEASON == s)

  season_data <- season_data %>%
    arrange(desc(Points))

  color_map <- season_colors[[s]]

  season_data$Team <- factor(season_data$Team, levels = season_data$Team)

  p <- ggplot(season_data, aes(x = Points, y = Team)) +
    geom_col(aes(fill = Team), width = 0.6) +
    scale_fill_manual(values = color_map) +
    labs(title = paste("Season:", s, "- Points by Team"),
         x = "Points",
         y = "Team",
         fill = "Team") +
    theme_minimal() +
    theme(legend.position = "none")

  print(p)

}

dev.off()
```

```
## pdf
##   2
```

Task 6

```
df1 <- read.csv("bundesliga.csv", stringsAsFactors = FALSE)
df2 <- read.csv("bundesliga2.csv", stringsAsFactors = FALSE)

df <- bind_rows(df1, df2)

if("DATE" %in% names(df)) {
  df$DATE <- as.Date(df$DATE, format = "%Y-%m-%d")
}

names(df)
```

```
##  [1] "SEASON"   "LEAGUE"   "DATE"     "HOMETEAM" "AWAYTEAM" "FTSC"
##  [7] "FTHG"     "FTAG"     "FTTG"     "TEAM"     "M"        "W"
## [13] "D"        "L"        "GF"       "GA"       "DIFF"     "POINTS"
## [19] "POSITION"
```

```
head(df)
```

```
##   SEASON       LEAGUE       DATE      HOMETEAM       AWAYTEAM FTSC FTHG FTAG
## 1   1994 Bundesliga 1 1993-08-07 Bayern Munich       Freiburg  3-1    3    1
## 2   1994 Bundesliga 1 1993-08-07      Dortmund      Karlsruhe  2-1    2    1
## 3   1994 Bundesliga 1 1993-08-07      Duisburg     Leverkusen  2-2    2    2
## 4   1994 Bundesliga 1 1993-08-07       FC Koln Kaiserslautern  0-2    0    2
## 5   1994 Bundesliga 1 1993-08-07       Hamburg       Nurnberg  5-2    5    2
## 6   1994 Bundesliga 1 1993-08-07       Leipzig        Dresden  3-3    3    3
##   FTTG TEAM  M  W  D  L GF GA DIFF POINTS POSITION
## 1    4 <NA> NA NA NA NA NA NA   NA     NA       NA
## 2    3 <NA> NA NA NA NA NA NA   NA     NA       NA
## 3    4 <NA> NA NA NA NA NA NA   NA     NA       NA
## 4    2 <NA> NA NA NA NA NA NA   NA     NA       NA
## 5    7 <NA> NA NA NA NA NA NA   NA     NA       NA
## 6    6 <NA> NA NA NA NA NA NA   NA     NA       NA
```

```
home_goals <- df %>%
  select(SEASON, HOMETEAM, FTHG) %>%
  rename(Team = HOMETEAM, Goals = FTHG)

away_goals <- df %>%
  select(SEASON, AWAYTEAM, FTAG) %>%
  rename(Team = AWAYTEAM, Goals = FTAG)

team_goals <- bind_rows(home_goals, away_goals) %>%
  group_by(SEASON, Team) %>%
  summarize(TotalGoals = sum(Goals, na.rm = TRUE), .groups = "drop")

head(team_goals)
```

```
## # A tibble: 6 x 3
##   SEASON Team          TotalGoals
```

```
##    <int> <chr>              <int>
## 1   1994 Bayern Munich         68
## 2   1994 Dortmund              49
## 3   1994 Dresden               33
## 4   1994 Duisburg              41
## 5   1994 Ein Frankfurt         57
## 6   1994 FC Koln               49
```

```
teams_of_interest <- c("Bayern Munich", "Bayer Leverkusen", "Borussia Dortmund")

team_params <- team_goals %>%
  filter(Team %in% teams_of_interest) %>%
  group_by(Team) %>%
  summarize(
    lambda = mean(TotalGoals, na.rm = TRUE),
    sd_goals = sd(TotalGoals, na.rm = TRUE),
    seasons_counted = n()
  ) %>%
  ungroup()

team_params
```

```
## # A tibble: 1 x 4
##   Team          lambda sd_goals seasons_counted
##   <chr>          <dbl>    <dbl>           <int>
## 1 Bayern Munich   74.2     11.1              26
```

