ETC5513-Assignment3

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(ggplot2)
library(Polychrome)
library(scales)
```

```

Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

```
epl_raw <- read.csv("data/epl_final.csv")
epl_data <- epl_raw %>%
  select(
    Season,
    HomeTeam, AwayTeam,
    FullTimeHomeGoals, FullTimeAwayGoals,
    FullTimeResult,
    HomeShots, AwayShots
  )
```

Executive Summary

Introduction

Methodology

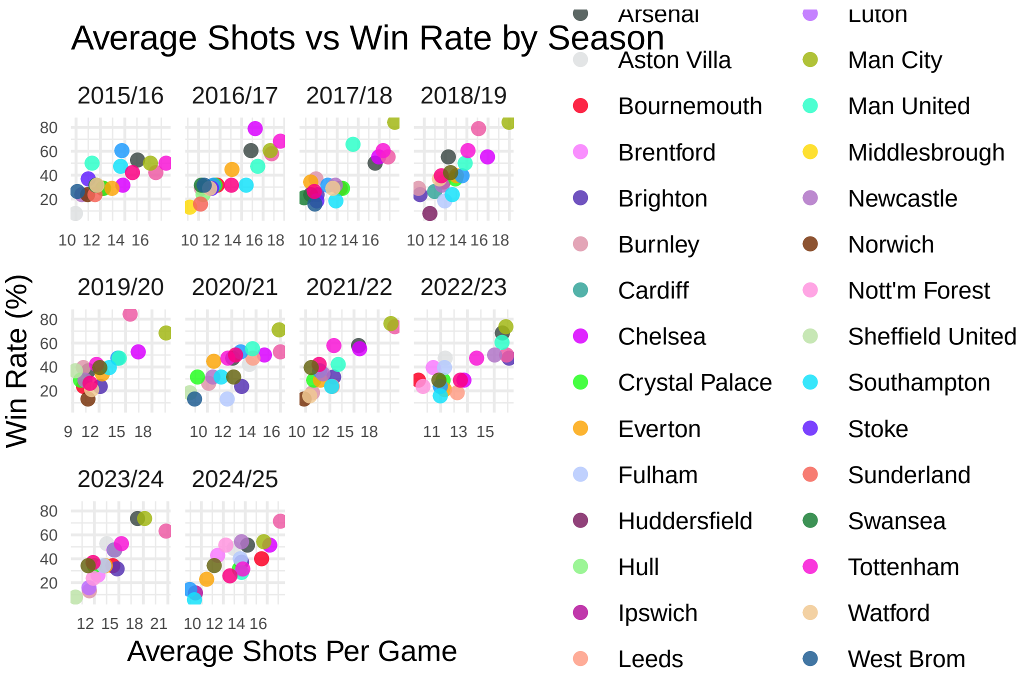
Results

```
# Import data and process
epl_raw <- read.csv("data/epl_final.csv")

# Selecting the Columns relevant to analysis
epl_data <- epl_raw %>%
  select(
    Season,
    HomeTeam,
    AwayTeam,
    FullTimeResult,
    HomeShots,
    AwayShots
    )

# Filter to only include most recent seasons
recent_seasons <- c("2015/16", "2016/17", "2017/18", "2018/19", "2019/20",
                    "2020/21", "2021/22", "2022/23", "2023/24", "2024/25")

epl_recent <- epl_data %>%
  filter(Season %in% recent_seasons)

# Get stats for home games
```

```r
home_stats <- epl_recent %>%
  mutate(
    Team = HomeTeam,
    Shots = HomeShots,
    Win = ifelse(FullTimeResult == "H", 1, 0)
  ) %>%
  select(Season, Team, Shots, Win)

# Get stats for away games
away_stats <- epl_recent %>%
  mutate(
    Team = AwayTeam,
    Shots = AwayShots,
    Win = ifelse(FullTimeResult == "A", 1, 0)
  ) %>%
  select(Season, Team, Shots, Win)

# Combine datasets for home and away stats
team_match_data <- bind_rows(home_stats, away_stats)

team_season_summary <- team_match_data %>%
  group_by(Season, Team) %>%
  summarise(
    TotalMatches = n(),
    TotalShots = sum(Shots, na.rm = TRUE),
    TotalWins = sum(Win, na.rm = TRUE),
    AvgShotsPerGame = TotalShots / TotalMatches,
    WinRatePercent = 100 * TotalWins / TotalMatches
  ) %>%
  ungroup()
```

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

```r
# Facet Plot to compare the Average Shos taken vs Win rate per season by each team

# Number of Teams:
number_of_teams <- length(unique(team_season_summary$Team))
# Creating color palette:
distinct_team_colors <- createPalette(number_of_teams,  seedcolors = c("#000000", "#FFFFFF")
# Assign team names as names of the color vector
names(distinct_team_colors) <- unique(team_season_summary$Team)

ggplot(
  team_season_summary,
  aes(x = AvgShotsPerGame,
      y = WinRatePercent,
      color = Team)) +
  geom_point(size = 2, alpha = 0.85) +
  facet_wrap(~ Season, scales = "free_x") +
  scale_color_manual(values = distinct_team_colors) +
   scale_x_continuous(labels = number_format(accuracy = 1)) +
  scale_y_continuous(labels = number_format(accuracy = 1)) +
  labs(
    title = "Average Shots vs Win Rate by Season",
    x = "Average Shots Per Game",
    y = "Win Rate (%)",
    color = "Team"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(
    hjust = 1, size = 6
      ),
    axis.text.y =  element_text(
      hjust = 1, size = 6
      )
    )
```



Discussion,Conclusion & Recommendatinons

References