Random Analysis on English Premier league

Abinash Mallick

12/6/2021

##Premier League - The Stage

English Premier League - Most exciting domestic football league in the world. With the presence of superstars like Mo Salah, Cristiano Ronaldo, Harry Kane, Kavin De Bruyne and Acknowledgable minds in football world like Pep Guardiola, Jugern Klopp, Thomas Tuchel, Antonio Conte, Marcelo Bilesa it attracts viewrs from all over the world. Let's Analyse some of the pattern in this league with help of data. Let's dig in.

The dataset we're using is a dataset on kaggle containing information about more than 10,000 Premier League matches played. To get the dataset click here.

We will use R libraries "tidyverse", "tidyr", "psych", "ggsci", "ggpubr", "rshape" & "rshape2" in this project

Downloading and importing the Dataset to R Studio

we can download the data set from here. Then import it with below instruction.

```
EplData <- read.csv("EplResults93-21.csv", header = TRUE, sep = ",")
```

The data set is now imported to R Studio.

Data Preparation and Cleaning

In any data analysis project,It is very necessary to clean the real world raw data and ready it for our analysis. There could be wrong values, missing values that need to be dealt with. Along with that, we might want to add new columns which are useful for our analysis to the dataset, we might want to merge a few datasets together, this should be done as a preliminary step.

First we have to install all the libraries we are gonna use to perform our task. let's start

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
install.packages("tidyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(tidyr)
install.packages("ggsci")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(ggsci)
install.packages("ggpubr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(ggpubr)
install.packages("psych")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(dplyr)
install.packages("reshape2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(ggplot2)
```

Now let's clean the data In our case, there are lots of missing values. so we want to remove those rows from the data set.

```
clean <- drop_na(EplData)
```

The data set is now dropped all the missing values and store in the "clean" data set.

Next we don't need time and year in the DatetIime column. so we have to extract Date and Month and store those values in different column.

```
clean$DateTimeN <- strptime(clean$DateTime, format = "%Y-%m-%dT%H:%M:%SZ")
clean$Day <- format(clean$DateTimeN, "%d")
clean$Month <- format(clean$DateTimeN, "%m")
clean <- clean[, !(names(clean) %in% c("DateTime", "DateTimeN"))]
```

Now or data is clean and reay for the analysis.

Analysis and Visualization

Now that we have prepared our dataset, we can use it to get interesting analysis and visualisation. For that we will be using data visualisation libraries, ggplot2, ggpubr, ggsci.

Before plotting graphs, Let's first get some insights from our dataset using the describe() function provided by "psych" library.