```
set.seed(123)
N <- 10000
future_seasons <- 1:10

sim_results <- data.frame()

for(team_i in teams_of_interest){

  lam <- team_params$lambda[team_params$Team == team_i]

  for(seas in future_seasons){

    draws <- rpois(N, lam)

    df_temp <- data.frame(
      Team = team_i,
      FutureSeason = seas,
      SimTrial = 1:N,
      GoalsSimulated = draws
    )

    sim_results <- bind_rows(sim_results, df_temp)
  }
}
```

```
## Warning in rpois(N, lam): NAs produced
```

```
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
## Warning in rpois(N, lam): NAs produced
```

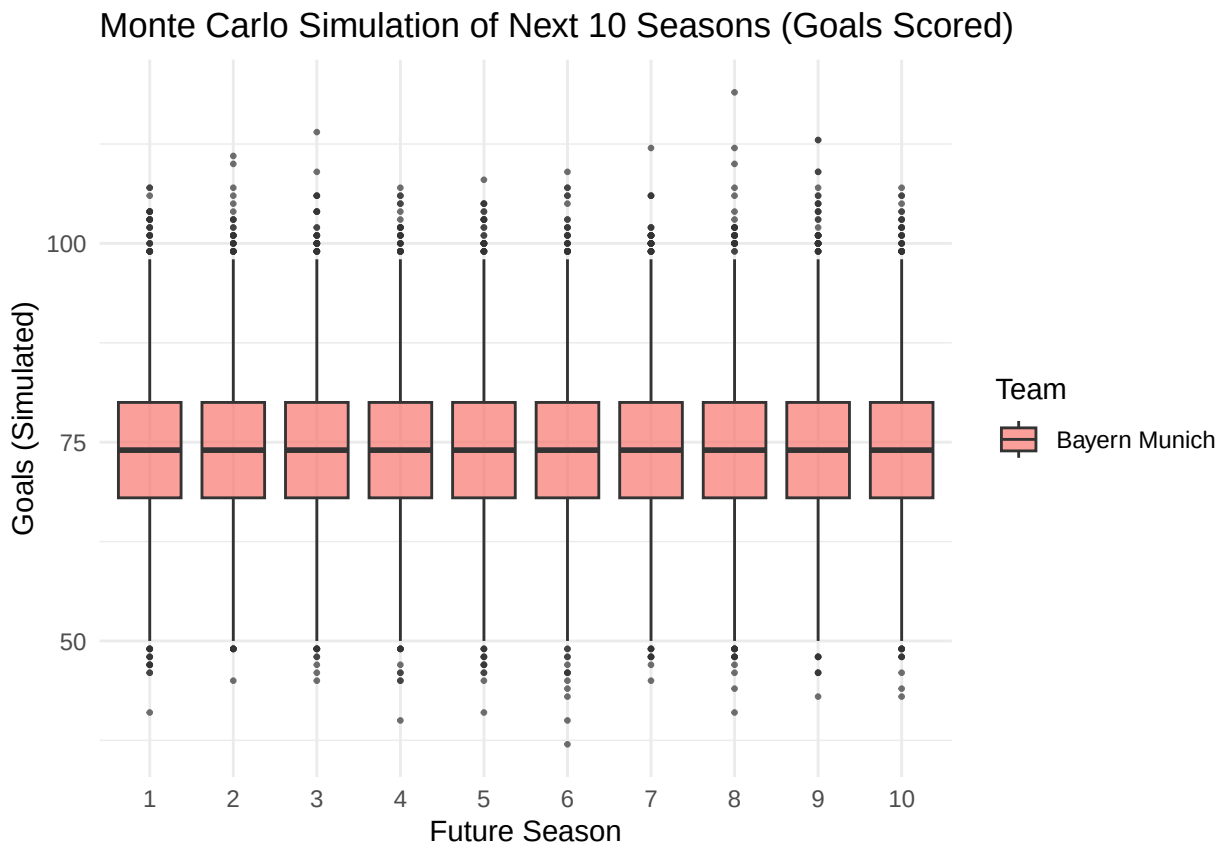
```
head(sim_results)
```

```
##            Team FutureSeason SimTrial GoalsSimulated
## 1 Bayern Munich            1        1             69
## 2 Bayern Munich            1        2             84
## 3 Bayern Munich            1        3             59
## 4 Bayern Munich            1        4             75
## 5 Bayern Munich            1        5             88
## 6 Bayern Munich            1        6             78
```

```
sim_summary <- sim_results %>%
  group_by(Team, FutureSeason) %>%
  summarize(
    MeanGoals = mean(GoalsSimulated, na.rm = TRUE),
    P10 = quantile(GoalsSimulated, 0.10, na.rm = TRUE),
    P90 = quantile(GoalsSimulated, 0.90, na.rm = TRUE),
    .groups = "drop"
  )
```

