Premiere League Insights

Harman Singh Saggu T00727652, Raunaq Singh Dev T00737367, Nwaokenneya Precious T00727498

2023-11-30

Contents

1 Introduction

1.1 Overview

This project takes the data set of premier league teams from the 2020-21 season and uses the 6 teams based in London for its visualizations and analysis.

The focus will be on the following teams for the project:

1. Arsenal
2. Chelsea
3. West Ham
4. Crystal Palace
5. Brentford
6. Tottenham

This project will give us an insight of each teams general performance based on the different positions each player plays in and will be divided in the 3 following categories depending on the role of the players in the team:

1. Defenders (DF)
2. Midfielders (MF)
3. Forwards (FW)

The project is divided in 3 parts and each part will determine which players have statistically played well for the 2020-21 season in each of the three positions.

1.2 Data Description

The data has been sourced from the following link:

https://github.com/kedarghule/Premier-League-Player-Statistics-Dashboard/tree/main/datasets

The data includes details of performance of each of the six team players. It provides us with in depth information about various key statistics of individual performances of each player. We will be focusing on the following variables:

1. Player 2. Goals (Gls)
3. Assists (Ast)
4. Expected Goals (xG)
5. Expected Assists (xA)

6. Matches Played

1.3 Objective

This project aims to :

1. Find the best Defenders for the 2020-21 season.
2. Find the best Midfielders for the 2020-21 season.
3. Find the best Forwards for the 2020-21 season.
4. To see which team was statistically superior in the London region.

1.4 Tools Used

1. R Studio

2. R programming language

3. Data Visualization and Manipulation Techniques from ADSC 1010 course.

2 Data Manipulation

The data sets used in the project were initially divided team-wise and had to be cleaned and joined. To achieve the same the unnecessary columns were removed and then positions(DF, MF, FW) were assigned to each player judging there playing position in real life. Firstly, the data frames of each team were filtered by position (DF, MF, FW) and a final data frames were created. The following columns were then added to these new data frames:

1. Total Touches
2. Total Tackles
3. Total Pressures
4. Tackles per Match

Top five assist providers as well as goal scorers were filtered to create visualizations.

This project has 2 user defined functions:

1. The first function gives the summary of average minutes played, average tackles in the final 3rd, average goals scored and average assist of players for defenders and midfielders in each team.

2. The second function uses built-in spread function from tidyr package to extract total tackles of player of a specific team.

A user-defined loop was created using for loop to extract first 50 player names, team names, position and total touches for defenders and midfielders.

3 Visualizing Data

Visualizing data for defenders, midfielders and forwards using boxplot, scatterplot and barplot for further inference:

