

EPL Match Analytics: A Data-Driven Exploration of Team Performance (2010-2020) Using R

Hillary Uzoh
2025-12-14

Project Overview

This project analyzes **ten seasons (2010/11 – 2019/20)** of English Premier League match statistics to uncover patterns in **team performance**, **referee tendencies**, and **match outcomes**.

By applying **data science techniques in R**, we transform raw match data into **actionable football insights** that can inform:

- Tactical decision-making
- Player recruitment strategies
- Referee match appointments

Objectives

- Understand dataset structure, handle missing values, and transformation of the data for insights analysis.
- Identify top-performing teams, home/away advantages, and consistency metrics.
- Examine referee tendencies in card distribution and match outcomes
- Create publication-quality visualizations for modern football analysis.

Loading Required Libraries

```
library(tidyverse)

## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr    1.1.4    ✓ readr    2.1.6
## ✓forcats  1.0.1    ✓ stringr  1.6.0
## ✓ ggplot2  4.0.1    ✓ tibble   3.3.0
## ✓ lubridate 1.9.4    ✓ tidyr   1.3.1
## ✓ purrr   1.2.0

## — Conflicts ————— tidyverse_conflicts() —
## ✘ dplyr::filter() masks stats::filter()
## ✘ dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(ggplot2)
library(ggrepel)
library(ggcorrplot)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(knitr)
library(DT)
```

```
library(ggthemes)
library(patchwork)
library(glue)
```

Load Dataset

```
# Read the dataset

epl_data <- read.csv("C:/Users/DELL/Desktop/DATASETS/football Data/epl-allseasons-matchstats.csv", stringsAsFactors = FALSE)
```

```
head(epl_data)
```

```
##   Season      Date    Referee HomeTeam AwayTeam FullTime Halftime
## 1 2010/11 2010-08-14     M Dean  Aston Villa  West Ham  HomeWin  HomeWin
## 2 2010/11 2010-08-14     P Dowd  Blackburn  Everton  HomeWin  HomeWin
## 3 2010/11 2010-08-14     S Attwell  Bolton  Fulham   Draw   Draw
## 4 2010/11 2010-08-14  M Clattenburg  Chelsea  West Brom  HomeWin  HomeWin
## 5 2010/11 2010-08-14     A Taylor Sunderland Birmingham   Draw  HomeWin
## 6 2010/11 2010-08-14  A Marriner  Tottenham  Man City   Draw   Draw
##   HomeGoals HomeGoalsHalftime HomeShots HomeShotsOnTarget HomeCorners HomeFouls
## 1        3                  2       23             11          16          15
## 2        1                  1       7              2           1          19
## 3        0                  0      13             9           4          12
## 4        6                  2      18             13          3          10
## 5        2                  1       6              2           3          13
## 6        0                  0      22             18          10          13
##   HomeYellowCards HomeRedCards AwayGoals AwayGoalsHalftime AwayShots
## 1            1            0            0            0          12
## 2            2            0            0            0          17
## 3            1            0            0            0          12
## 4            1            0            0            0          10
## 5            3            1            2            0          13
## 6            0            0            0            0          11
##   AwayShotsOnTarget AwayCorners AwayFouls AwayYellowCards AwayRedCards
## 1            2            7           15            2            0
## 2           12            3           14            1            0
## 3            7            8           13            3            0
## 4            4            1           10            0            0
## 5            7            6           10            3            0
## 6            7            3           16            2            0
```

```
# Initial exploration

cat("Dataset Dimensions:", dim(epl_data), "\n")
```

```
## Dataset Dimensions: 3800 23
```

```
str(epl_data)
```

```
## 'data.frame': 3800 obs. of 23 variables:  
## $ Season : chr "2010/11" "2010/11" "2010/11" "2010/11" ...  
## $ Date : chr "2010-08-14" "2010-08-14" "2010-08-14" "2010-08-14" ...  
## $ Referee : chr "M Dean" "P Dowd" "S Attwell" "M Clattenburg" ...  
## $ HomeTeam : chr "Aston Villa" "Blackburn" "Bolton" "Chelsea" ...  
## $ AwayTeam : chr "West Ham" "Everton" "Fulham" "West Brom" ...  
## $ FullTime : chr "HomeWin" "HomeWin" "Draw" "HomeWin" ...  
## $ Halftime : chr "HomeWin" "HomeWin" "Draw" "HomeWin" ...  
## $ HomeGoals : int 3 1 0 6 2 0 0 2 1 3 ...  
## $ HomeGoalsHalftime: int 2 1 0 2 1 0 0 2 0 2 ...  
## $ HomeShots : int 23 7 13 18 6 22 11 13 7 18 ...  
## $ HomeShotsOnTarget: int 11 2 9 13 2 18 6 7 4 10 ...  
## $ HomeCorners : int 16 1 4 3 3 10 6 5 9 5 ...  
## $ HomeFouls : int 15 19 12 10 13 13 8 17 13 9 ...  
## $ HomeYellowCards : int 1 2 1 1 3 0 1 0 1 2 ...  
## $ HomeRedCards : int 0 0 0 0 1 0 0 0 1 0 ...  
## $ AwayGoals : int 0 0 0 0 2 0 4 1 1 0 ...  
## $ AwayGoalsHalftime: int 0 0 0 0 0 0 3 0 0 0 ...  
## $ AwayShots : int 12 17 12 10 13 11 9 10 14 7 ...  
## $ AwayShotsOnTarget: int 2 12 7 4 7 7 7 6 7 3 ...  
## $ AwayCorners : int 7 3 8 1 6 3 4 5 11 3 ...  
## $ AwayFouls : int 15 14 13 10 10 16 11 13 15 5 ...  
## $ AwayYellowCards : int 2 1 3 0 3 2 1 2 3 2 ...  
## $ AwayRedCards : int 0 0 0 0 0 0 0 1 0 ...
```

```
cat("\nSummary statistics:\n")
```

```
##  
## Summary statistics:
```

```
summary(epl_data)
```

```
##   Season          Date        Referee      HomeTeam  
##  Length:3800    Length:3800    Length:3800    Length:3800  
##  Class :character Class :character Class :character Class :character  
##  Mode  :character Mode  :character Mode  :character Mode  :character  
##  
##  
##  
##   AwayTeam       FullTime      Halftime     HomeGoals  
##  Length:3800    Length:3800    Length:3800    Min.   :0.000  
##  Class :character Class :character Class :character  1st Qu.:1.000  
##  Mode  :character Mode  :character Mode  :character  Median :1.000  
##                                         Mean   :1.552  
##                                         3rd Qu.:2.000  
##                                         Max.   :8.000  
##  
##   HomeGoalsHalftime  HomeShots  HomeShotsOnTarget  HomeCorners  
##  Min.   :0.0000    Min.   : 0.00  Min.   : 0.000  Min.   : 0.000  
##  1st Qu.:0.0000    1st Qu.:10.00  1st Qu.: 3.000  1st Qu.: 4.000  
##  Median :0.0000    Median :14.00  Median : 5.000  Median : 6.000  
##  Mean   :0.6871    Mean   :14.14  Mean   : 5.698  Mean   : 5.984  
##  3rd Qu.:1.0000    3rd Qu.:17.00  3rd Qu.: 8.000  3rd Qu.: 8.000  
##  Max.   :5.0000    Max.   :43.00  Max.   :24.000  Max.   :19.000  
##  
##   HomeFouls      HomeYellowCards  HomeRedCards     AwayGoals
```

```

## Min. : 0.0  Min. :0.000  Min. :0.00000  Min. :0.000
## 1st Qu.: 8.0  1st Qu.:1.000  1st Qu.:0.00000  1st Qu.:0.000
## Median :10.0  Median :1.000  Median :0.00000  Median :1.000
## Mean    :10.5  Mean   :1.499  Mean   :0.05921  Mean   :1.193
## 3rd Qu.:13.0  3rd Qu.:2.000  3rd Qu.:0.00000  3rd Qu.:2.000
## Max.   :24.0  Max.  :7.000  Max.  :2.00000  Max.  :9.000
## AwayGoalsHalftime  AwayShots  AwayShotsOnTarget  AwayCorners
## Min. :0.0000  Min. : 0.0  Min. : 0.000  Min. : 0.00
## 1st Qu.:0.0000  1st Qu.: 8.0  1st Qu.: 3.000  1st Qu.: 3.00
## Median :0.0000  Median :11.0  Median : 4.000  Median : 4.00
## Mean   :0.5303  Mean   :11.3  Mean   : 4.571  Mean   : 4.77
## 3rd Qu.:1.0000  3rd Qu.:14.0  3rd Qu.: 6.000  3rd Qu.: 6.00
## Max.   :5.0000  Max.  :30.0  Max.  :20.000  Max.  :19.00
## AwayFouls  AwayYellowCards  AwayRedCards
## Min.   : 1.00  Min.  :0.000  Min. :0.00000
## 1st Qu.: 9.00  1st Qu.:1.000  1st Qu.:0.00000
## Median :11.00  Median :2.000  Median :0.00000
## Mean   :10.98  Mean   :1.761  Mean   :0.08132
## 3rd Qu.:13.00  3rd Qu.:3.000  3rd Qu.:0.00000
## Max.   :26.00  Max.  :9.000  Max.  :2.00000

