

NBA data analysis

Rohan Jayaram, Mahin Kadabi, Nate DeSisto, Hari Purnapatre, Rucha Dandavate

9/16/2021

We have Used this Reproducible Source code “https://raw.githubusercontent.com/ixarchakos/nba-games/master/data/nba_games_2015_2016.csv”

Short Description:-

The National Basketball Association is a professional basketball league in the United States. It consists of thirty teams distributed into two conferences. Each year, each team plays 82 regular season games, followed by playoffs and championship.

We have taken data from the 2015-2016 regular season and will be taking an in depth look at it. We want to analyze the data to be able to answer a few questions. We are interested in finding the factors that help contribute to a win. We will be analyzing halftime data, home and away teams, and different box score statistics including: three pointers, rebounds, etc.

```
library(tidyverse)

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

## v ggplot2 3.3.5      v purrr   0.3.4
## v tibble  3.1.2      v dplyr  1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

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

library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
## lift
```

```
library(dplyr)
```

Reading the data and getting idea of all the columns and rows.

```
data_url <- url("https://raw.githubusercontent.com/ixarchakos/nba-games/master/data/nba_games_2015_2016")
data <- read_csv(data_url)
```

```
##
## -- Column specification -----
## cols(
##   .default = col_double(),
##   date = col_date(format = ""),
##   home_team = col_character(),
##   away_team = col_character(),
##   half_time_score = col_character(),
##   fg_made_attempted_home = col_character(),
##   fg_made_attempted_away = col_character(),
##   '3pt_made_attempted_home' = col_character(),
##   '3pt_made_attempted_away' = col_character(),
##   ft_made_attempted_home = col_character(),
##   ft_made_attempted_away = col_character()
## )
## i Use 'spec()' for the full column specifications.
```

```
nba.data <- as_tibble(data)
df <- as_tibble(data)
```

After looking at the data we need to Separate the Half time score column to two different columns namely home half time score and away half time score.

```
nba.data <- nba.data %>%
  separate(half_time_score, c("home_half_time_score", "away_half_time_score"), sep = "-")
nba.data
```

```
## # A tibble: 1,230 x 48
##       id date      home_team home_points away_team away_points
##   <dbl> <date>    <chr>         <dbl> <chr>         <dbl>
## 1 400827888 2015-10-27 ATL             94 DET             106
## 2 400827889 2015-10-27 CHI             97 CLE             95
## 3 400827890 2015-10-27 GS             111 NO              95
## 4 400827891 2015-10-28 ORL             87 WSH             88
## 5 400827892 2015-10-28 BOS             112 PHI             95
## 6 400827893 2015-10-28 BKN             100 CHI            115
## 7 400827894 2015-10-28 DET             92 UTAH             87
```

```
## 8 400827895 2015-10-28 MIA          104 CHA          94
## 9 400827896 2015-10-28 TOR          106 IND          99
## 10 400827897 2015-10-28 HOU          85 DEN         105
## # ... with 1,220 more rows, and 42 more variables: home_half_time_score <chr>,
## #   away_half_time_score <chr>, fg_made_attempted_home <chr>,
## #   fg_made_attempted_away <chr>, field_goal_%_home <dbl>,
## #   field_goal_%_away <dbl>, 3pt_made_attempted_home <chr>,
## #   3pt_made_attempted_away <chr>, three_point_%_home <dbl>,
## #   three_point_%_away <dbl>, ft_made_attempted_home <chr>,
## #   ft_made_attempted_away <chr>, free_throw_%_home <dbl>,
## #   free_throw_%_away <dbl>, total_rebounds_home <dbl>,
## #   total_rebounds_away <dbl>, offensive_rebounds_home <dbl>,
## #   offensive_rebounds_away <dbl>, defensive_rebounds_home <dbl>,
## #   defensive_rebounds_away <dbl>, team_rebounds_home <dbl>,
## #   team_rebounds_away <dbl>, assists_home <dbl>, assists_away <dbl>,
## #   steals_home <dbl>, steals_away <dbl>, blocks_home <dbl>, blocks_away <dbl>,
## #   turnovers_home <dbl>, turnovers_away <dbl>, points_home <dbl>,
## #   points_away <dbl>, fast_break_points_home <dbl>,
## #   fast_break_points_away <dbl>, points_in_paint_home <dbl>,
## #   points_in_paint_away <dbl>, personal_fouls_home <dbl>,
## #   personal_fouls_away <dbl>, technical_fouls_home <dbl>,
## #   technical_fouls_away <dbl>, flagrant_fouls_home <dbl>,
## #   flagrant_fouls_away <dbl>
```