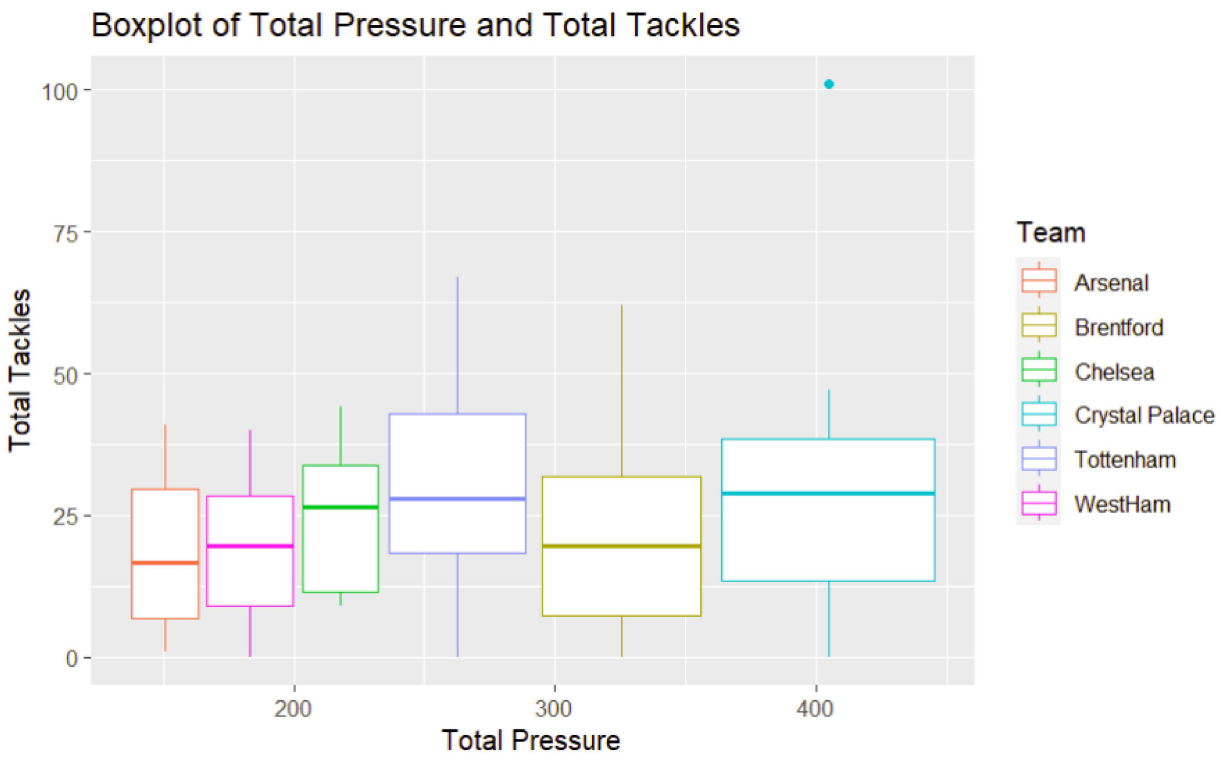

Figure 1: Total tackles and total pressures DF

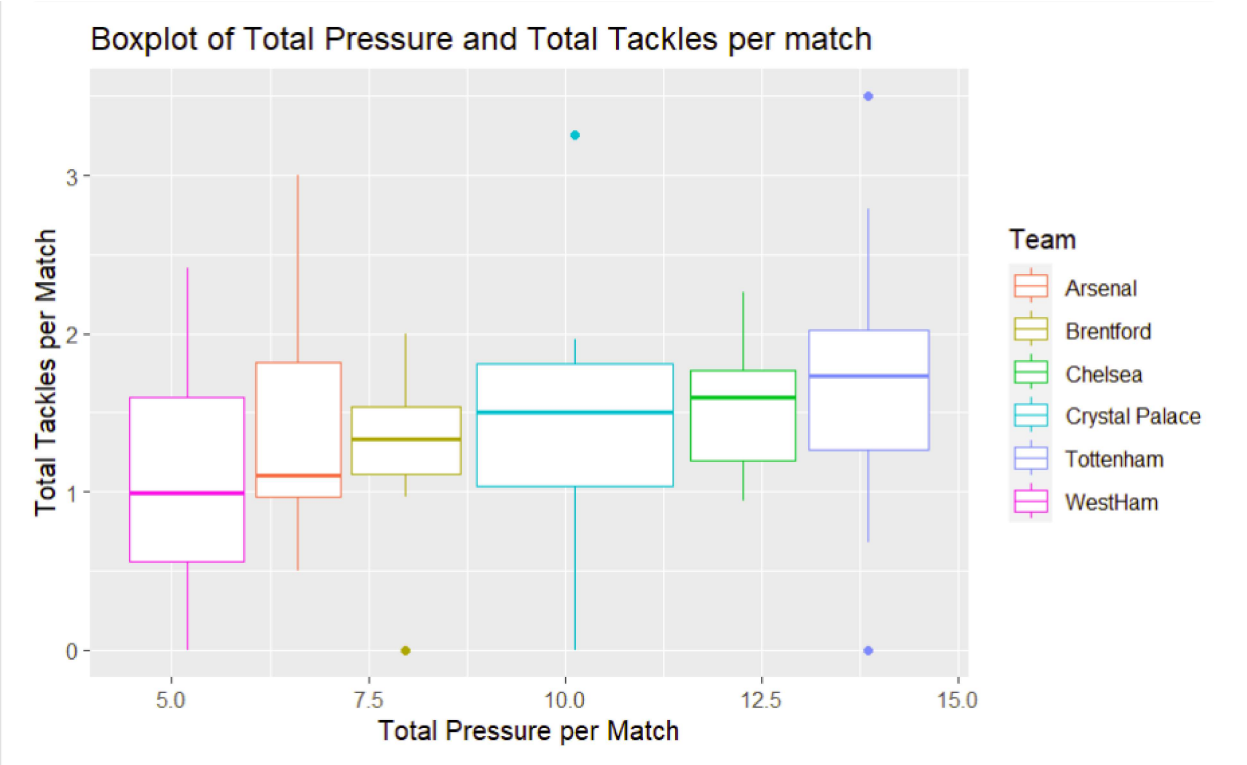

Figure 2: Tackles per match and pressures per match DF

