Project Part II

Dylan Li, Liam Quach, Brendan Callender

1. Data Cleaning Summary

To create our multi-level dataset, we had to scrape data from fbref.com (Football Reference) and transfermarkt.com. We scraped data for premier league teams starting from the 2017 season up to the most recently completed 2023 season. The data from fbref included the total points for each team, goals scored, goals conceded, expected goals, expected goals conceded, average rate of possession over the season, average age of squad, total shots, total shots on target, average distance of shots from goal and more. The data scraped for each team from transfermarkt related to each teams activity in the transfer market that season. These variables included money spent on incoming players, money gained on outgoing players, net spend, number of players in, and number of players out. To combine the two datasets, we had to clean the squad names in the datasets to reconcile the different naming conventions for teams on different sites (See cases below). Once the data was combined we performed a final step of data cleaning to create our final analysis dataset. This cleaning included transforming applicable variables to per90 (per game) values instead of season total. This allows for more easily interpretable results. Additionally, transfermarkt, had "-" to represent 0 which later turned into NA values so these values were replaced with zeros.

Code to Scrape from Fbref

```r
years <- 2017:2023 # grab data between 2017-2018 season up to 2023-2024 season
res_df1 <- data.frame() # initialize dataframe for results

for(year in years) {

  # Create url for current year
  url <- glue::glue("https://fbref.com/en/comps/9/{year}-{year+1}/{year}-{year+1}-Premier-League-Stats")
  webpage <- read_html(url) # pull html for current year
  tables <- webpage %>%
  html_elements(".stats_table") %>% # pull all .stats_table elements from webpage
  html_table()

  # Extract variables from different tables on the webpage

  table1 <- tables %>% .[[1]] %>% select(Squad, GF, GA, Pts, xG, xGA)

  table2 <- tables %>% .[[3]] %>% .[,1:4]
  names(table2) <- table2[1,] # fix column names
  table2 <- table2[-1,] %>% select(-`# Pl`)

  table3 <- tables %>% .[[9]]
  names(table3) <- table3[1,] # fix column names
  table3 <- table3[-1,]
  table3 <- table3 %>% select(Squad, Sh, SoT, Dist)

  table4 <- tables %>% .[[5]] %>% .[,1:17]
  names(table4) <- table4[1,] # fix names
  table4 <- table4[-1,]
  table4 <- table4 %>% select(Squad, SoTA, CS)

  # join to combine all data for the current year
  final_table <- table1 %>%
    left_join(table2, by = "Squad") %>%
    left_join(table3, by = "Squad") %>%
    left_join(table4, by = "Squad") %>%
    mutate(Season = glue::glue("{year}-{year+1}"), .before = 2)

  # add current year table to results
  res_df1 <- rbind(
    res_df1,
    final_table
  )

  # delay call to webpage
  Sys.sleep(2)

}

rm(tables, table1, table2, table3, table4) # clear temporary variables
```

Code to Scrape from Transfermarkt

```r
years <- 2017:2023 # grab data between 2017-2018 season up to 2023-2024 season
res_df2 <- data.frame() # initialize dataframe for results

for(year in years) {

  # Create url for current year
  url <- glue::glue("https://www.transfermarkt.us/premier-league/einnahmenausgaben/wettbewerb
/GB1/plus/0?ids=a&sa=&saison_id={year}&saison_id_bis={year}&nat=&pos=&altersklasse=&w_s=&leihe
=&intern=0")

  webpage <- read_html(url) # load html for current year

  table <- webpage %>%
    html_elements("table.items") %>% # grab all "table.items" elements from html
    html_table() %>%
    .[[1]] %>%
    .[,3:8]

  # fix column names in table
  names(table) <- c("Squad", "Expenditure", "Arrivals", "Income", "Depatures", "Balance")

  # create season variable
  table <- table %>% mutate(Season = glue::glue("{year}-{year+1}"), .before = 2)

  # add current year table to results
  res_df2 <- rbind(
    res_df2,
    table
  )

  Sys.sleep(2) # delay vall to webpage

}

rm(table) # clear temporaty variable(s)
```