```

```

# Total missing values in the dataset
total_missing <- sum(is.na(epl_data))

cat(glue(
  "Total missing values in the dataset: {total_missing}\n"
))

```

```
## Total missing values in the dataset: 0
```

```

# Check for duplicates:

duplicate_rows <- sum(duplicated(epl_data))

cat(glue(
  "Number of duplicate rows in the dataset: {duplicate_rows}\n"
))

```

```
## Number of duplicate rows in the dataset: 0
```

Data Cleaning and Transformation

```

# Convert Date to proper format
epl_data$Date <- as.Date(epl_data$Date)

# Extract additional features
epl_data <- epl_data %>%
  mutate(
    # Extract month and year
    Month = month(Date, label = TRUE),
    Year = year(Date),

    # Create match result categories
    Result = case_when(
      HomeTeam == "Brentford" & AwayTeam == "West Ham United" ~ "H vs A",
      ...
    )
  )

```

```

Result = factor(FullTime, levels = c("HomeWin", "Draw", "AwayWin")),

# Calculate additional metrics
HomeConversionRate = ifelse(HomeShots > 0, (HomeGoals / HomeShots) * 100, 0),
AwayConversionRate = ifelse(AwayShots > 0, (AwayGoals / AwayShots) * 100, 0),
HomeShotAccuracy = ifelse(HomeShots > 0, (HomeShotsOnTarget / HomeShots) * 100, 0),
AwayShotAccuracy = ifelse(AwayShots > 0, (AwayShotsOnTarget / AwayShots) * 100, 0),

# Calculate disciplinary metrics
HomeDisciplinaryPoints = HomeYellowCards + (HomeRedCards * 2),
AwayDisciplinaryPoints = AwayYellowCards + (AwayRedCards * 2),

# Game intensity metrics
TotalShots = HomeShots + AwayShots,
TotalGoals = HomeGoals + AwayGoals,
TotalCards = HomeYellowCards + AwayYellowCards + HomeRedCards + AwayRedCards,

# Goal difference
GoalDifference = HomeGoals - AwayGoals,

# Match competitiveness (closer games have lower values)
CompetitivenessIndex = abs(HomeGoals - AwayGoals)
)

```

Create Separate Dataframes for Team-centric Analysis (Home and Away)

```

home_team_data <- epl_data %>%
  select(Season, Date, Team = HomeTeam, Opponent = AwayTeam,
         Goals = HomeGoals, GoalsConceded = AwayGoals,
         Shots = HomeShots, ShotsOnTarget = HomeShotsOnTarget,
         Corners = HomeCorners, Fouls = HomeFouls,
         YellowCards = HomeYellowCards, RedCards = HomeRedCards,
         ConversionRate = HomeConversionRate,
         ShotAccuracy = HomeShotAccuracy,
         Result = FullTime) %>%
  mutate(Venue = "Home")

away_team_data <- epl_data %>%
  select(Season, Date, Team = AwayTeam, Opponent = HomeTeam,
         Goals = AwayGoals, GoalsConceded = HomeGoals,
         Shots = AwayShots, ShotsOnTarget = AwayShotsOnTarget,
         Corners = AwayCorners, Fouls = AwayFouls,
         YellowCards = AwayYellowCards, RedCards = AwayRedCards,
         ConversionRate = AwayConversionRate,
         ShotAccuracy = AwayShotAccuracy,
         Result = FullTime) %>%
  mutate(Venue = "Away")

# Combine for team performance analysis
team_performance <- bind_rows(home_team_data, away_team_data) %>%
  mutate(
    Points = case_when(
      (Venue == "Home" & Result == "HomeWin") ~ 3,
      (Venue == "Away" & Result == "AwayWin") ~ 3,
      Result == "Draw" ~ 1,
      TRUE ~ 0
    )
  )

```

```

    TRUE ~ 0
),
Win = ifelse(Points == 3, 1, 0),
Draw = ifelse(Points == 1, 1, 0),
Loss = ifelse(Points == 0, 1, 0)
)

head(team_performance)

```

##	Season	Date	Team	Opponent	Goals	GoalsConceded	Shots			
## 1	2010/11	2010-08-14	Aston Villa	West Ham	3	0	23			
## 2	2010/11	2010-08-14	Blackburn	Everton	1	0	7			
## 3	2010/11	2010-08-14	Bolton	Fulham	0	0	13			
## 4	2010/11	2010-08-14	Chelsea	West Brom	6	0	18			
## 5	2010/11	2010-08-14	Sunderland	Birmingham	2	2	6			
## 6	2010/11	2010-08-14	Tottenham	Man City	0	0	22			
					ShotsOnTarget	Corners	Fouls			
					Fouls	YellowCards	RedCards	ConversionRate	ShotAccuracy	
## 1		11	16	15	1	0	13.04348	47.82609		
## 2		2	1	19	2	0	14.28571	28.57143		
## 3		9	4	12	1	0	0.00000	69.23077		
## 4		13	3	10	1	0	33.33333	72.22222		
## 5		2	3	13	3	1	33.33333	33.33333		
## 6		18	10	13	0	0	0.00000	81.81818		
					Result	Venue	Points	Win	Draw	Loss
## 1	HomeWin	Home	3	1	0	0				
## 2	HomeWin	Home	3	1	0	0				
## 3	Draw	Home	1	0	1	0				
## 4	HomeWin	Home	3	1	0	0				
## 5	Draw	Home	1	0	1	0				
## 6	Draw	Home	1	0	1	0				

Match Stats Analysis and Visualization

```

# TOP 10 TEAMS BY TOTAL POINTS.

top_teams <- team_performance %>%
  group_by(Team) %>%
  summarise(
    TotalPoints = sum(Points),
    TotalMatches = n(),
    WinRate = mean(Win) * 100,
    PPG = TotalPoints / TotalMatches,
    .groups = 'drop'
  ) %>%
  arrange(desc(TotalPoints)) %>%
  head(10)

# visualization
p1 <- ggplot(top_teams, aes(x = TotalPoints, y = reorder(Team, TotalPoints))) +
  geom_col(aes(fill = TotalPoints == max(TotalPoints))) +
  geom_text(aes(label = TotalPoints),
            hjust = 1.2,
            size = 4,
            color = "white",