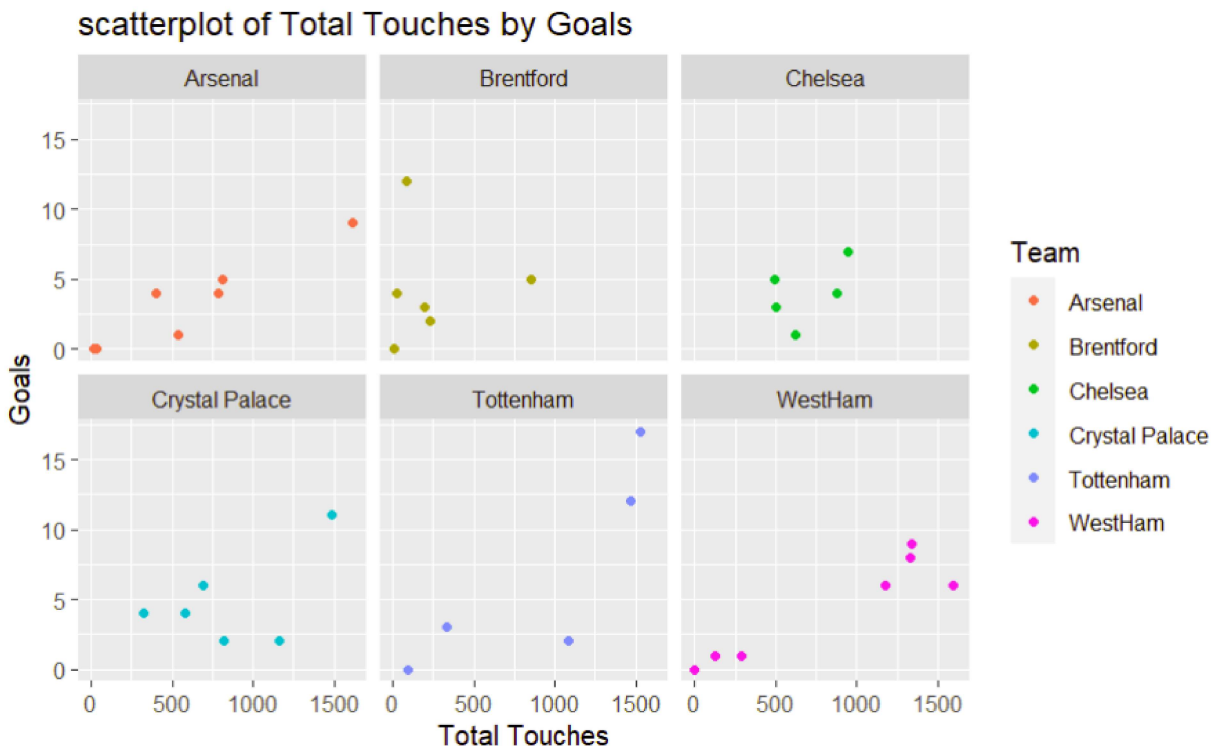

Figure 3: Total touches by goals per team for FW

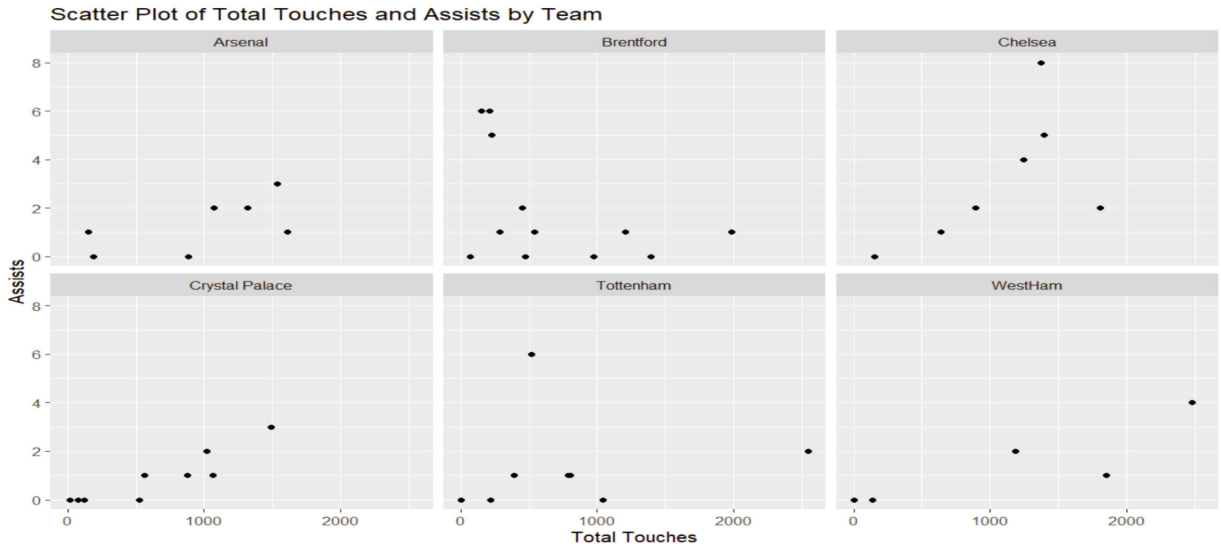