```
pivot <- clean[ , c("Season", "FTHG", "FTAG", "HTHG", "HTAG", "HS", "AS", "HST", "AST", "HC", "AC", "HF
insight <- describe(pivot)
```

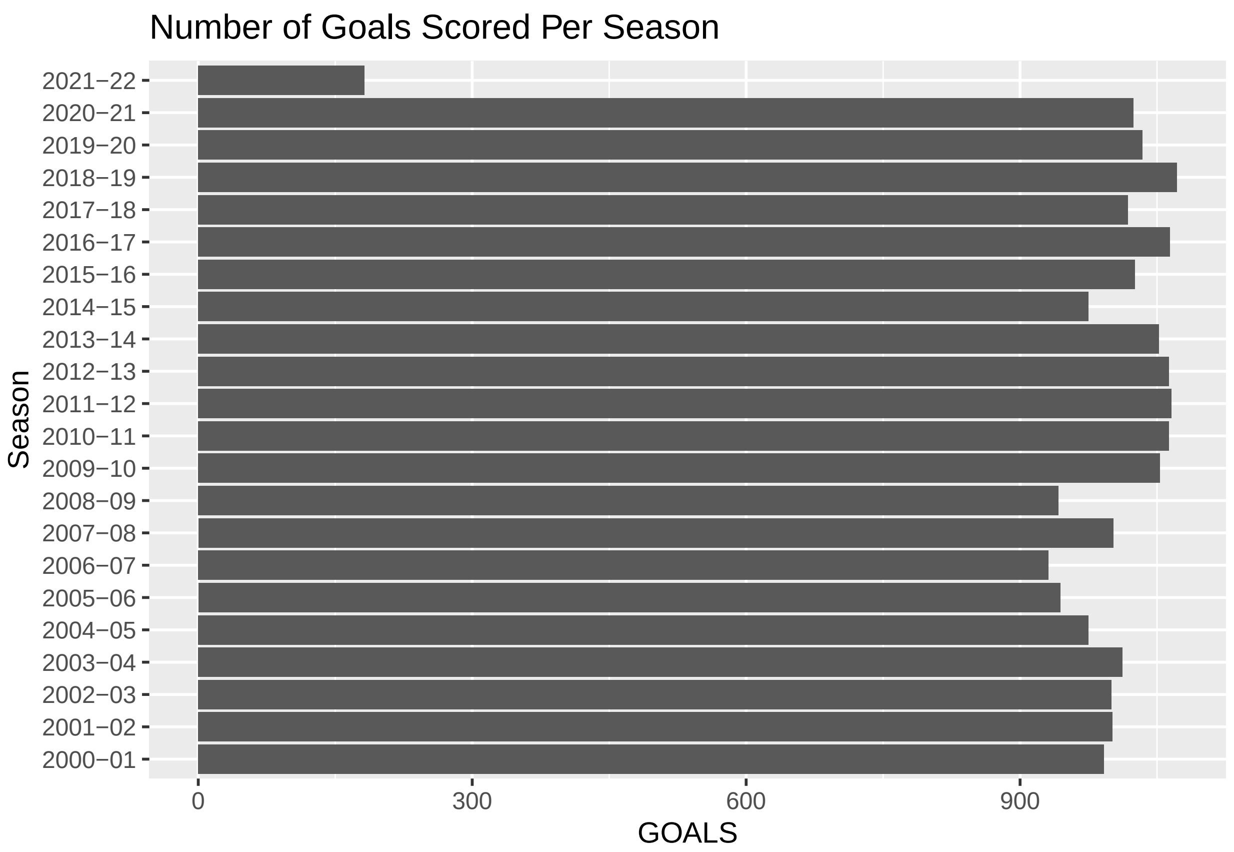
This gives us following preliminary insights.

1. On an average the home team scores more goals and takes more shots. 2. The away team commits more fouls. *3. Away team gets more yellow and red cards.* 4. Most goals scored by one team in a match are 9. *5. Most goals scored at hal time are 5.

Season-wise analysis- Goal scored per season

Let's look at the data and compare the trends based on season. Here we will first install the "ggsci" and "ggpubr" libraries and sort the data according to the season by using `aggregate()` function. Then we will plot a bar graph comparing Goals per Season.