Combine Fbref and Transfermarkt data

```r
prem_points <- res_df1

team_expenditures <- res_df2 %>%
  mutate(Squad = str_replace(Squad, " FC", ""),
         Squad = str_replace(Squad, "AFC ", ""),
         Squad = str_replace(Squad, " Town", ""),
         Squad = str_replace(Squad, "United", "Utd"),
         Squad = case_when(
           Squad == "Brighton & Hove Albion" ~ "Brighton",
           Squad == "Wolverhampton Wanderers" ~ "Wolves",
           Squad == "Tottenham Hotspur" ~ "Tottenham",
           Squad == "West Ham Utd" ~ "West Ham",
           Squad == "West Bromwich Albion" ~ "West Brom",
           Squad == "Leeds Utd" ~ "Leeds United",
           Squad == "Nottingham Forest" ~ "Nott'ham Forest",
           Squad == "Luton" ~ "Luton Town",
           TRUE ~ Squad
         ))


prem_data <- prem_points %>%
  left_join(team_expenditures, by = c("Squad", "Season"), keep = FALSE)
```

Clean Data

```r
make_per_90 <- function(stat) {stat / 38}

prem_data_out <- prem_data %>%
   # convert string transfer varoables to numeric
  mutate(Expenditure = as.numeric(str_replace(str_replace(Expenditure, "€", ""), "m", "")),
         Income = as.numeric(str_replace(str_replace(Income, "€", ""), "m", "")),
         Balance = as.numeric(str_replace(str_replace(Balance, "€", ""), "m", ""))) %>%
  mutate(across(.cols = c(Age, Poss, Sh, SoT, Dist, SoTA, CS), .fns = as.numeric)) %>%
  # convert team variables to per/90 values
  mutate(across(.cols = c(GF, GA, xG, xGA, Sh, SoT, SoTA), .fns = make_per_90)) %>%
    mutate_all(~replace(., is.na(.), 0)) # replace NA values with 0
```

2. Exploratory Data Analysis

Library Package

```
library(tidyverse)
library(labelled)
library(corrplot)
library(lme4)
```

Load Data

```
prem <- read_csv(here::here("data", "prem_multi_level.csv"))
```

Add Labels

```
prem <- prem %>%
  set_variable_labels(
    Squad       = "Team Name",
    Season      = "Premier League Season",
    GF          = "Goals Scored per90",
    GA          = "Goals Against per90",
    Pts         = "Point Total",
    xG          = "Expected Goals per90",
    xGA         = "Expected Goals Against per90",
    Age         = "Average Age of Squad",
    Poss        = "Average Posession",
    Sh          = "Shots per90",
    SoT         = "Shots on Target per90",
    Dist        = "Average Dist of Shots from Goal",
    SoTA        = "Shots on Target Against per90",
    CS          = "Clean Sheets",
    Expenditure = "Money spent on incomings",
    Arrivals    = "# Players In",
    Income      = "Money gained on outgoings",
    Departures   = "# Players Out",
    Balance     = "Net Spend (Income - Expenditures)"
  )
```