Adding new columns, if the home points are greater than away points then the winning team will be Home team else away team.

```
nba.data <- nba.data %>%
  mutate(nba.data, H_or_A_win = ifelse(home_points > away_points, 'H', 'A'), H_or_A_halftime = ifelse(h
nba.data
```

```
## # A tibble: 1,230 x 51
##       id date      home_team home_points away_team away_points
##       <dbl> <date>      <chr>          <dbl> <chr>          <dbl>
## 1 400827888 2015-10-27 ATL             94 DET             106
## 2 400827889 2015-10-27 CHI             97 CLE             95
## 3 400827890 2015-10-27 GS              111 NO              95
## 4 400827891 2015-10-28 ORL             87 WSH             88
## 5 400827892 2015-10-28 BOS             112 PHI             95
## 6 400827893 2015-10-28 BKN             100 CHI             115
## 7 400827894 2015-10-28 DET             92 UTAH             87
## 8 400827895 2015-10-28 MIA             104 CHA             94
## 9 400827896 2015-10-28 TOR             106 IND             99
## 10 400827897 2015-10-28 HOU             85 DEN            105
## # ... with 1,220 more rows, and 45 more variables: home_half_time_score <chr>,
## #   away_half_time_score <chr>, fg_made_attempted_home <chr>,
## #   fg_made_attempted_away <chr>, field_goal_%_home <dbl>,
## #   field_goal_%_away <dbl>, 3pt_made_attempted_home <chr>,
## #   3pt_made_attempted_away <chr>, three_point_%_home <dbl>,
## #   three_point_%_away <dbl>, ft_made_attempted_home <chr>,
## #   ft_made_attempted_away <chr>, free_throw_%_home <dbl>,
## #   free_throw_%_away <dbl>, total_rebounds_home <dbl>,
```

```
## # total_rebounds_away <dbl>, offensive_rebounds_home <dbl>,
## # offensive_rebounds_away <dbl>, defensive_rebounds_home <dbl>,
## # defensive_rebounds_away <dbl>, team_rebounds_home <dbl>,
## # team_rebounds_away <dbl>, assists_home <dbl>, assists_away <dbl>,
## # steals_home <dbl>, steals_away <dbl>, blocks_home <dbl>, blocks_away <dbl>,
## # turnovers_home <dbl>, turnovers_away <dbl>, points_home <dbl>,
## # points_away <dbl>, fast_break_points_home <dbl>,
## # fast_break_points_away <dbl>, points_in_paint_home <dbl>,
## # points_in_paint_away <dbl>, personal_fouls_home <dbl>,
## # personal_fouls_away <dbl>, technical_fouls_home <dbl>,
## # technical_fouls_away <dbl>, flagrant_fouls_home <dbl>,
## # flagrant_fouls_away <dbl>, H_or_A_win <chr>, H_or_A_halftime <chr>,
## # winning_team <chr>
```