Figure 4: Total touches and assists by team for MF

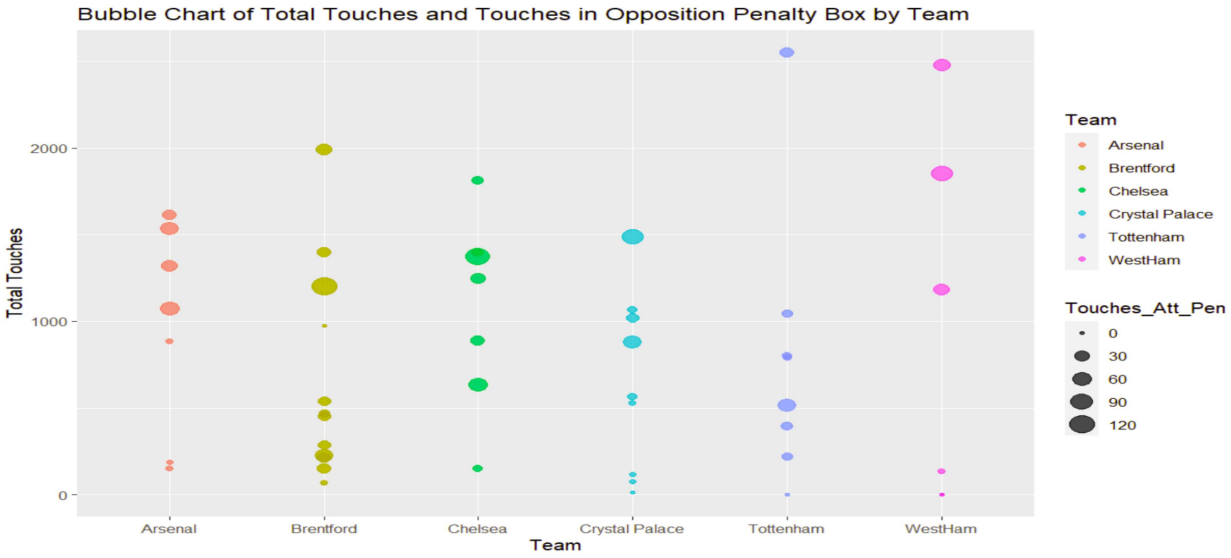

Figure 5: Bubble Chart of total touches and touches in opposition penalty box for midfielders