```
ggplot(sim_results, aes(x = factor(FutureSeason), y = GoalsSimulated, fill = Team)) +
  geom_boxplot(outlier.size = 0.5, alpha = 0.7) +
  labs(
    title = "Monte Carlo Simulation of Next 10 Seasons (Goals Scored)",
    x = "Future Season",
    y = "Goals (Simulated)"
  ) +
  theme_minimal()
```

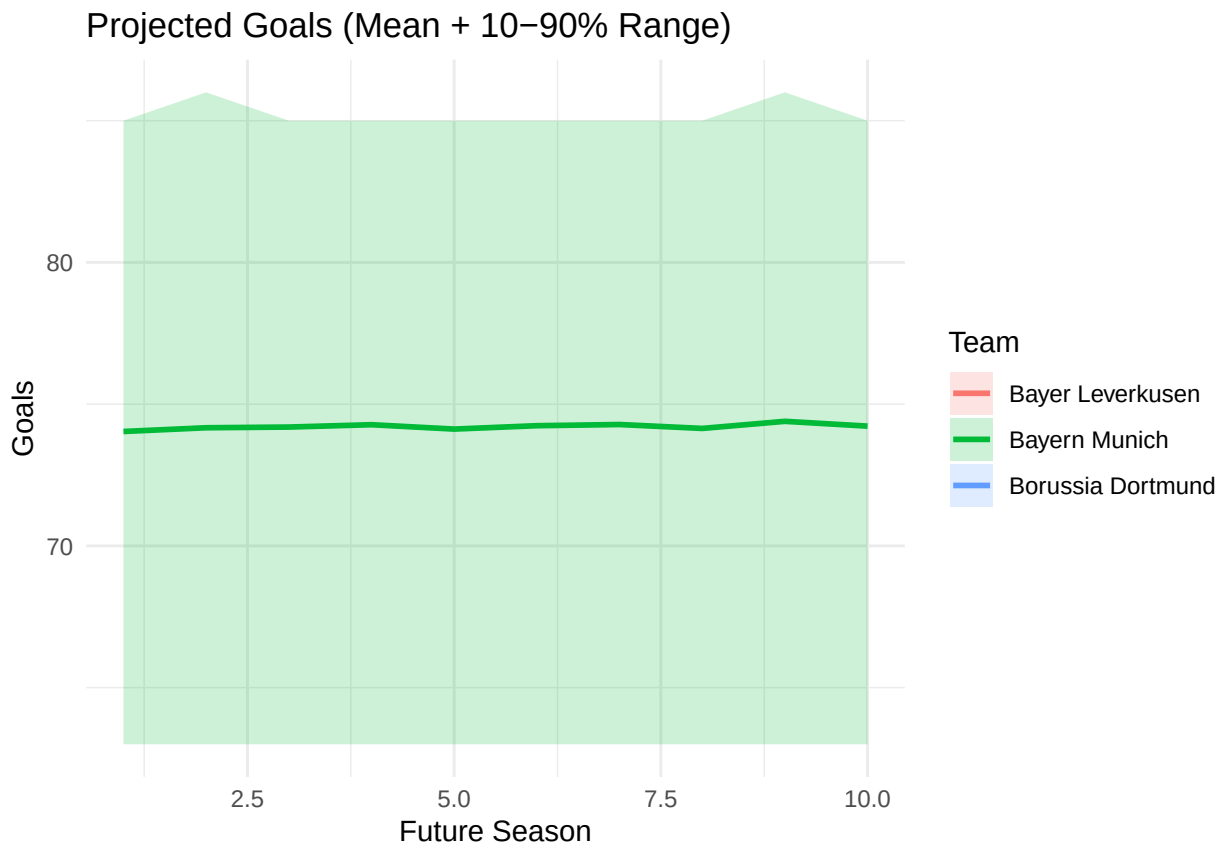
```
## Warning: Removed 200000 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
ggplot(sim_summary, aes(x = FutureSeason, y = MeanGoals, color = Team)) +
  geom_line(size = 1) +
  geom_ribbon(aes(ymin = P10, ymax = P90, fill = Team), alpha = 0.2, color = NA) +
  labs(
    title = "Projected Goals (Mean + 10-90% Range)",
    x = "Future Season",
    y = "Goals"
  ) +
  theme_minimal()
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```


Projected Goals (Mean + 10−90% Range)
80
70
Goals
2.5
5.0
7.5
10.0
Future Season
Team
Bayer Leverkusen
Bayern Munich
Borussia Dortmund