```

```

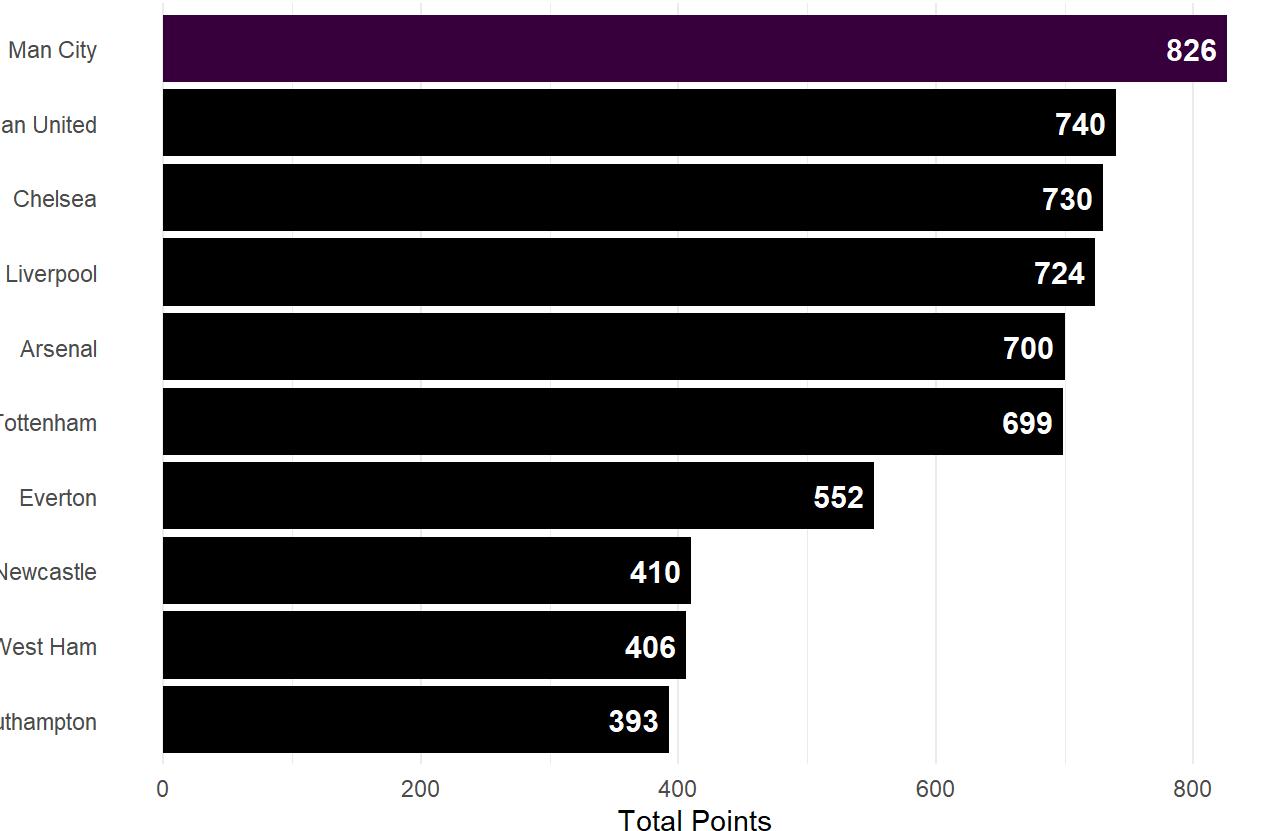
fontface = "bold") +
scale_fill_manual(values = c("TRUE" = "#38003c", "FALSE" = "black")) +
labs(title = "Top 10 Teams by Total Points (2010-2020)",
subtitle = "10 Seasons Combined Performance",
x = "Total Points",
y = NULL) +
theme_minimal() +
theme(
  legend.position = "none",
  plot.title = element_text(face = "bold", size = 16),
  panel.grid.major.y = element_blank(),
  panel.grid.minor.y = element_blank()
)

# Display
print(p1)

```

Top 10 Teams by Total Points (2010-2020)

10 Seasons Combined Performance



Distribution of Match Results

```

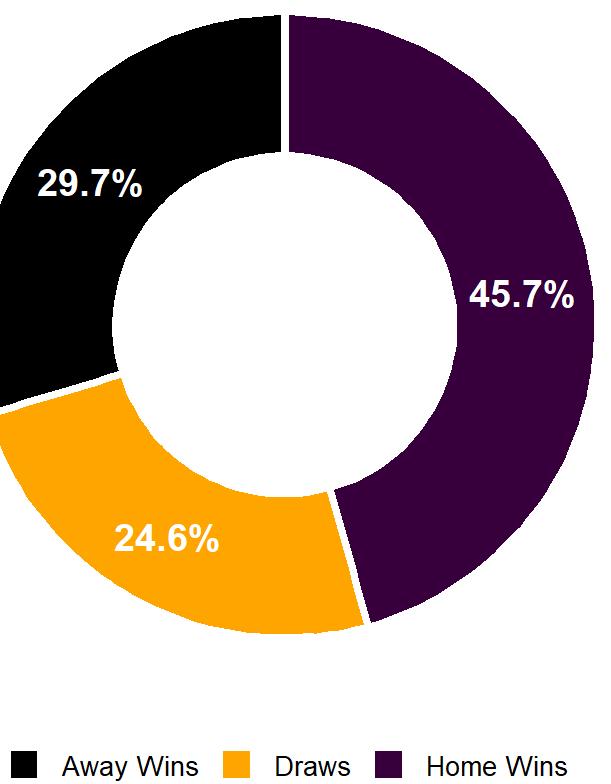
wins <- epl_data %>%
  summarise(
    `Home Wins` = sum(Result == "HomeWin"),
    Draws = sum(Result == "Draw"),
    `Away Wins` = sum(Result == "AwayWin")
  ) %>%
  pivot_longer(everything()) %>%
  mutate(
    Percent = round(value / sum(value) * 100, 1),
    Label = paste0(Percent, "%")
  )

```

```
# Visualization
ggplot(wins, aes(x = 2, y = value, fill = name)) +
  geom_col(color = "white", linewidth = 1.5) +
  geom_text(aes(label = Label),
            position = position_stack(vjust = 0.5),
            color = "white", size = 5, fontface = "bold") +
  coord_polar(theta = "y") +
  xlim(0.5, 2.5) +
  scale_fill_manual(
    name = NULL,
    values = c(
      `Home Wins` = "#38003C", # navy blue for Home Wins
      `Draws` = "#FFA500", # ORANGE for Draws
      `Away Wins` = "black" # black for Away Wins
    ),
    labels = c(
      `Home Wins` = "Home Wins",
      `Draws` = "Draws",
      `Away Wins` = "Away Wins"
    )
  ) +
  labs(
    title = "Distribution of Match Results",
    subtitle = "Analysis of the 10 Premier League Seasons"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5, size = 15),
    plot.subtitle = element_text(hjust = 0.5, size = 11, color = "gray40"),
    legend.position = "bottom",
    legend.text = element_text(size = 10)
  ) +
  guides(fill = guide_legend(nrow = 1))
```

Distribution of Match Results

Analysis of the 10 Premier League Seasons



■ Away Wins ■ Draws ■ Home Wins

Goals Analysis

```
# GOALS ANALYSIS: SCORED VS CONCEDED
goals_analysis <- team_performance %>%
  group_by(Team) %>%
  summarise(
    GoalsScored = sum(Goals),
    GoalsConceded = sum(GoalsConceded),
    GoalDifference = GoalsScored - GoalsConceded,
    .groups = 'drop'
  ) %>%
  filter(Team %in% top_teams$Team)

goals_analysis_sorted <- goals_analysis %>%
  arrange(desc(GoalsScored))

# VIZ
DT::datatable(
  goals_analysis_sorted,
  rownames = FALSE,
  options = list(
    pageLength = 10,
    autoWidth = TRUE,
    order = list(list(1, "desc")),
    dom = "tip"
  ),
  caption = htmltools::tags$caption()
```

```

style = "caption-side: top; text-align: left; font-weight: bold;",
"Top 10 Team Goals Performance (Ranked by Goals Scored)"
)
) %>%
formatStyle(
  "GoalsScored",
  background = styleColorBar(
    goals_analysis_sorted$GoalsScored,
    "#38003c"
  ),
  backgroundSize = "95% 80%",
  backgroundRepeat = "no-repeat",
  backgroundPosition = "center",
  color = "white",
  fontWeight = "bold"
) %>%
formatStyle(
  "GoalsConceded",
  background = styleColorBar(
    goals_analysis_sorted$GoalsConceded,
    "#E74C3C"
  ),
  backgroundSize = "95% 80%",
  backgroundRepeat = "no-repeat",
  backgroundPosition = "center"
) %>%
formatStyle(
  "GoalDifference",
  color = styleInterval(
    c(0),
    c("#E74C3C", "#38003c")
  ),
  fontWeight = "bold"
)
)

```

Top 10 Team Goals Performance (Ranked by Goals Scored)

Team	GoalsScored	GoalsConceded	GoalDifference
Man City	858	336	522
Liverpool	729	410	319
Arsenal	702	436	266
Chelsea	691	394	297
Man United	681	375	306
Tottenham	657	420	237
Everton	528	473	55
West Ham	33	521	-88
Newcastle	40	516	-113