```
seasonwise <- aggregate(cbind(FTHG, FTAG, HTHG, HTAG, HS, AS, HST, AST, HC, AC, HF, AF, HY, AY, HR, AR)
seasonwise <- seasonwise %>% mutate(GOALS=FTHG+FTAG)
ggplot(data=seasonwise, aes(x=GOALS, y=Season)) +
  geom_bar(stat = "identity") +
  labs(title = "Number of Goals Scored Per Season")
```



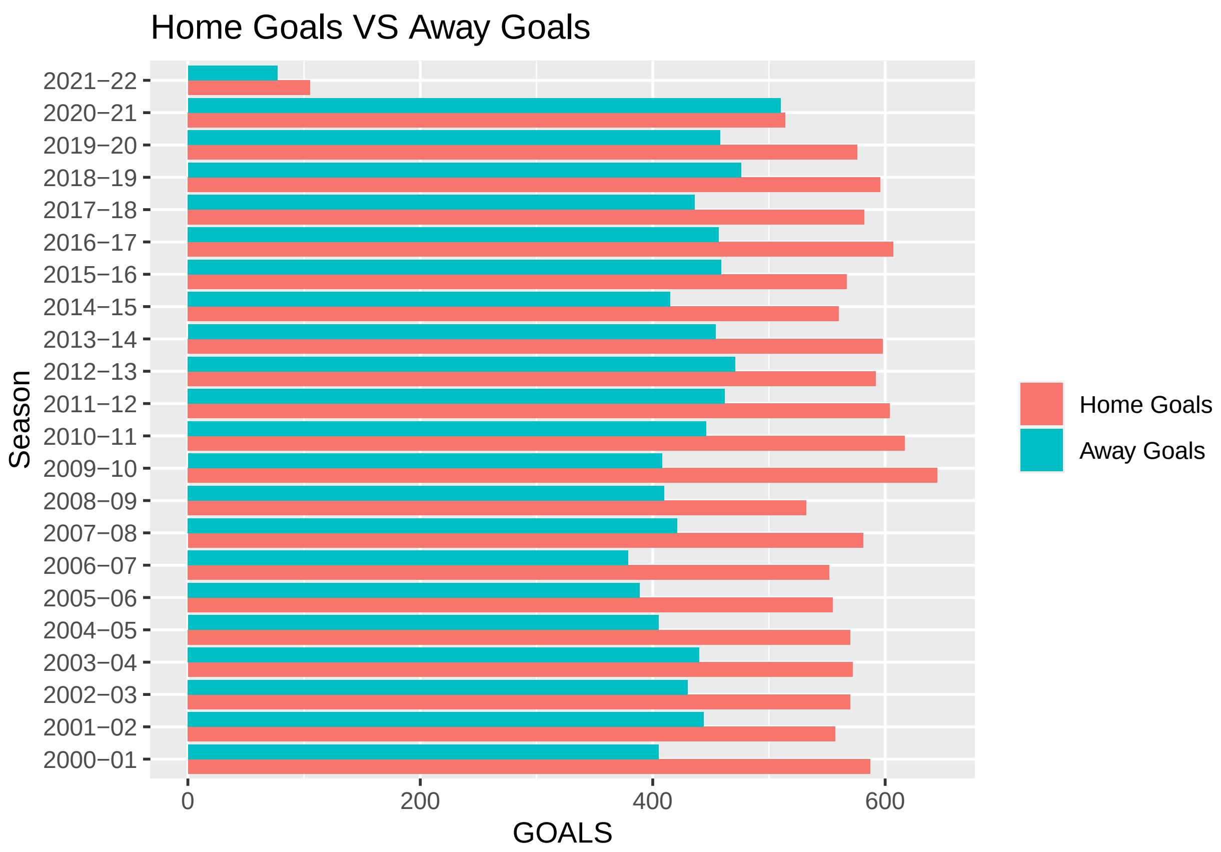
As we can see from the bar graph, 1. Number of goals scored per season have been fairly similar, around 1000. 2. 2018-19 season saw the highest number of goals scored. 3. 2005-06 season saw the lowest number of goals scored.

Home Goals vs Away Goals We will now plot a double barplot for the same using Home Goals and Away Goals. So we need to `melt` our data which lets us use the comparison parameter as hue.

```
Epl_melted <- seasonwise[ , c("Season", "FTHG", "FTAG")]
EPL_melt = melt(Epl_melted)
```