Summary Statistics

```
# Summarize numeric variables in the dataset to get a sense of the data
summary_stats <- prem %>%
  select(GF, GA, xG, xGA, Poss, Age, Sh, SoT, Dist, SoTA, CS, Expenditure, Arrivals, Income,
  summary()
summary_stats
```

```
       GF               GA               xG              xGA
 Min.   :0.5263   Min.   :0.5789   Min.   :0.7579   Min.   :0.6263
 1st Qu.:1.0526   1st Qu.:1.2039   1st Qu.:1.0842   1st Qu.:1.2026
 Median :1.3421   Median :1.4211   Median :1.2987   Median :1.3829
 Mean   :1.4190   Mean   :1.4190   Mean   :1.3655   Mean   :1.3656
 3rd Qu.:1.7434   3rd Qu.:1.6579   3rd Qu.:1.5520   3rd Qu.:1.5645
 Max.   :2.7895   Max.   :2.7368   Max.   :2.4211   Max.   :2.0526
      Poss             Age              Sh              SoT
 Min.   :35.40   Min.   :24.20   Min.   : 8.395   Min.   :2.421
 1st Qu.:43.77   1st Qu.:26.07   1st Qu.:10.862   1st Qu.:3.493
 Median :48.90   Median :26.60   Median :12.053   Median :3.908
 Mean   :50.00   Mean   :26.64   Mean   :12.530   Mean   :4.184
 3rd Qu.:54.70   3rd Qu.:27.20   3rd Qu.:13.967   3rd Qu.:4.875
 Max.   :71.00   Max.   :29.40   Max.   :20.553   Max.   :6.947
      Dist            SoTA             CS          Expenditure
 Min.   :15.2   Min.   :2.132   Min.   : 1.00   Min.   :  0.00
 1st Qu.:16.7   1st Qu.:3.730   1st Qu.: 8.00   1st Qu.: 55.89
 Median :17.4   Median :4.395   Median :10.00   Median : 83.90
 Mean   :17.4   Mean   :4.310   Mean   :10.29   Mean   :107.51
 3rd Qu.:18.0   3rd Qu.:4.895   3rd Qu.:12.25   3rd Qu.:144.87
 Max.   :19.2   Max.   :6.921   Max.   :21.00   Max.   :630.25
    Arrivals          Income          Departures      Balance
 Min.   : 4.00   Min.   :  0.000   Min.   : 4.0   Min.   :-562.39
 1st Qu.:14.00   1st Qu.:  8.075   1st Qu.:14.0   1st Qu.: -94.95
 Median :17.50   Median : 31.725   Median :18.0   Median : -50.95
 Mean   :19.02   Mean   : 48.192   Mean   :18.8   Mean   : -59.30
 3rd Qu.:23.00   3rd Qu.: 68.195   3rd Qu.:23.0   3rd Qu.: -11.38
 Max.   :42.00   Max.   :277.500   Max.   :46.0   Max.   : 118.07
```

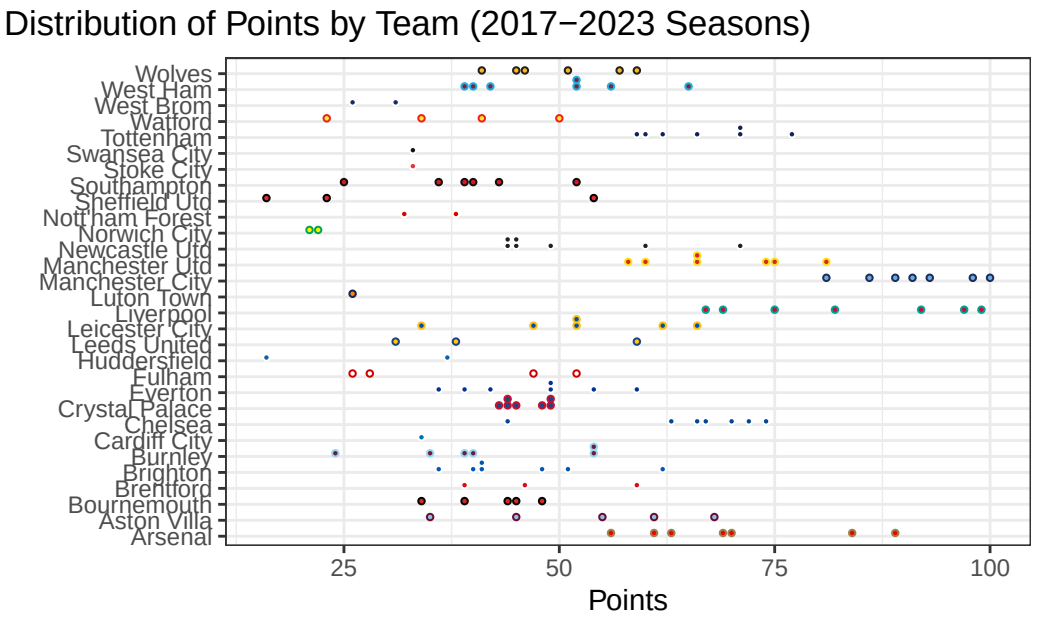
The summary statistics show that teams have an average of around 1.42 goals scored per game (GF) and a similar average of goals conceded (GA). Expected goals (xG) are generally slightly higher than actual goals, indicating that teams may underperform compared to expected chances. The average possession (Poss) ranges from 35.4% to 71%, suggesting a diverse range