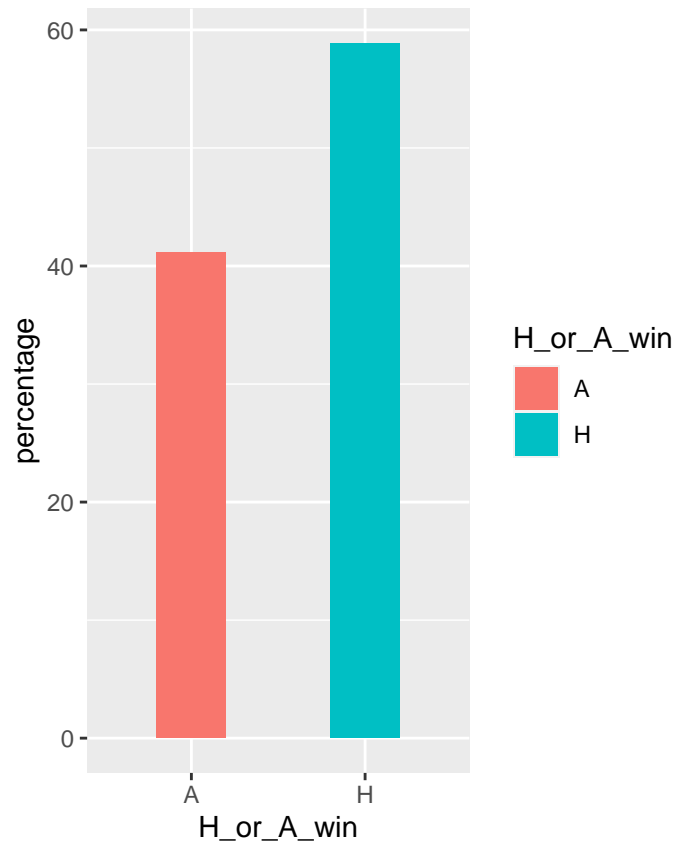
Cleaning the Dataset, eliminating missing values, errors.

```
nba.data.clean <- nba.data %>%
  select(id,date, home_team, home_points, away_team, away_points, winning_team ,H_or_A_win, home_half_t.
nba.data.clean
```

```
## # A tibble: 1,230 x 13
##       id date      home_team home_points away_team away_points winning_team
##       <dbl> <date>      <chr>         <dbl> <chr>         <dbl> <chr>
## 1 400827888 2015-10-27 ATL             94 DET             106 DET
## 2 400827889 2015-10-27 CHI             97 CLE             95 CHI
## 3 400827890 2015-10-27 GS              111 NO              95 GS
## 4 400827891 2015-10-28 ORL             87 WSH             88 WSH
## 5 400827892 2015-10-28 BOS             112 PHI             95 BOS
## 6 400827893 2015-10-28 BKN             100 CHI            115 CHI
## 7 400827894 2015-10-28 DET             92 UTAH             87 DET
## 8 400827895 2015-10-28 MIA             104 CHA             94 MIA
## 9 400827896 2015-10-28 TOR             106 IND             99 TOR
## 10 400827897 2015-10-28 HOU             85 DEN             105 DEN
## # ... with 1,220 more rows, and 6 more variables: H_or_A_win <chr>,
## #   home_half_time_score <chr>, away_half_time_score <chr>,
## #   H_or_A_halftime <chr>, three_point_%_home <dbl>, three_point_%_away <dbl>
```