```
## Using Season as id variables
```

```
EPL_melt <- rename(EPL_melt, HA = variable, GOALS= value)
ggplot(data=EPL_melt, aes(x=GOALS, y=Season, fill=HA)) +
  geom_bar(stat = "identity", position = "dodge")+
  theme(legend.title = element_blank())+
  scale_fill_discrete(labels= c('Home Goals', 'Away Goals')) +
  labs(title = "Home Goals VS Away Goals")
```



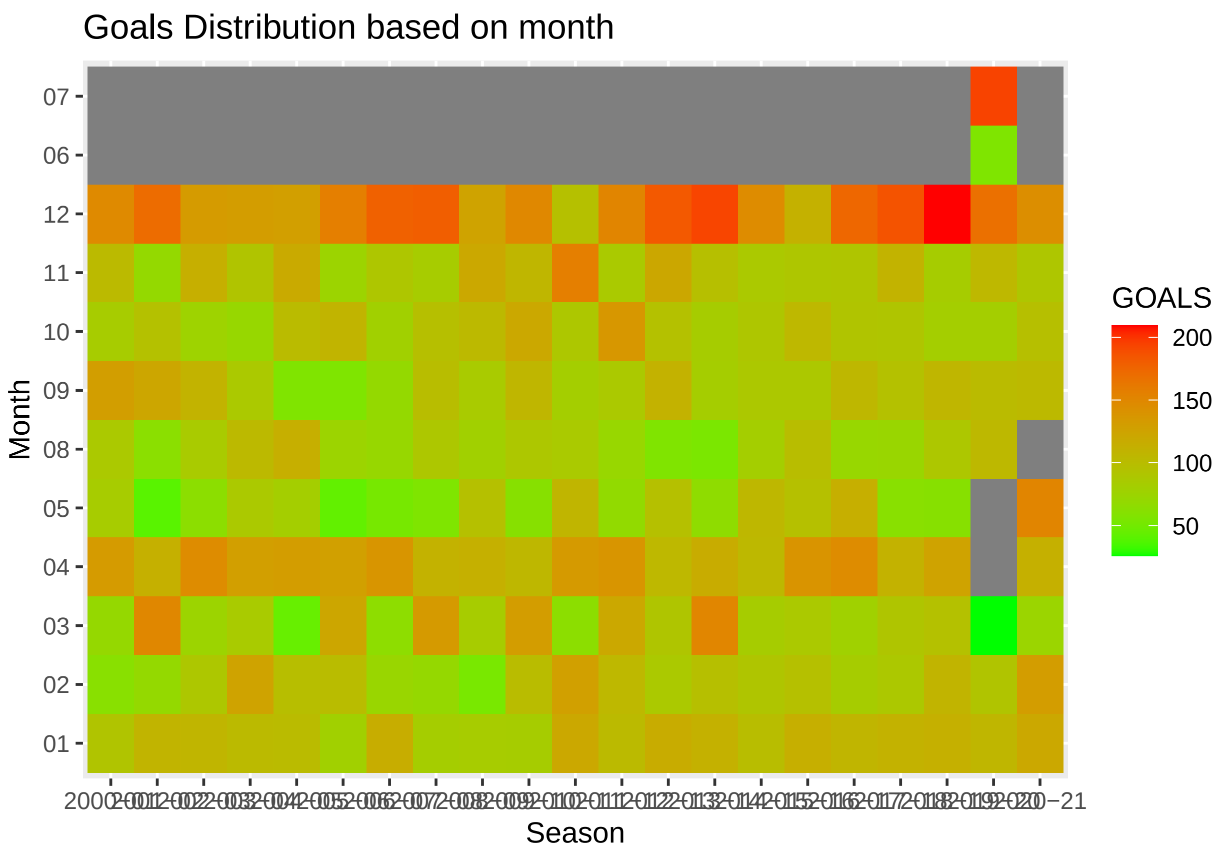
1. We can note that Goals scored by home team are almost always more than goals scored by away team. 2. In 2020-21 Season, the games were played behind close doors. The absense of home crowds clearly reflects in the stats as goals scored by away team are more than those scored by home team for this season.

Plotting a heatmap for distribution of goals per month Now let's prepare our data for a heatmap that shows us the goals scored per month with respect to each season. For that, we use a special funcion provided by reshape2 library called pivot_wider() which converts the dataframe into a 2D matrix, with first argument (series) being rows, 2nd argument being columns and third argument being the values corresponding to series 1 and series 2. Check it out below.