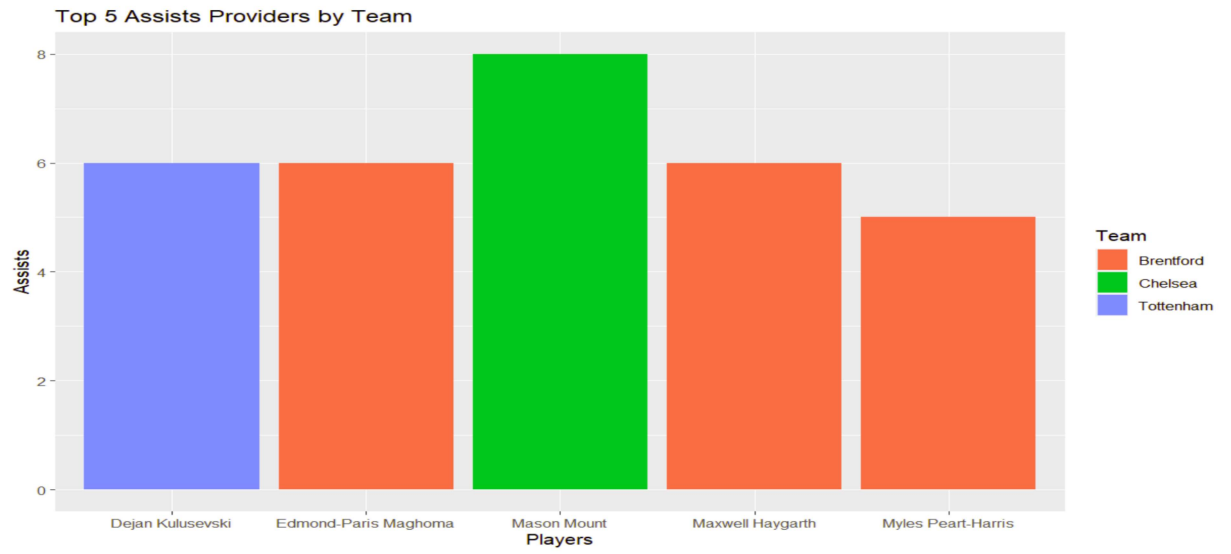

Figure 6: Top assist providers by team

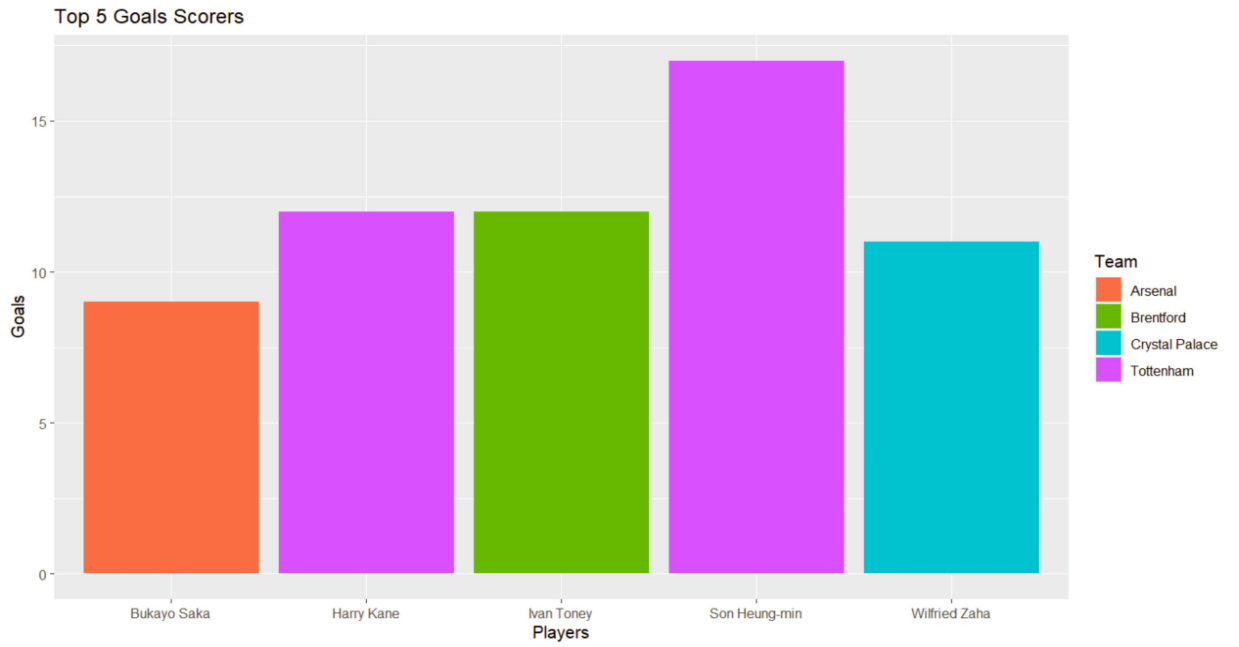

Figure 7: Top 5 goal scorers by team

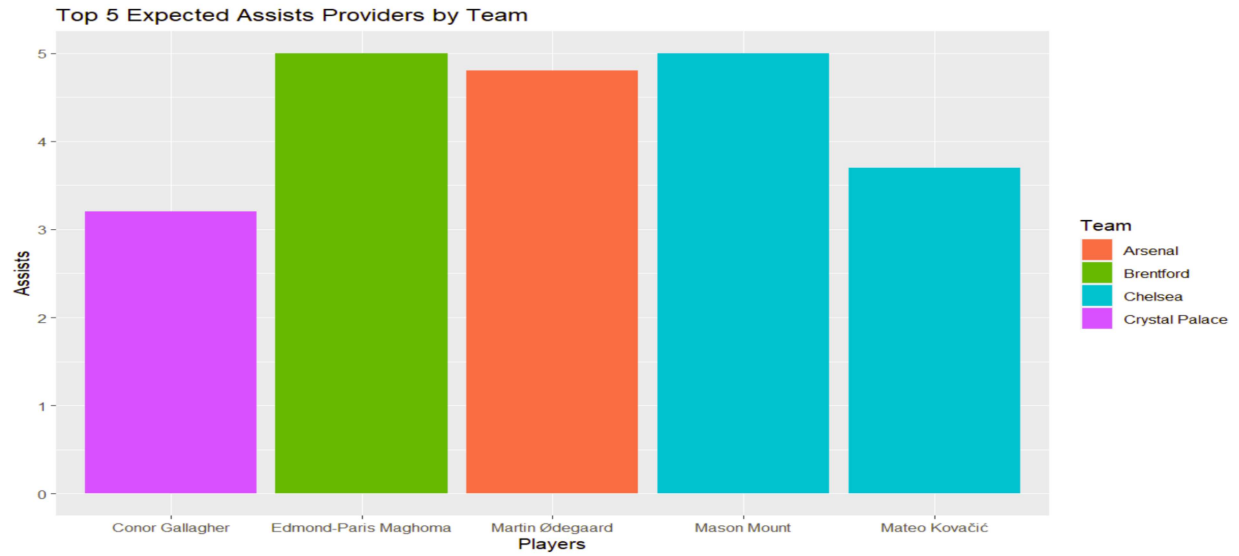

Figure 8: Top expected assist(xA) providers by team

4 Results

From boxplot of Total Pressure and Total Tackles it can be seen that **Tottenham** were better than the rest when it came to tackles. Crystal Palace were the most pressing team. Meanwhile, **Arsenal** were the worst team statistically when it came to total tackles commited and total pressure. **Crystal Palace** shows an outlier with outstanding defensive statistics and this player was **Tyrick Mitchell**. The boxplot of **Pressure and Tackles per Match** shows that **Tottenham** were the most aggressive team defensively again. **Arsenal** moved up to second last position as there per match tackles and pressing were better than West Ham, However, median tackles committed were almost same for both. **Tarique Fosu** from Brentford and **Joe Rodon** made 0 tackles but only played 1 match for 17 minutes and 79 minutes respectively. Tyrick Mitchell and Emerson were the best performers.