of playing styles among teams. Net expenditure (Balance) also varies significantly, from highly negative values to positive, reflecting different transfer market strategies.

Distribution of Points by Team

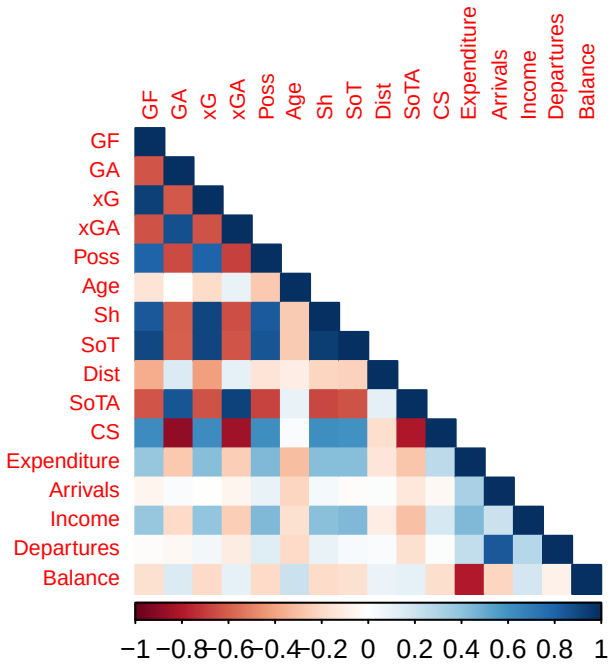
```
squad_colors <- read_csv(here::here("data", "prem_team_colors.csv"), show_col_types = FALSE)

prem %>%
  left_join(squad_colors, by = "Squad") %>%
  ggplot() +
  geom_dotplot(aes(x = Pts, y = Squad, fill = hex_fill, color = hex_color), binwidth = 1, do
  theme(legend.position = "none") +
  scale_fill_identity() +
  scale_color_identity() +
  theme_bw() +
  labs(
    x = "Points",
    y = "",
    title = "Distribution of Points by Team (2017-2023 Seasons)",
    caption = "Data from Fbref.com"
    ) +
  theme(plot.title.position = "plot")
```