```
EPL_heat <- aggregate(cbind(FTHG, FTAG) ~ Month + Season, pivot, sum)
EPL_heat <- EPL_heat %>% mutate(GOALS=FTHG+FTAG)
EPL_heat <- EPL_heat[, !(names(EPL_heat) %in% c("FTHG", "FTAG"))]
EPL_heat <- pivot_wider(EPL_heat, names_from = Month, values_from = GOALS)
EPL_heat <- EPL_heat[-22,]
EPL_heat <- melt(EPL_heat)
```

```
## Using Season as id variables
```

```
colnames(EPL_heat) <- c("Season", "Month", "GOALS")
ggplot(EPL_heat, aes(Season, Month))+
  geom_tile(aes(fill= GOALS))+
  scale_fill_gradient(low = "green", high = "red")+
  labs(title ='Goals Distribution based on month' )
```



The above graphs tells us more about the data.

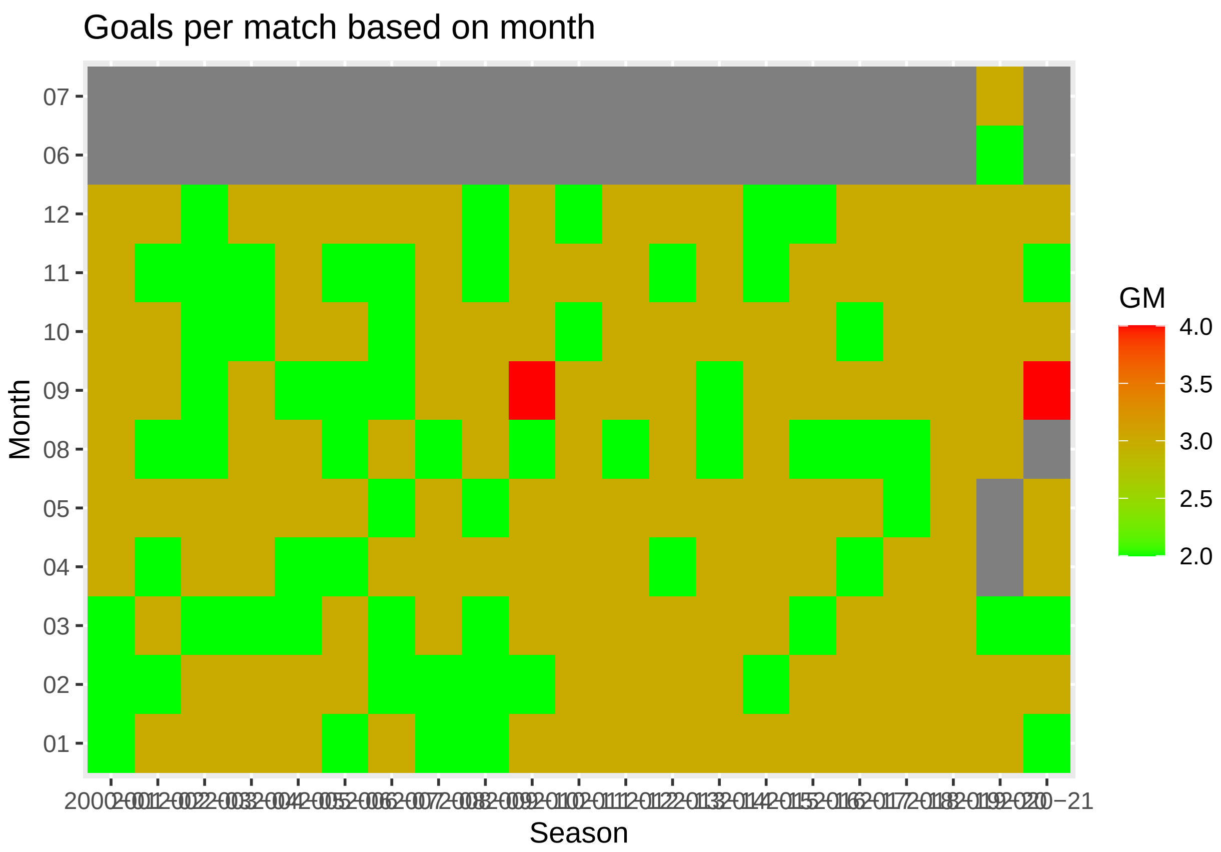
1. No games are played in the months of June and July, thus no goals are scored.
2. Almost every year, December has seen the most number of goals. This is because december sees the most number of matches played.
3. The 2019-20 season was disrupted due to the covid pandemic. Games were played in June and July instead of April and May under project restart.
4. The season ends mid-May, thus the low number of goals scored in May.

Let's plot another graph of Goals scored per game.