Showing 1 to 10 of 10 entries

Previous 1 Next

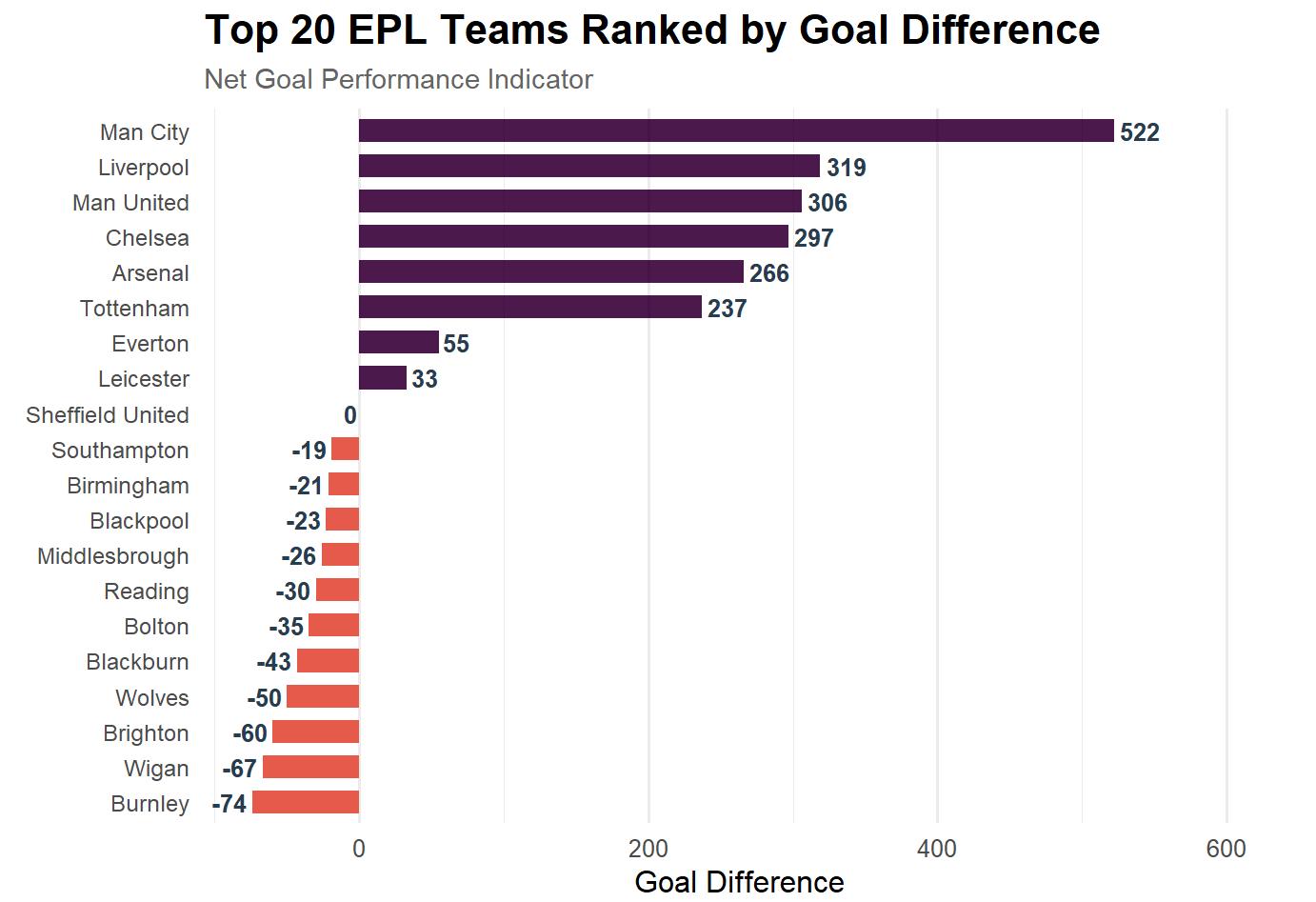
```
goals_analysis_top20 <- team_performance %>%
  group_by(Team) %>%
  summarise(
    GoalsScored = sum(Goals),
    GoalsConceded = sum(GoalsConceded),
    GoalDifference = GoalsScored - GoalsConceded,
    .groups = "drop"
  ) %>%
  arrange(desc(GoalDifference)) %>%
  slice_head(n = 20)

# viz
ggplot(
  goals_analysis_top20,
  aes(
    x = reorder(Team, GoalDifference),
    y = GoalDifference,
    fill = GoalDifference > 0
  )
) +
  geom_col(width = 0.65, alpha = 0.9) +
  geom_text(
    aes(
      label = GoalDifference,
      hjust = ifelse(GoalDifference > 0, -0.15, 1.15)
    ),
    size = 3.4,
    fontface = "bold",
    color = "#2C3E50"
  ) +
  coord_flip() +
  scale_fill_manual(
    values = c("TRUE" = "#38003c", "FALSE" = "#E74C3C"), #2ECC71
    guide = "none"
  ) +
  expand_limits(
    y = max(abs(goals_analysis_top20$GoalDifference)) * 1.15
  ) +
  labs(
    title = "Top 20 EPL Teams Ranked by Goal Difference",
    subtitle = "Net Goal Performance Indicator",
    x = NULL,
    y = "Goal Difference"
  ) +
  theme_minimal(base_size = 12) +
  theme(
```

```

    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 11, color = "grey40"),
    axis.text.y = element_text(size = 9),
    panel.grid.major.y = element_blank()
)

```



Season Trends

```

# SEASON TRENDS

points_trend <- team_performance %>%
  group_by(Season, Team) %>%
  summarise(Points = sum(Points), .groups = 'drop') %>%
  filter(Team %in% top_teams$Team)

# Get top 5 teams only
top_5_teams <- head(top_teams$Team, 5)

# Filter for top 5 teams
points_trend <- team_performance %>%
  group_by(Season, Team) %>%
  summarise(Points = sum(Points), .groups = 'drop') %>%
  filter(Team %in% top_5_teams)

# Create plot with rotated x-axis labels
p4 <- ggplot(points_trend, aes(x = Season, y = Points, group = Team)) +
  geom_line(aes(color = Team), linewidth = 1.2, alpha = 0.7) +
  geom_point(aes(color = Team), size = 2) +
  scale_color_viridis_d(option = "C", name = "Team") +
  theme_minimal() +
  theme(panel.grid.major.x = element_line())

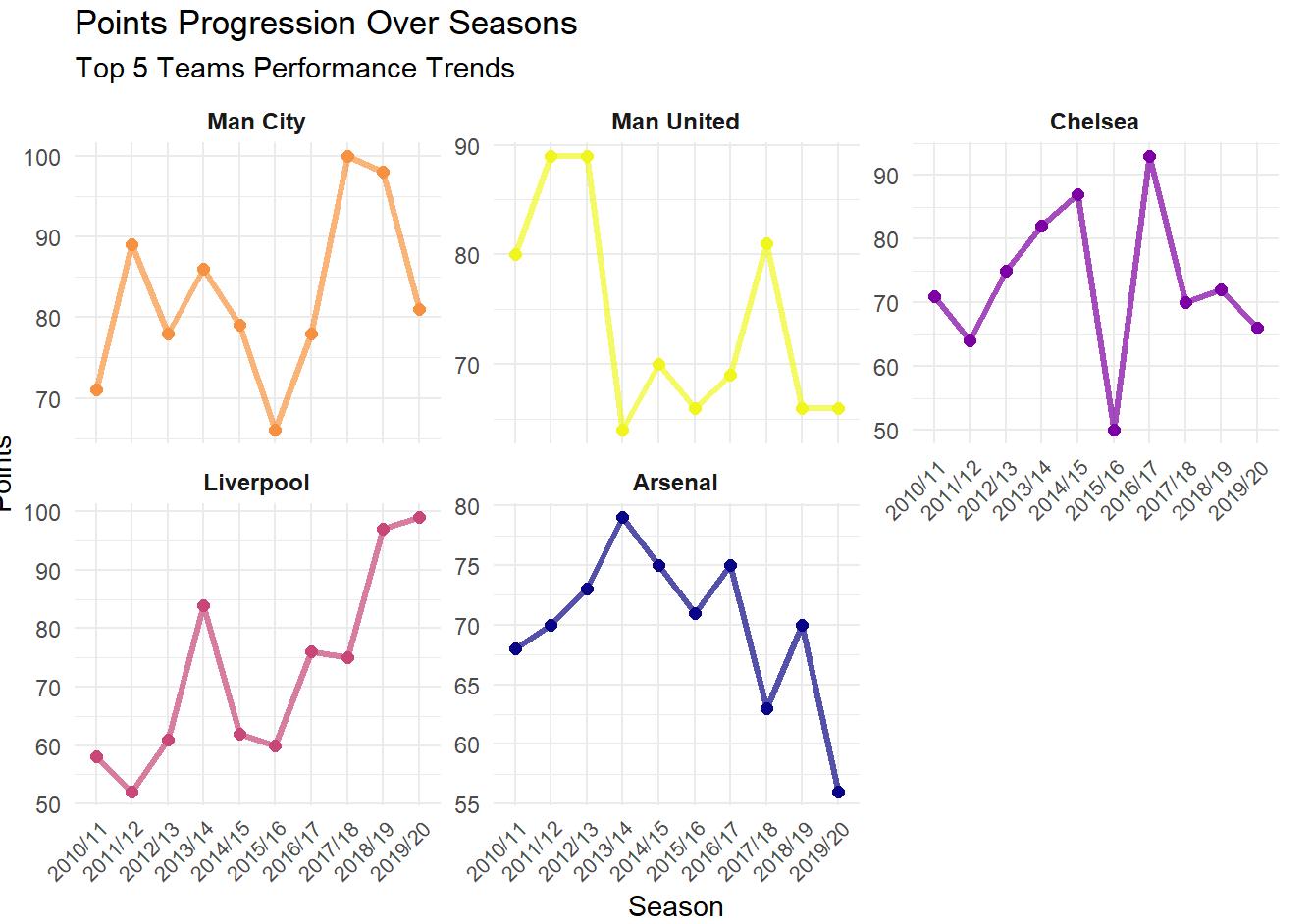
```

```

  labs(title = "Points Progression Over Seasons",
       subtitle = "Top 5 Teams Performance Trends",
       x = "Season", y = "Points") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 8), # Adjusted size
    legend.position = "none",
    strip.text = element_text(face = "bold", size = 9) # Facet title styling
  ) +
  facet_wrap(~ reorder(Team, -Points), scales = "free_y", ncol = 3) # Changed to 3 columns

print(p4)