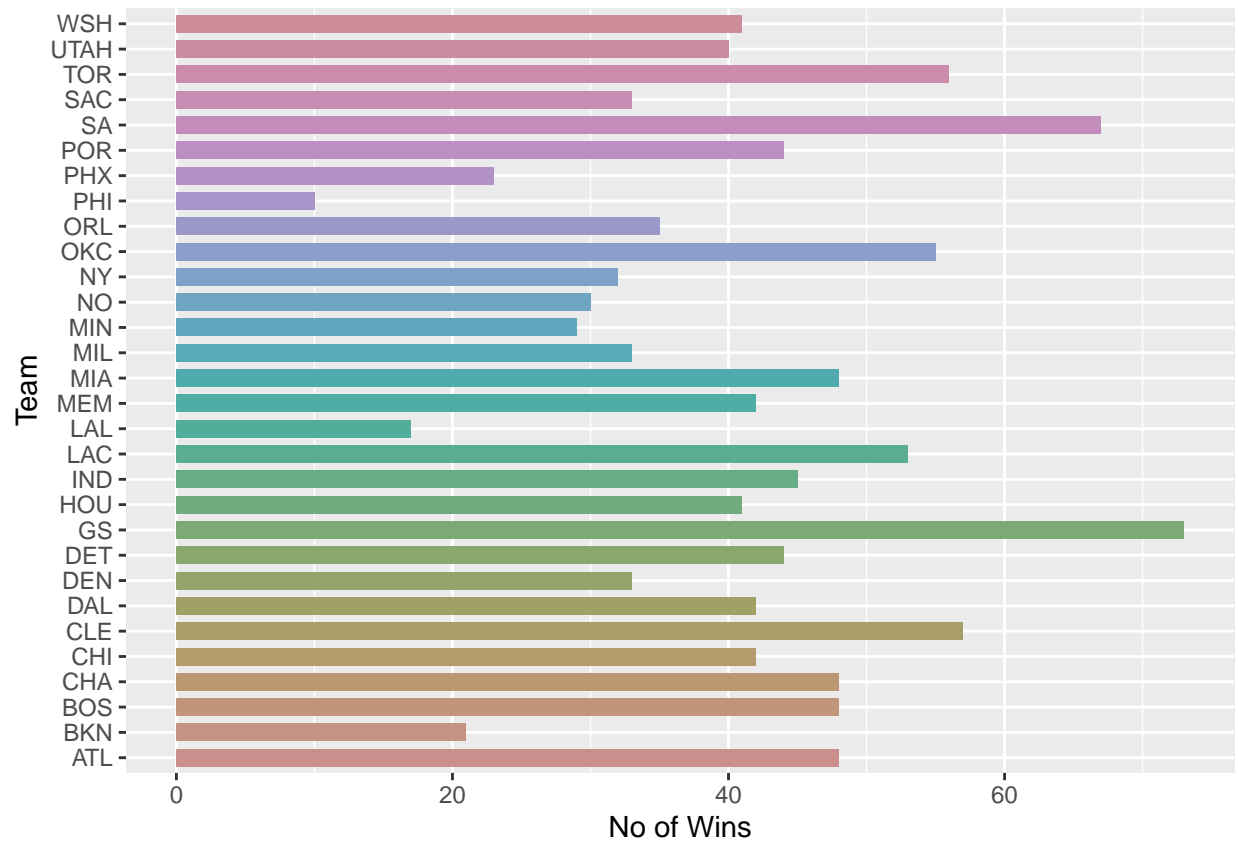
The Bar Chart represents the percentage of home team and away team winning.

```
counttable <- dplyr::count(nba.data.clean, H_or_A_win)
counttable <- counttable %>%
  mutate(percentage = n/sum(n)*100) %>%
  ggplot(aes(x = H_or_A_win, y = percentage, fill=H_or_A_win)) + geom_bar(stat="identity", width = 0.4)
  theme(aspect.ratio = 2/1)
counttable
```



The Graph gives an overview of all Teams and there number of wins.

```
barplot_winning_team <-
  ggplot(nba.data.clean, aes(x = winning_team, fill = winning_team)) + geom_bar(width = 0.7) + coord_flip() +
  scale_fill_hue(c = 40) +
  theme(legend.position="none") + labs(y = 'No of Wins', x = 'Team')
barplot_winning_team
```