```
EPL_count <- aggregate(FTHG ~ Month + Season, pivot, length)
EPL_count <- pivot_wider(EPL_count, names_from = Month, values_from = FTHG)
EPL_count <- EPL_count[-22,]
EPL_count <- melt(EPL_count)
```

```
## Using Season as id variables
```

```
colnames(EPL_count) <- c("Season", "Month", "FTHG")
EPL_heat$GM = round(EPL_heat$GOALS / EPL_count$FTHG)
FINAL <- EPL_heat[, -3]
ggplot(EPL_heat, aes(Season, Month))+
  geom_tile(aes(fill= GM))+
  scale_fill_gradient(low = "green", high = "red")+
  labs(title = 'Goals per match based on month')
```



Here we can observe that Goals per game remains fairly obvious which shows that there are no real visible "Settling period", "Thriving period" or "Burnout period" or at least it doesn't affect the scoreline much. Pretty interesting that September witnessed the highest goals per game twice.

###COMEBACKS AFTER HALFTIME IN DIFFERENT CASES

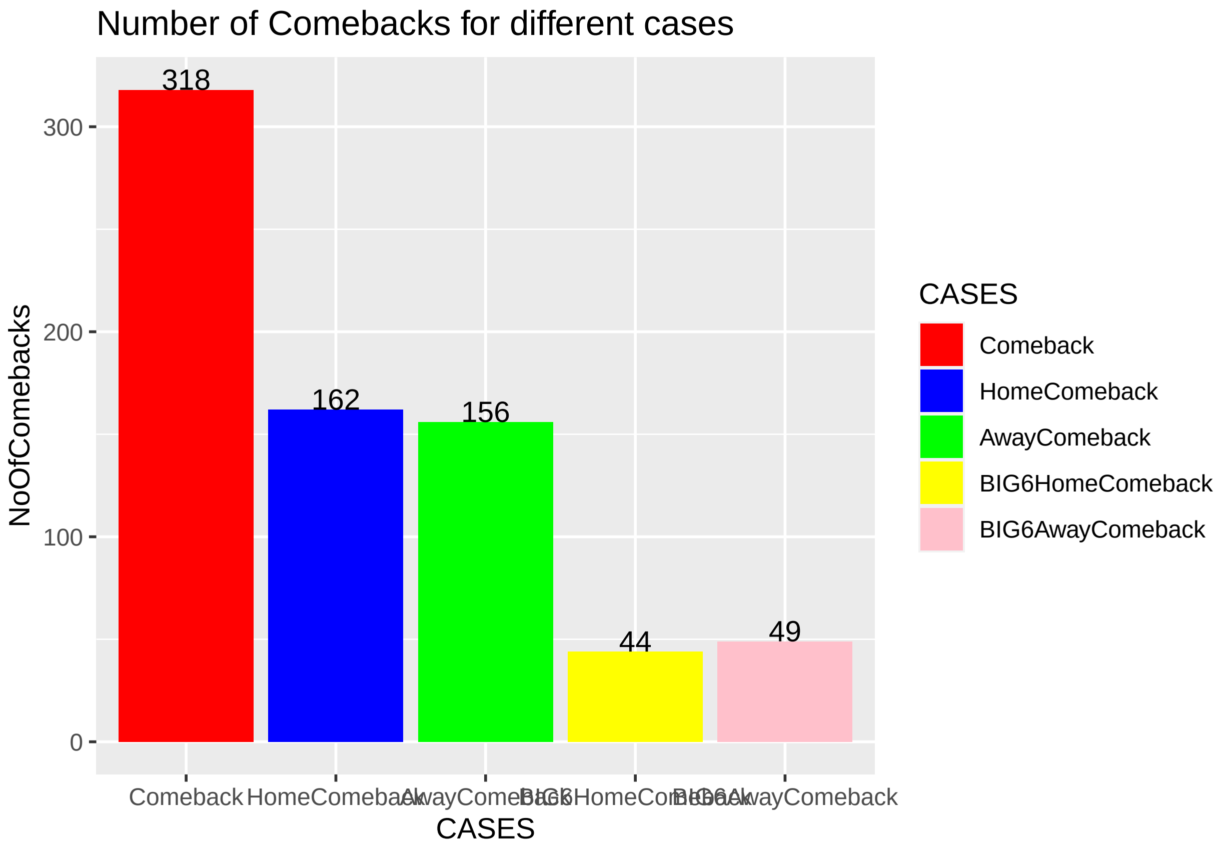
We are curious after seeing the previous analysis. Does Home crowd makes team comeback more often after ressults in halftime gone against them? Let's see!

we will use only data from 2015-16 season after Jurgen Klopp era strated at Anfield.