```



Home and Away Team Performance

```

# Calculate away wins for each team
away_wins <- team_performance %>%
  filter(Venue == "Away") %>% # Focus only on away matches
  group_by(Team) %>%
  summarise(
    AwayMatches = n(),
    AwayWins = sum(Win), # Count wins
    AwayDraws = sum(Draw),
    AwayLosses = sum(Loss),
    AwayPoints = sum(Points),
    AwayWinRate = (AwayWins / AwayMatches) * 100,
    .groups = 'drop'
  ) %>%
  arrange(desc(AwayWins))

```

```

# Top 5 teams with most away wins
top5_away_wins <- away_wins %>%
  head(5) %>%
  mutate(Rank = "Top 5: Most Away Wins")

# Bottom 5 teams with least away wins
bottom5_away_wins <- away_wins %>%
  filter(AwayMatches >= 50) %>% # Filter for teams with sufficient matches
  tail(5) %>%
  mutate(Rank = "Bottom 5: Least Away Wins")

# Combine for visualization
away_wins_comparison <- bind_rows(top5_away_wins, bottom5_away_wins)

p_top <- ggplot(top5_away_wins,
                 aes(x = reorder(Team, -AwayWins), y = AwayWins)) +

  # Create custom fill colors: Manchester City gets sky blue, others get original green
  geom_bar(stat = "identity",
            aes(fill = ifelse(Team == "Man City", "Man City", "Other Teams")),
            alpha = 0.8,
            width = 0.6) + # Reduced bar width from default 0.9 to 0.6

  # Set custom colors
  scale_fill_manual(values = c("Man City" = "#38003c", # Sky blue
                               "Other Teams" = "black"), # Original green #2E8B57
                    guide = "none") + # Hide legend

  geom_text(aes(label = paste0(AwayWins, "\n(", round(AwayWinRate, 1), "%)")),
            vjust = -0.3, size = 3.5, lineheight = 0.8) +

  # Expand y-axis limits to accommodate labels
  expand_limits(y = max(top5_away_wins$AwayWins) * 1.15) +

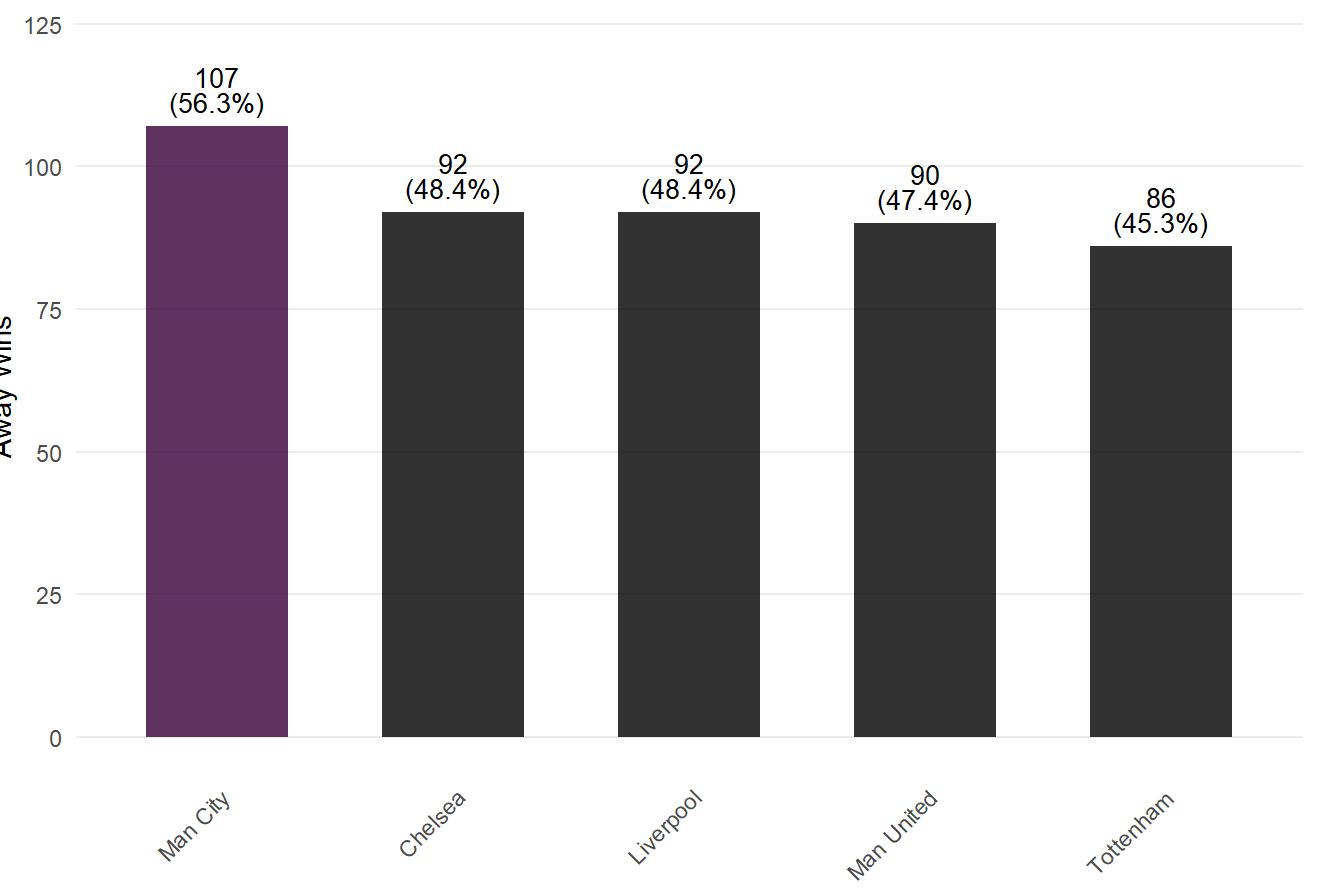
  labs(title = "TOP 5: Most Away Wins",
       x = NULL, y = "Away Wins") +

  theme_minimal() +
  theme(plot.title = element_text(face = "bold", color = "#38003c", hjust = 0.5),
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major.x = element_blank(), # Remove vertical grid lines for cleaner look
        panel.grid.minor.y = element_blank()) # Remove minor grid lines

print(p_top)

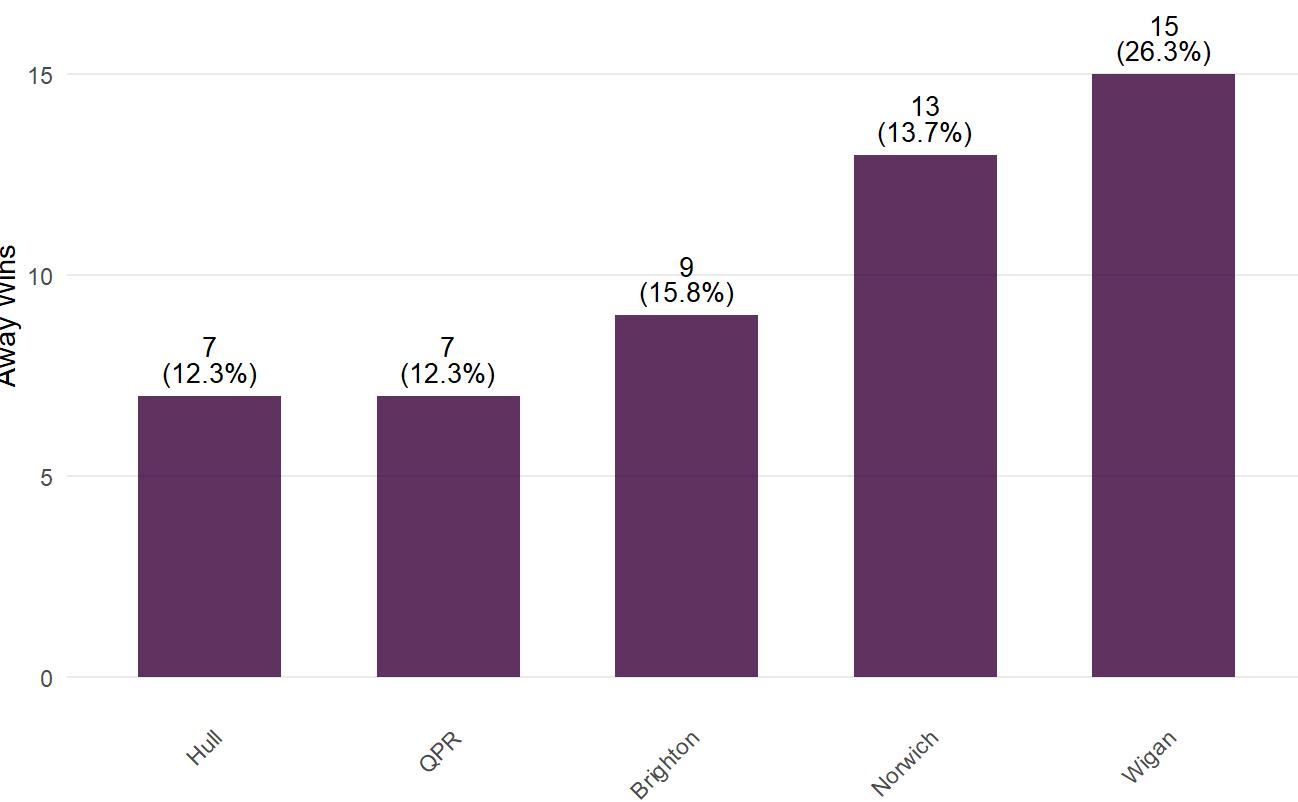
```