Correlation Analysis

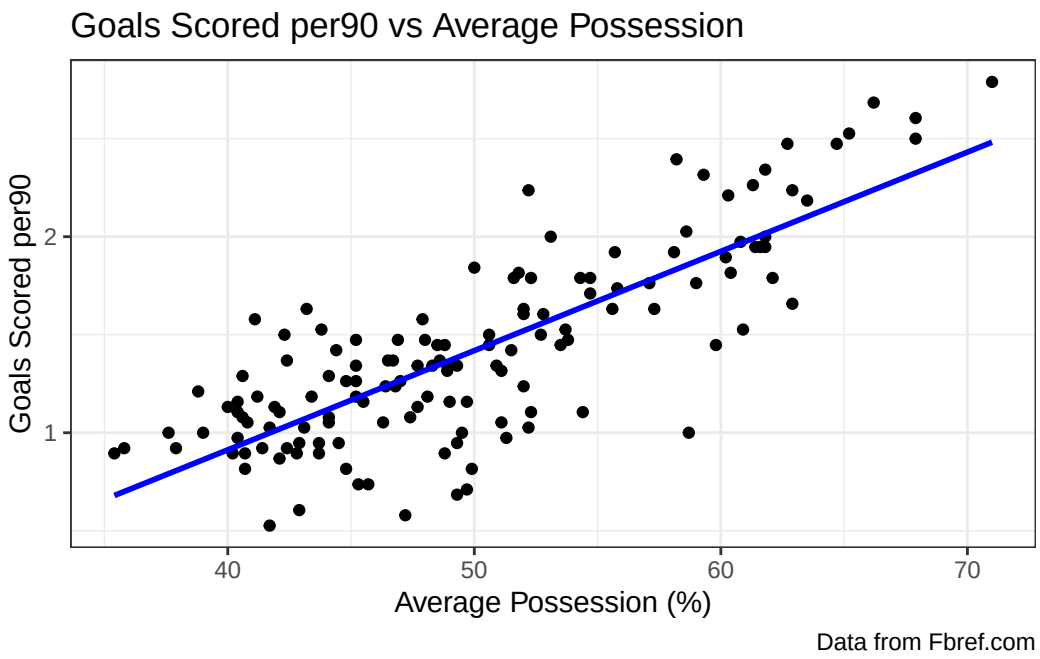
```
# Create a correlation plot to identify relationships between numeric variables
corr_matrix <- prem %>%
  select(GF, GA, xG, xGA, Poss, Age, Sh, SoT, Dist, SoTA, CS, Expenditure, Arrivals, Income,
  cor()
corrplot(corr_matrix, method = "color", type = "lower", tl.cex = 0.7, tl.pos = "lt")#, addCo
```



The correlation plot reveals some strong relationships between variables. Goals Scored (GF) has a strong positive correlation with Expected Goals (xG) (0.93) and Shots on Target (SoT) (0.91), indicating that creating high-quality chances and converting them are key factors for success. Conversely, Goals Against (GA) is negatively correlated with metrics like Clean Sheets (CS) (-0.88), suggesting that teams that concede fewer goals also have more clean sheets. However, Net Spend (Balance) has a weak correlation with point total (Pts), implying that spending money does not necessarily guarantee better results.

Possession vs Goals Scored

```
# Scatter plot for goals scored per90 vs average possession
ggplot(prem, aes(x = Poss, y = GF)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Goals Scored per90 vs Average Possession",
       x = "Average Possession (%)",
       y = "Goals Scored per90",
       caption = "Data from Fbref.com") +
  theme_bw()
```