```
EPL15 <- clean %>% slice(c(5701:8050))
FPL_result <- EPL15[ , c("Season", "HomeTeam", "AwayTeam", "HTR", "FTR")]
FPL_result <- FPL_result[FPL_result$HTR != "D", ]
FPL_result$Results <- if_else(FPL_result$HTR == FPL_result$FTR, "No Comeback", "Comeback")
COMEBACKS <- FPL_result[FPL_result$Results == "Comeback", ]
NOCOMEBACKS <- FPL_result[FPL_result$Results == " No Comeback", ]
HCOMEBACKS <- COMEBACKS[COMEBACKS$HTR == "A", ]
ACOMEBACKS <- COMEBACKS[COMEBACKS$HTR == "H", ]
ANALYSIS <- data.frame("Comeback", "HomeComeback", "AwayComeback")
ANALYSIS$Comeback <- length(COMEBACKS$Results == "Comeback" )
ANALYSIS$HomeComeback <- length(HCOMEBACKS$Results == "Comeback")
ANALYSIS$AwayComeback <- length(ACOMEBACKS$Results == "Comeback")
ANALYSIS <- ANALYSIS[ -c(1:3)]
BHCOMEBACKS <- HCOMEBACKS[HCOMEBACKS$HomeTeam %in% c("Arsenal", "Liverpool", "Chelsea", "Man United", "
BACOMEBACKS <- ACOMEBACKS[ACOMEBACKS$HomeTeam %in% c("Arsenal", "Liverpool", "Chelsea", "Man United", "
ANALYSIS$BIG6HomeComeback <- length(BHCOMEBACKS$Results == "Comeback")
ANALYSIS$BIG6AwayComeback <- length(BACOMEBACKS$Results == "Comeback")
ANALYSIS <- melt(ANALYSIS)
```

```
## No id variables; using all as measure variables
```

```
colnames(ANALYSIS) <- c("CASES", "NoOfComebacks")
ggplot(data=ANALYSIS, aes(x=CASES, y=NoOfComebacks, fill= CASES))+
  geom_bar(stat = "identity")+
  labs(title = "Number of Comebacks for different cases")+
  scale_fill_manual(values = c("Comeback" = "red",
                               "HomeComeback" = "blue",
                               "AwayComeback" = "green",
                               "BIG6HomeComeback" = "yellow",
                               "BIG6AwayComeback" = "pink"))+
  geom_text(aes(label = NoOfComebacks), vjust = 0)
```



We can see out of 2350 games played in this period only in 318 cases we see a comeback. So comeback is not easy! it shows the toughness of the EPL in this era. *Out of 318 comebacks home team comebacks 168 times ansd away team comebacks 156 times. So in this case there is no significant effect of home crowds.* Out of 168 home comebacks BIG6 only managed 44 and out of 156 away comebacks they only managed 49. Which is suprise as BIG6 managed more comebacks in away matches rather than home matches. This favour the point no significant home crowds effect if the oposition played quality football on that day.

Conclusion

As entertaining as Premier League is to watch, the data surrounding it is interesting to dive into. We saw how stats from different seasons compare, how different teams fare in the Premier League. We got a jist of what Premier League is and how it works. We saw some interesting trends also.

The main objective of this article was to find ot does Home crowds really affect the result of the match. As we here find in our analysis some mixed opinion we will do another in depth analysis on it!