TOP 5: Most Away Wins



```
# Bottom 5 chart
p_bottom <- ggplot(bottom5_away_wins,
                     aes(x = reorder(Team, AwayWins), y = AwayWins)) +
  # Reduced bar width from default 0.9 to 0.6
  geom_bar(stat = "identity", fill = "#38003c", alpha = 0.8, width = 0.6) +
  # Position labels with adjusted vjust and expand limits for visibility
  geom_text(aes(label = paste0(AwayWins, "\n", round(AwayWinRate, 1), "%"))),
  vjust = -0.3, size = 3.5, lineheight = 0.8) +
  # Expand y-axis to ensure all labels are visible
  expand_limits(y = max(bottom5_away_wins$AwayWins) * 1.2) +
  labs(title = "BOTTOM 5: Least Away Wins",
       x = NULL, y = "Away Wins") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", color = "#38003c", hjust = 0.5),
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major.x = element_blank(), # Remove vertical grid lines
        panel.grid.minor.y = element_blank()) # Remove minor grid lines
print(p_bottom)
```

BOTTOM 5: Least Away Wins



Referee Insights

```
# Create referee analysis dataframe

referee_analysis <- epl_data %>%
  group_by(Referee) %>%
  summarise(
    Total_Matches = n(),
    Avg_Total_Cards = mean(TotalCards, na.rm = TRUE),
    Avg_Total_Fouls = mean(HomeFouls + AwayFouls, na.rm = TRUE),
    Avg_Total_Goals = mean(TotalGoals, na.rm = TRUE),
    Avg_Home_Yellows = mean(HomeYellowCards, na.rm = TRUE),
    Avg_Away_Yellows = mean(AwayYellowCards, na.rm = TRUE),
    Total_Red_Cards = sum(HomeRedCards + AwayRedCards, na.rm = TRUE),
    Home_Win_Rate = mean(Result == "H", na.rm = TRUE) * 100,
    Draw_Rate = mean(Result == "D", na.rm = TRUE) * 100,
    Away_Win_Rate = mean(Result == "A", na.rm = TRUE) * 100,
    Avg_Competitiveness = mean(CompetitivenessIndex, na.rm = TRUE),
    Avg_Shots = mean(TotalShots, na.rm = TRUE)
  ) %>%
  filter(Total_Matches >= 10) %>%
  mutate(
    Cards_per_Foul = Avg_Total_Cards / Avg_Total_Fouls,
    Strictness_Score = Avg_Total_Cards * 0.6 + Avg_Total_Fouls * 0.4,
    Yellow_Card_Bias = Avg_Home_Yellows - Avg_Away_Yellows
  ) %>%
  arrange(desc(Total_Matches))

# Create Referee strictness analysis dataframe
referee_strictness <- referee_analysis %>%
```

```

arrange(Avg_Total_Cards) %>%
mutate(
  Strictness_Rank = row_number(),
  Strictness_Category = case_when(
    Avg_Total_Cards <= quantile(Avg_Total_Cards, 0.25, na.rm = TRUE) ~ "Lenient",
    Avg_Total_Cards >= quantile(Avg_Total_Cards, 0.75, na.rm = TRUE) ~ "Strict",
    TRUE ~ "Average"
  )
)

# Create correlation dataframe for visualization
ref_cor_data <- referee_analysis %>%
  select(Avg_Total_Cards, Avg_Total_Fouls, Avg_Total_Goals,
         Home_Win_Rate, Avg_Competitiveness, Avg_Shots, Total_Matches)

```

```

# INSIGHT Top 10 Referees by Matches Officiated
top10_referees <- referee_analysis %>%
  head(10) %>%
  arrange(Total_Matches)