Half time analysis

```
df = as_tibble(nba.data)
halftime = df %>%
  separate(fg_made_attempted_home, into = c("fg_made_home", "fg_attempts_home")) %>%
  separate(fg_made_attempted_away, into = c("fg_made_away", "fg_attempts_away")) %>%
  separate(`3pt_made_attempted_home`, into = c("3pt_made_home", "3pt_attempts_home")) %>%
  separate(`3pt_made_attempted_away`, into = c("3pt_made_away", "3pt_attempts_away")) %>%
  separate(ft_made_attempted_home, into = c("ft_made_home", "ft_attempts_home")) %>%
  separate(ft_made_attempted_away, into = c("ft_made_away", "ft_attempts_away"))
```

```
halftime2 = halftime %>%
  mutate_if(is.character, as.integer) %>%
  mutate(point_difference = home_points - away_points) %>%
  mutate(final_score_home = home_points - away_points) %>%
  mutate(halftime_difference_home = home_half_time_score - away_half_time_score)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

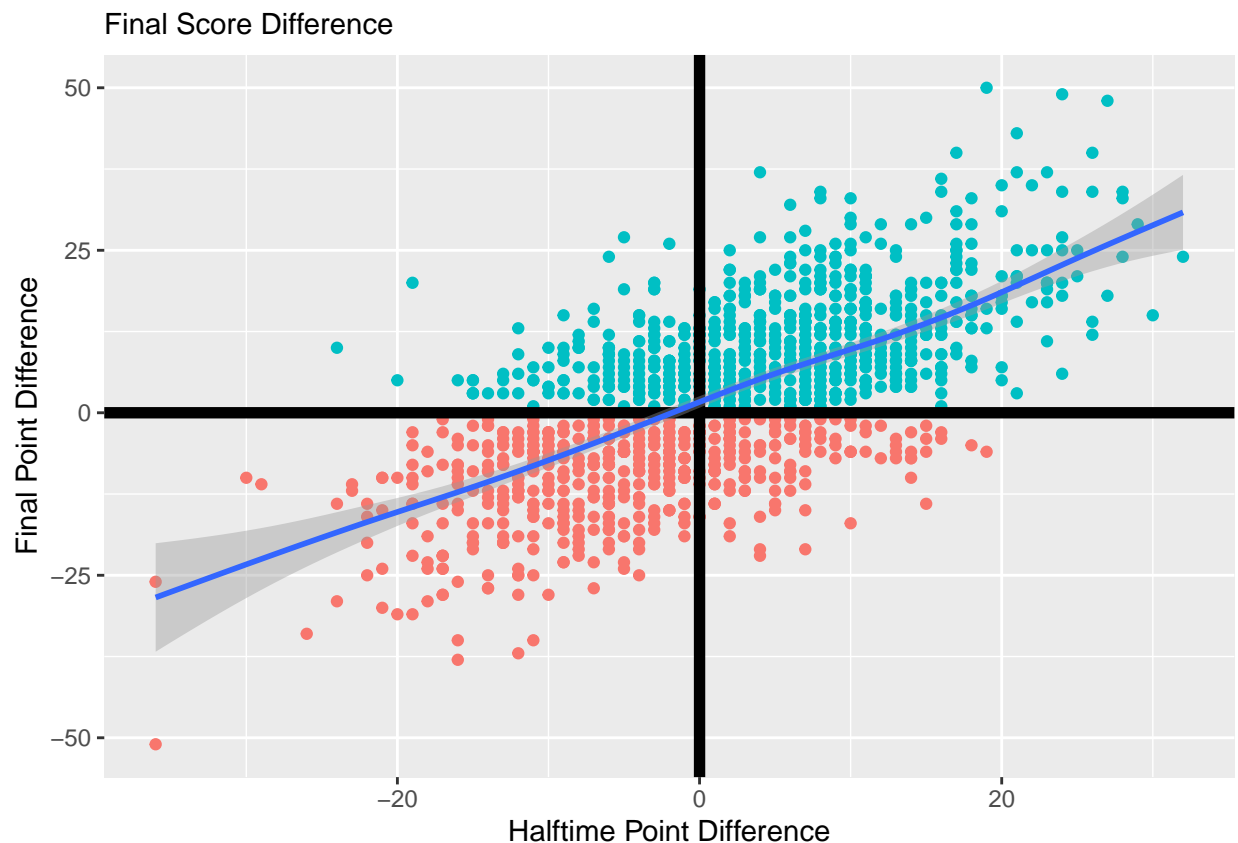
```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

If the team is winning in the first half then the percentage of that team wins the game increases.

```
ggplot(halftime2, aes( x = halftime_difference_home, y = final_score_home)) +  
  geom_point(aes(color = final_score_home > 0), show.legend = FALSE) + labs(x = "Halftime Point Difference",  
    subtitle = "Final Score Difference") + geom_hline(yintercept=0, color = "Black",
```

```
## 'geom_smooth()' using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Creating confusion Matrix, prediction for home and away team is obtained. From this results home team playing a Home game has more advantage and the count we received was 545.