Scatterplot of total touches by goals shows that **Bukayo Saka** was the most involved as well as the top goalscorer for Arsenal. **Ivan Toney** is also a standout performer for his team but not as involved in the playmaking as the others. **Wilfried Zaha** was the talisman for **Crystal Palace** being heavily involved in the gameplay as well as goalscoring. The same goes for **Son Hueng-Min** for Tottenham who is also the most prolific goalscorer amongst everyone.

Barplot of top 5 assist providers by team shows top assist providers among the six teams. **Mason Mount** is number one assist provider. In terms of **expected assists** Mason Mount and Edmond Paris-Maghoma are tied for the top spot and **Martin Odegaard** is a close second which can be clearly seen in barplot of top expected assists. From these two barplots we can see that Mason Mount of Chelsea had the best conversion rate in terms of providing an assist whereas Martin Odegaard from Arsenal was expected to provide more assists than he did.

Bubble Chart of Total Touches and Touches in Opposition Penalty Box shows that **Pierre-Emile Hojberg** registered the most touches but had below average involvement inside the penalty box of the opposition. **Christian Eriksen** for Brentford was a threat inside the box and was also heavily involved in his teams gameplay. **Tomas Soucek** for West Ham registered 82 touches inside the penalty box and also had above average touches in total. Meanwhile, the **best player** again seemed to be Mason Mount from Chelsea.

5 Conclusions

From the results we can conclude that:

1. Tyrick Mitchell from Crystal Palace was found to be the highest presser and tackler in the whole season. His defensive stats were better than rest of the players.

2. Mason Mount from Chelsea was the best midfielder with the most assists and he was also heavily involved in his team's build-up game play.

3. Son Heung-Min from Tottenham was the most prolific goal scorer and also heavily involved creatively in his team's gameplay.

4. Tottenham was found to be the best team in the London region for 2020-2021 season.

5. Son Heung-Min was the overall best player because he had 10 assists and 17 goals.

List of Figures

6 Abbreviations

DF: Defenders

MF: Mid-Fielders

FW: Forwards

Gls: Goals

Ast: Assists

xA: Expected Assists

xG: Expected Goals

7 References

The data was sourced from https://github.com/kedarghule/Premier-League-Player-Statistics-Dashboard/tree/main/datasets