```

```

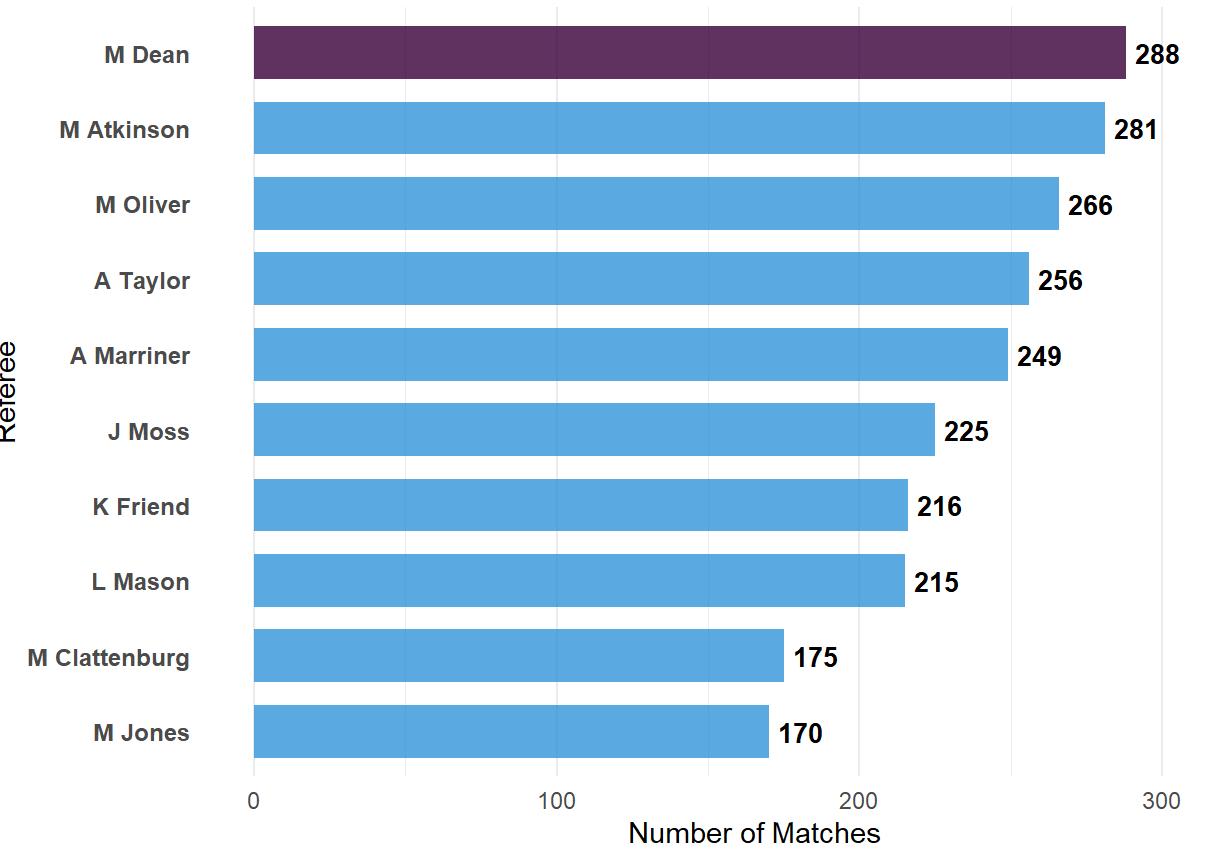
# Top 10 Referees by Matches Officiated
p1 <- ggplot(top10_referees,
             aes(x = reorder(Referee, Total_Matches), y = Total_Matches)) +
  # Color bars based on whether they're the maximum
  geom_bar(stat = "identity",
            aes(fill = Total_Matches == max(Total_Matches),
                alpha = 0.8, width = 0.7) +
  scale_fill_manual(values = c("TRUE" = "#38003c",
                             "FALSE" = "#3498DB"),
                    guide = "none") +
  geom_text(aes(label = Total_Matches),
            hjust = -0.2, size = 3.5, fontface = "bold") +
  coord_flip() +
  expand_limits(y = max(top10_referees$Total_Matches) * 1.15) +
  labs(title = "Top 10 Most Experienced EPL Referees",
       subtitle = "By Total Matches Officiated",
       x = "Referee",
       y = "Number of Matches") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", color = "#2C3E50", size = 14),
        plot.subtitle = element_text(color = "#7F8C8D", size = 10),
        axis.text.y = element_text(size = 9, face = "bold"),
        panel.grid.major.y = element_blank())

print(p1)

```

Top 10 Most Experienced EPL Referees

By Total Matches Officiated



```
# Create top10_strict dataframe
top10_strict <- referee_analysis %>%
  arrange(desc(Avg_Total_Cards)) %>%
  head(10) %>%
  mutate(Referee = factor(Referee, levels = Referee))

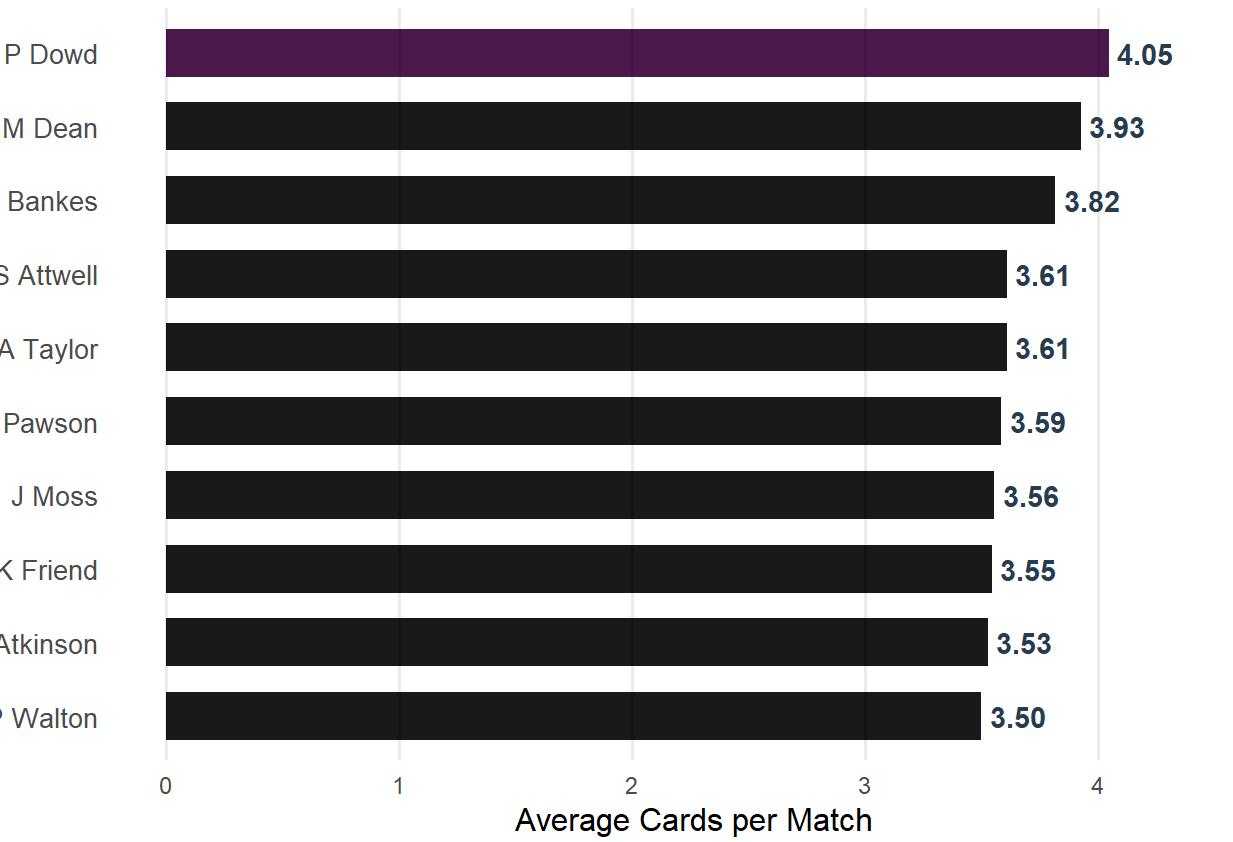
# Identify the strictest referee (highest avg cards)
top_ref <- top10_strict %>%
  slice_max(Avg_Total_Cards, n = 1) %>%
  pull(Referee)

# visualization
p6 <- ggplot(
  top10_strict,
  aes(
    x = reorder(Referee, Avg_Total_Cards),
    y = Avg_Total_Cards,
    fill = Referee == top_ref
  )
) +
  geom_col(width = 0.65, alpha = 0.9) +
  # Value labels
  geom_text(
    aes(label = sprintf("%.2f", Avg_Total_Cards)),
    hjust = -0.15,
    size = 3.8,
    fontface = "bold",
    color = "#2C3E50"
  ) +
```

```
# Flip for readability
coord_flip() +  
  
# Improve spacing
expand_limits(y = max(top10_strict$Avg_Total_Cards) * 1.12) +  
  
# Colors (highlight top referee)
scale_fill_manual(
  values = c("TRUE" = "#38003c", "FALSE" = "black"),
  guide = "none"
) +  
  
# Labels
labs(
  title = "Top 10 Strictest EPL Referees",
  subtitle = "Average Total Cards per Match",
  x = NULL,
  y = "Average Cards per Match"
) +  
  
# Theme improvements
theme_minimal(base_size = 12) +
theme(
  plot.title = element_text(
    face = "bold",
    size = 16,
    color = "#2C3E50"
  ),
  plot.subtitle = element_text(
    size = 11,
    color = "#7F8C8D"
  ),
  axis.text.y = element_text(size = 10),
  axis.text.x = element_text(size = 9),
  panel.grid.major.y = element_blank(),
  panel.grid.minor = element_blank()
)  
  
print(p6)
```