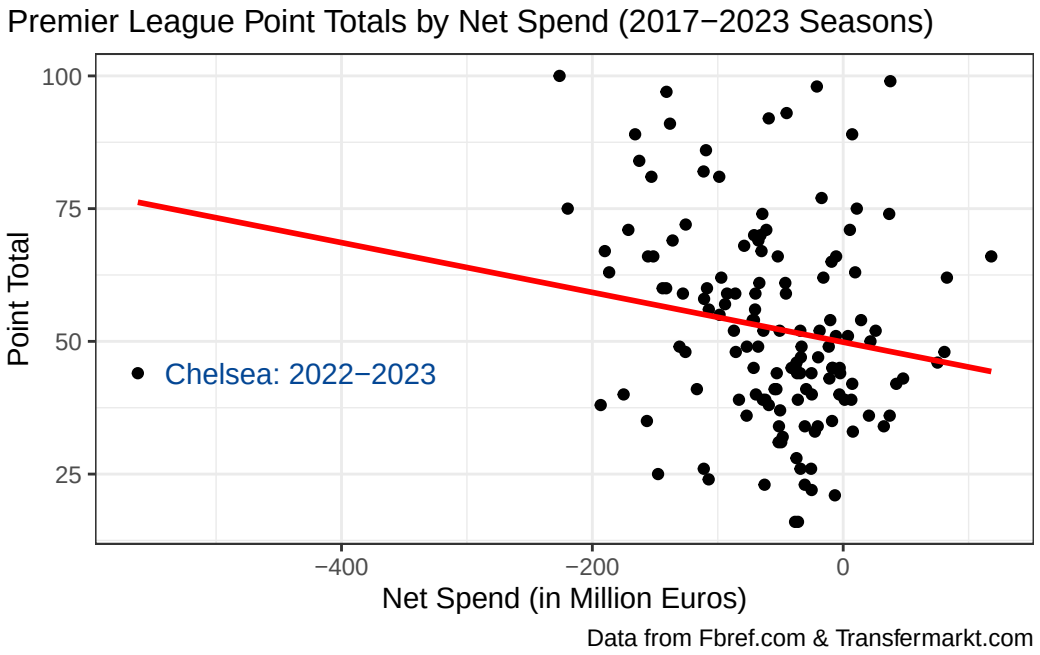


The scatter plot demonstrates a positive linear trend between possession and goals scored per game. Teams with higher possession percentages tend to score more goals, which suggests that controlling the game often leads to better attacking outcomes. However, there is still variability, indicating that other factors beyond possession influence goal-scoring effectiveness.

Net Spend vs Points

```
# Scatter plot for net spend vs point total
prem %>%
  mutate(label = ifelse(Balance < -400, paste0(Squad, ": ", Season), "")) %>%
  ggplot(aes(x = Balance, y = Pts)) +
    geom_point() +
    geom_text(aes(x = Balance, y = Pts, label = label), hjust = -0.1, color = "#034694") +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(title = "Premier League Point Totals by Net Spend (2017-2023 Seasons)",
         caption = "Data from Fbref.com & Transfermarkt.com",
         x = "Net Spend (in Million Euros)",
         y = "Point Total") +
  theme_bw() +
  theme(
    plot.title.position = "plot",
    plot.title = element_text(size = 12)
```

```
  )
```



The scatter plot between net spend and point total shows a slightly negative relationship, which is somewhat counter intuitive. This suggests that high spending teams do not always achieve higher point totals, potentially due to inefficiencies in spending or challenges in integrating new players. Teams with a lower or negative net spend can still achieve success, likely due to better tactical planning, consistency, and effective resource utilization.

Null Model + Shrinkage Plot

```
model0 <- lmer(Pts ~ 1 + (1 | Squad), data = prem)
summary(model0)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Pts ~ 1 + (1 | Squad)
   Data: prem

REML criterion at convergence: 1108.9

Scaled residuals:
```

```
     Min       1Q   Median       3Q      Max
-2.02395 -0.60329 -0.08726  0.65915  2.11491

Random effects:
 Groups   Name        Variance Std.Dev.
 Squad    (Intercept) 255.32   15.979
 Residual              99.66    9.983
Number of obs: 140, groups:  Squad, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)   47.388      3.091   15.33
```

```r
fits = predict(model0, prem); prem$fits <- fits
prem %>%
  group_by(Squad) %>%
  mutate(mean_pts = mean(Pts)) %>%
  ungroup() %>%
  ggplot(aes(y = Squad, x = mean_pts, group = Squad)) +
  geom_point() +
  geom_point(aes(y = Squad, x = fits),  col = "red") +
  geom_point(aes(x = Pts), col="grey") +
  theme_bw() +
  geom_vline(xintercept = mean(prem$Pts), col="black") +
  labs(y = "", x = "Points", title = "Random Effect Shrinkage") +
  theme(plot.title.position = "plot")
```

Random Effect Shrinkage