8 Appendix1: Libraries used

dplyr

ggplot2

tidyr

9 Appendix2: Roles of authors

Data Manipulation: Harman Singh Saggu,Raunaq Singh Dev,Nwaokenneya Precious

Data Visualization: Raunaq Singh Dev, Nwaokenneya Precious

Functions and reshaping data using tidyr: Harman Singh Saggu

Report: Harman Singh Saggu, Raunaq Singh Dev

10 Appendix3: Source code

```
knitr::opts_chunk$set(echo = FALSE)
library(dplyr)
library(ggplot2)
library(tidyverse)
library(plotly)
#Reading all csv files
arsenal <- read.csv("Arsenal.csv", header = TRUE)
brentford <- read.csv("Brentford.csv", header = TRUE)
chelsea <- read.csv("Chelsea.csv", header = TRUE)
crystalpalace <- read.csv("Crystal Palace.csv", header = TRUE)
tottenham <- read.csv("Tottenham.csv", header = TRUE)
westham <- read.csv("West Ham United.csv", header = TRUE)

#removing columns
ars <- arsenal[c(-28,-29),]
bre <- brentford[c(-28,-29),]
che <- chelsea[c(-28,-29),]
cry <- crystalpalace[c(-28,-29),]
tot <- tottenham[c(-28,-29),]
ham <- westham[c(-28,-29),]

#changing col names
colnames(bre)[9] <- "Pressure_Mid_3rd"
colnames(bre)[10] <- "Pressure_Att_3rd"
colnames(bre)[11] <- "Passing_Total_Cmp."

#arsenal position table
table(ars$Pos)

#changing position to DF
```

```
che[19,3]<-'DF'
che[23,3]<- 'DF'
tot[18,3]<- 'DF'
ham[16,3]<- 'DF'
#removing columns for final data frames to use
cry <- cry[,!(names(cry) %in% c("X90s","X90s.1"))]
ham <- ham[,!(names(ham) %in% "X90s")]

#changing positions to Mid Fielders
ars$Pos[10] <- "MF"
bre$Pos[21] <- "MF"
bre$Pos[24] <- "MF"
che$Pos[5] <- "MF"
che$Pos[17] <- "MF"
che$Pos[19] <- "MF"
tot$Pos[16] <- "MF"
tot$Pos[21] <- "MF"
cry$Pos[10] <- "MF"
cry$Pos[21] <- "MF"

#defenders
arsDF <- filter(ars, ars$Pos == 'DF')
breDF <- filter(bre,bre$Pos == 'DF')
cheDF <- filter(che, che$Pos=='DF')
cryDF <- filter(cry, cry$Pos=='DF')
totDF <- filter(tot,tot$Pos=='DF')
hamDF <- filter(ham,ham$Pos=='DF')



#histograms of DF
par(mfrow = c(2,2)) #for plotting side by side
hist(arsDF$Tackles_Def_3rd, main = "Histogram of Ars DF Tackle 3",
```

```
     xlab = "Defender Tackles")
hist(breDF$Tackles_Def_3rd, main = "Histogram of Bre DF Tackle 3",
     xlab = "Defender Tackles")
hist(cheDF$Tackles_Def_3rd, main = "Histogram of Che DF Tackle 3",
     xlab = "Defender Tackles")

#adding column for team names
arsDF <- arsDF%>% mutate(Team = "Arsenal")
breDF <- breDF%>% mutate(Team = "Brentford")
cheDF <- cheDF%>% mutate(Team = "Chelsea")
cryDF <- cryDF%>% mutate(Team = "Crystal Palace")
totDF <- totDF %>% mutate(Team = "Tottenham")
hamDF <- hamDF %>% mutate(Team = "WestHam")

#defenders data frame
Defenders <- rbind(arsDF, breDF, cheDF, cryDF, hamDF, totDF)

#adding total tackles, total pressure and total touches for defenders
Defenders <- Defenders%>%
  rowwise()%>%
  mutate(Total_Tackles = sum(c(Tackles_Att_3rd,Tackles_Def_3rd,Tackles_Mid.3rd)), Total_Pressure = sum(

#adding tackles per matches for defenders
vis_def <- Defenders%>%
  select(Player,Age,Team,Total_Tackles,Total_Pressure,Total_Touches,Matches_Played,Minutes_Played)%>%
  mutate(Tackles_per_matches = Total_Tackles/Matches_Played,Pressure_per_matches = Total_Pressure/Match

#boxplot for total pressure and total tackles defenders
par(mfrow=c(2,1))
ggplot(vis_def)+
  geom_boxplot(aes(x=Total_Pressure,y=Total_Tackles, color = Team))+
  labs(
```

```
    x = "Total Pressure",
    y = "Total Tackles",
    title = "Boxplot of Total Pressure and Total Tackles"
  )

#boxplot for pressure per matches and tackles per matches
ggplot(vis_def, aes(x=Pressure_per_matches, y = Tackles_per_matches, color = Team))+
  geom_boxplot()+
  labs(
    x = "Pressure per Match",
    y = "Tackles per Match",
    title = "Boxplot of Pressure and Tackles per Match "
  )
#scatterplot for matches played and total tackles defenders
scatterplot_DF <- ggplot(vis_def)+
  geom_point(aes(x=Matches_Played,y=Total_Tackles, color = Team))+
  labs(
    x = "Matches Played",
    y = "Total Tackles",
    title = "Scatterplot of Matches Played and Total Tackles"
  )



scatterplot_DF

#scatterplot of matches played and total pressure defenders
 pressure_scat <- ggplot(vis_def)+
  geom_point(aes(x=Matches_Played,y=Total_Pressure, color = Team))+
  labs(
    x = "Matches Played",
    y = "Total Pressure",
    title = "Scatterplot of Matches Played and Total Pressure"
```

```
  )


pressure_scat
#Changing position to Forward
ars$Pos[2] <- "FW"
ars$Pos[8] <- "FW"
ars$Pos[12] <- "FW"
ars$Pos[17] <- "FW"
bre$Pos[4] <- "FW"
bre$Pos[16] <- "FW"
bre$Pos[25] <- "FW"
che$Pos[8] <- "FW"
che$Pos[15] <- "FW"
che$Pos[16] <- "FW"
che$Pos[18] <- "FW"
tot$Pos[5] <- "FW"
tot$Pos[9] <- "FW"
tot$Pos[20] <- "FW"
tot$Pos[22] <- "FW"
cry$Pos[14] <- "FW"
ham$Pos[6] <- "FW"
ham$Pos[9] <- "FW"
ham$Pos[17] <- "FW"
ham$Pos[21] <- "FW"


#Forwards
arsFW <- filter(ars, ars$Pos == 'FW')