Top 10 Strictest EPL Referees

Average Total Cards per Match



```
# Calculate average cards per season for each team
team_season_cards <- team_performance %>%
  group_by(Team, Season) %>%
  summarise(
    SeasonMatches = n(),
    SeasonYellowCards = sum(YellowCards, na.rm = TRUE),
    SeasonRedCards = sum(RedCards, na.rm = TRUE),
    SeasonTotalCards = SeasonYellowCards + SeasonRedCards,
    .groups = 'drop'
  ) %>%
  group_by(Team) %>%
  summarise(
    Seasons = n_distinct(Season),
    TotalMatches = sum(SeasonMatches),
    AvgYellowPerSeason = mean(SeasonYellowCards, na.rm = TRUE),
    AvgRedPerSeason = mean(SeasonRedCards, na.rm = TRUE),
    AvgTotalPerSeason = mean(SeasonTotalCards, na.rm = TRUE),
    YellowPerMatch = sum(SeasonYellowCards) / TotalMatches,
    RedPerMatch = sum(SeasonRedCards) / TotalMatches,
    CardsPerMatch = sum(SeasonTotalCards) / TotalMatches,
    .groups = 'drop'
  ) %>%
  arrange(desc(AvgTotalPerSeason))

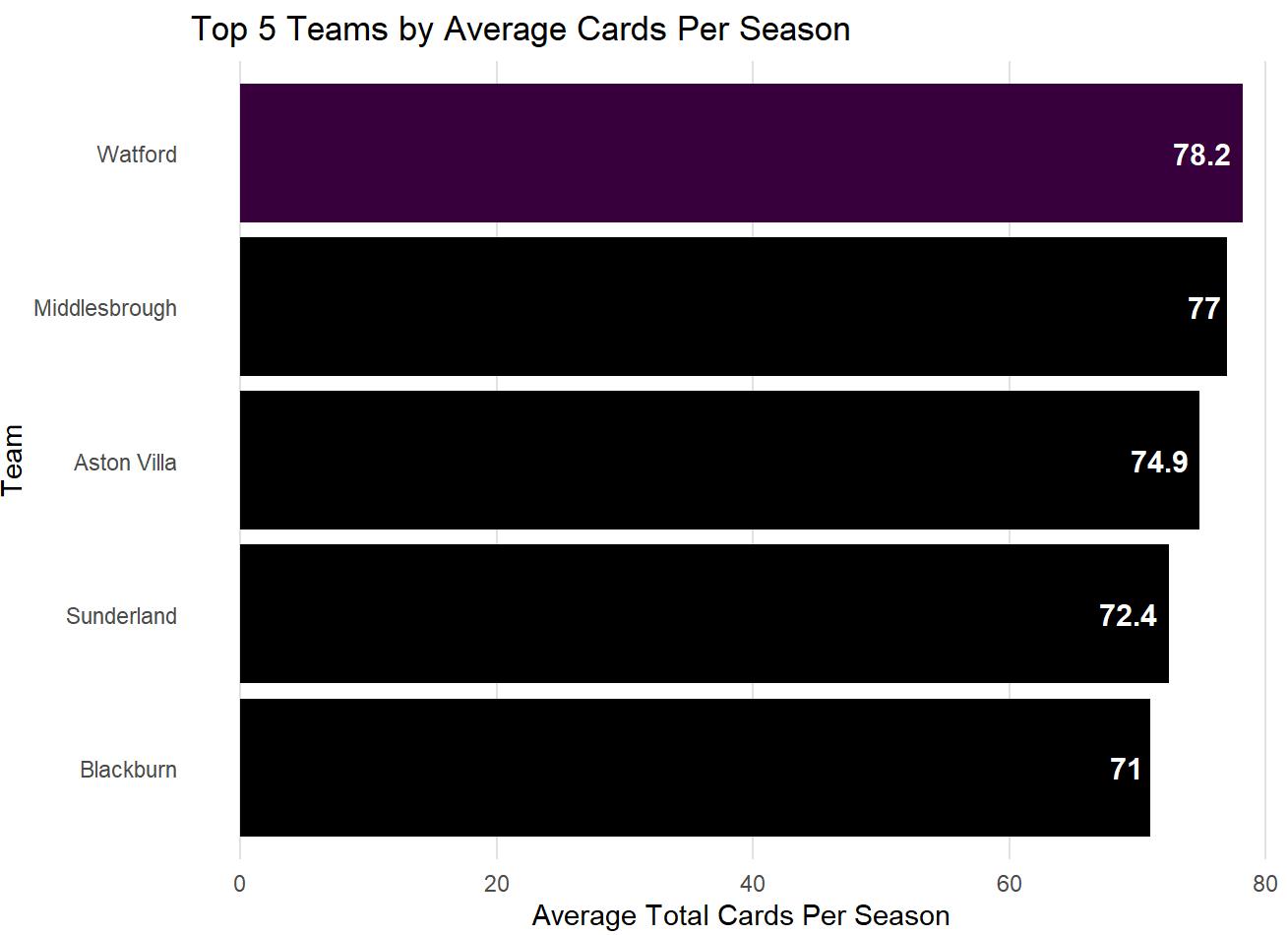
# Top 10 teams with highest average cards per season
top_cards_teams <- team_season_cards %>%
  filter(Seasons >= 3) %>% # Teams with at least 3 seasons
  head(10)
```

```
top10_cards_table <- team_season_cards %>%
```

```
arrange(desc(AvgRedPerSeason)) %>%
slice_head(n = 10) %>%
select(
  Team,
  Seasons,
  AvgYellowPerSeason,
  AvgRedPerSeason,
  AvgTotalPerSeason
)

# Get top 5 teams with highest average cards per season
top5_cards <- team_season_cards %>%
  arrange(desc(AvgTotalPerSeason)) %>%
  slice(1:5)

# horizontal bar chart
ggplot(top5_cards, aes(x = AvgTotalPerSeason, y = reorder(Team, AvgTotalPerSeason))) +
  geom_col(aes(fill = AvgTotalPerSeason == max(AvgTotalPerSeason))) +
  geom_text(aes(label = round(AvgTotalPerSeason, 1)),
            hjust = 1.2,
            size = 4,
            color = "white",
            fontface = "bold") +
  scale_fill_manual(values = c("TRUE" = "#38003c", "FALSE" = "black")) +
  labs(
    title = "Top 5 Teams by Average Cards Per Season",
    x = "Average Total Cards Per Season",
    y = "Team"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    panel.grid.major.x = element_line(color = "gray90"),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank()
)
```



Insights

Key findings from the analysis:

- Home advantage is strong with 45.7% of matches were home wins.
- Manchester City dominated the decade with the highest total points (826 points) and most away wins (56.3%) and other match deciding metrics.
- Referees vary in strictness with some averaging over 3.5 cards per match.
- Goal difference separates top teams, regular top 5 cub sides exceeded +300 difference.
- Away form is critical – Top clubs won over 40% of their away matches.

Project Summary

This project analyzed ten seasons (2010/11–2019/20) of English Premier League match statistics sourced from Kaggle to uncover patterns in team performance, disciplinary behavior, and referee officiating trends.

Using R and modern football analytics techniques, raw match-level data was transformed into structured metrics and visual insights, including goal difference rankings, team discipline indicators, and referee strictness measures. The analysis highlights how data-driven evaluation can support tactical planning, performance benchmarking, and evidence-based decision-making in football.

Conclusion

This project demonstrates how modern data science techniques in R can be applied to football analytics, bridging raw match data and strategic football intelligence through clear visualization and structured analysis.