breFW <- filter(bre, bre$Pos == 'FW')
cheFW <- filter(che, che$Pos=='FW')
cryFW <- filter(cry, cry$Pos=='FW')
totFW <- filter(tot,tot$Pos=='FW')
hamFW <- filter(ham,ham$Pos=='FW')
```

```r
#adding column team name
arsFW <- arsFW%>% mutate(Team = "Arsenal")
breFW <- breFW%>% mutate(Team = "Brentford")
cheFW <- cheFW%>% mutate(Team = "Chelsea")
cryFW <- cryFW%>% mutate(Team = "Crystal Palace")
totFW <- totFW %>% mutate(Team = "Tottenham")
hamFW <- hamFW %>% mutate(Team = "WestHam")



#new table for all forwards
newFW <- rbind(arsFW, breFW, cheFW, cryFW, hamFW, totFW)

#adding total tackles, total pressures and total touches for Forwards
newFW <- newFW%>%
  rowwise()%>%
  mutate(Total_Tackles = sum(c(Tackles_Att_3rd,Tackles_Def_3rd,Tackles_Mid.3rd)), Total_Pressure = sum(

#arranging the forwards df
top_fw_plaayer <- head(arrange(newFW,desc(Gls)), 5)

#scatterplot for total touches by team Forwards
touches_goal <- ggplot(newFW,aes(x = Total_Touches,y = Gls, col = Team))+
  geom_point()+
  facet_wrap(~Team)+
  labs(
    x = "Total Touches",
    y = "Goals",
    title = "scatterplot of Total Touches by Goals"
  )

touches_goal
```

```r
top_fw <- ggplot(top_fw_plaayer, aes(x = Player, y = Gls, fill = Team)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    x = "Players",
    y = "Goals",
    title = "Top 5 Goals Scorers"
  )

top_fw
#MF
arsMF <- filter(ars, ars$Pos == 'MF')
breMF <- filter(bre,bre$Pos == 'MF')
cheMF <- filter(che, che$Pos=='MF')
cryMF <- filter(cry, cry$Pos=='MF')
totMF <- filter(tot,tot$Pos=='MF')
hamMF <- filter(ham,ham$Pos=='MF')

#adding column for team names
arsMF <- arsMF%>% mutate(Team = "Arsenal")
breMF <- breMF%>% mutate(Team = "Brentford")
cheMF <- cheMF%>% mutate(Team = "Chelsea")
cryMF <- cryMF%>% mutate(Team = "Crystal Palace")
totMF <- totMF %>% mutate(Team = "Tottenham")
hamMF <- hamMF %>% mutate(Team = "WestHam")

#mid fielders data frame
newMF <- rbind(arsMF, breMF, cheMF, cryMF, totMF, hamMF)

#adding total tackles, total prressures and total touches for midfielders
newMF <- newMF %>%
  rowwise() %>%
  mutate(Total_Tackles = sum(c(Tackles_Att_3rd,Tackles_Def_3rd,Tackles_Mid.3rd)), Total_Pressure = sum(
```

```r
top_players <- head(arrange(newMF, desc(Ast)), 5)


top_bar <- ggplot(top_players, aes(x = Player, y = Ast, fill = Team)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    x = "Players",
    y = "Assists",
    title = "Top 5 Assists Providers by Team"
  )

#arranging top players for midfielders
top_players2 <- head(arrange(newMF, desc(xA)), 5)

#barplot of top players by team midfielders position
top_bar2 <- ggplot(top_players2, aes(x = Player, y = xA, fill = Team)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    x = "Players",
    y = "Assists",
    title = "Top 5 Expected Assists Providers by Team"
  )



#bubble chart of total touches by team midfielders
bubble <- ggplot(newMF, aes(x = Team, y = Total_Touches, size = Touches_Att_Pen, color = Team)) +
  geom_point(alpha = 0.7) +
  labs(
    x = "Team",
    y = "Total Touches",
    title = "Bubble Chart of Total Touches and Touches in Opposition Penalty Box by Team"
  )
```

```r
#scatterplot of total touches by team midfielders
scatter_touches <- ggplot(newMF, aes(x = Total_Touches, y = Ast)) +
  geom_point() +
  facet_wrap(~Team) +
  labs(
    x = "Total Touches",
    y = "Assists",
    title = "Scatter Plot of Total Touches and Assists by Team"
  )
top_bar
top_bar2
bubble
scatter_touches
#creating a data frame with defenders and midfielders
df_mf <- rbind(Defenders, newMF)



#Creating a function to summarise average minutes played, average defender
#tackles(Tackles_Def_3rd), average goals(Gls) and average assists of players in
#defenders and mid-fileder positions
player_info <- function(team, position){
  df_mf %>%
    group_by(Pos = position) %>%
    filter(Team == team) %>%
    summarise(Player = Player, Team = Team, Avg_Def_Tackle = Tackles_Def_3rd/Matches_Played,
              Avg_Goals = Gls/Matches_Played, Avg_Assists = Ast/Matches_Played,
              Avg_Min_Played = as.numeric(Minutes_Played)/Matches_Played)
}

player_info("Brentford", "MF")
#Creating a for loop to extract first fifty player name, team name, position and
#total touches in df_mf data frame
```

```r
for(i in 1:50){
  player <- df_mf$Player[i]
  team <- df_mf$Team[i]
  pos <- df_mf$Pos[i]
  total_touches <- df_mf$Total_Touches[i]
  info <- c(player, team, total_touches, pos)
  print(info)
}
#Creating a function using  tidyr to make data wider in newFW data frame by
#extracting total tackles of players of a specific team
fw_wide <- function(team_name){
    select(newFW, Player, Team , Total_Tackles, Pos) %>%
  filter(Team == team_name) %>%
  spread(key = Player, value = Total_Tackles)
}

fw_wide("Chelsea")
fw_wide("Brentford")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
```