```
halftime_lead <- factor(nba.data.clean$H_or_A_halftime)  
fulltime_win <- factor(nba.data.clean$H_or_A_win)
```

```

#Creating confusion matrix
cm <- confusionMatrix(data=fulltime_win, reference = halftime_lead)

#Display results
cm

```

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A    H
##           A 381 125
##           H 179 545
##
##           Accuracy : 0.7528
##           95% CI : (0.7277, 0.7767)
##           No Information Rate : 0.5447
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.4977
##
## Mcnemar's Test P-Value : 0.002368
##
##           Sensitivity : 0.6804
##           Specificity : 0.8134
##           Pos Pred Value : 0.7530
##           Neg Pred Value : 0.7528
##           Prevalence : 0.4553
##           Detection Rate : 0.3098
##           Detection Prevalence : 0.4114
##           Balanced Accuracy : 0.7469
##
##           'Positive' Class : A
##

```

```

draw_confusion_matrix <- function(cm) {
  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('Halftime lead turning into win', cex.main=2)
  # create the matrix
  rect(150, 430, 240, 370, col='#3F97D0')
  text(195, 435, 'Away_lead', cex=1.2)
  rect(250, 430, 340, 370, col='#F7AD50')
  text(295, 435, 'Home_lead', cex=1.2)
  text(125, 370, 'Full_time_winner', cex=1.3, srt=90, font=2)
  text(245, 450, 'Half_time_lead', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='#F7AD50')
  rect(250, 305, 340, 365, col='#3F97D0')
  text(140, 400, 'Away_win', cex=1.2, srt=90)
  text(140, 335, 'Home_win', cex=1.2, srt=90)
  #add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
}

```



```

text(195, 335, res[2], cex=1.6, font=2, col='white')
text(295, 400, res[3], cex=1.6, font=2, col='white')
text(295, 335, res[4], cex=1.6, font=2, col='white')
}

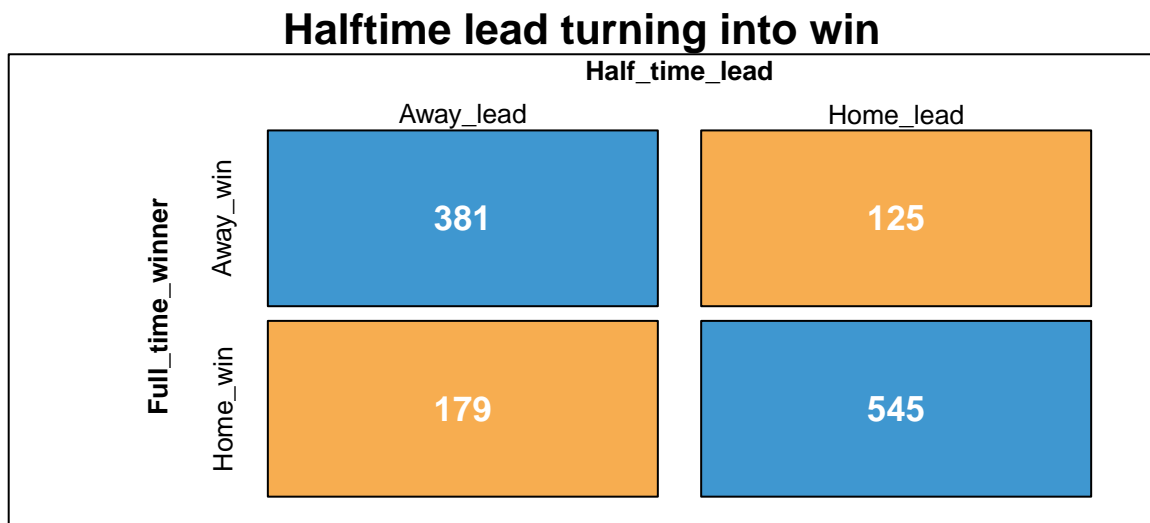
```

The above confusion matrix function was taken from stackoverflow. url - <https://stackoverflow.com/questions/23891140/r-how-to-visualize-confusion-matrix-using-the-caret-package>

```

# cm <- confusionMatrix(reference = nba.data.clean$fulltime_win, data = nba.data.clean$halftime_lead)
draw_confusion_matrix(cm)

```



Rebound analysis

We created a new column which is the difference between the total rebounds by home team and total rebounds by away team. We also analysed the point difference by getting difference from home points and away points.

```

halftime2 = halftime %>%
  mutate_if(is.character, as.integer)

```

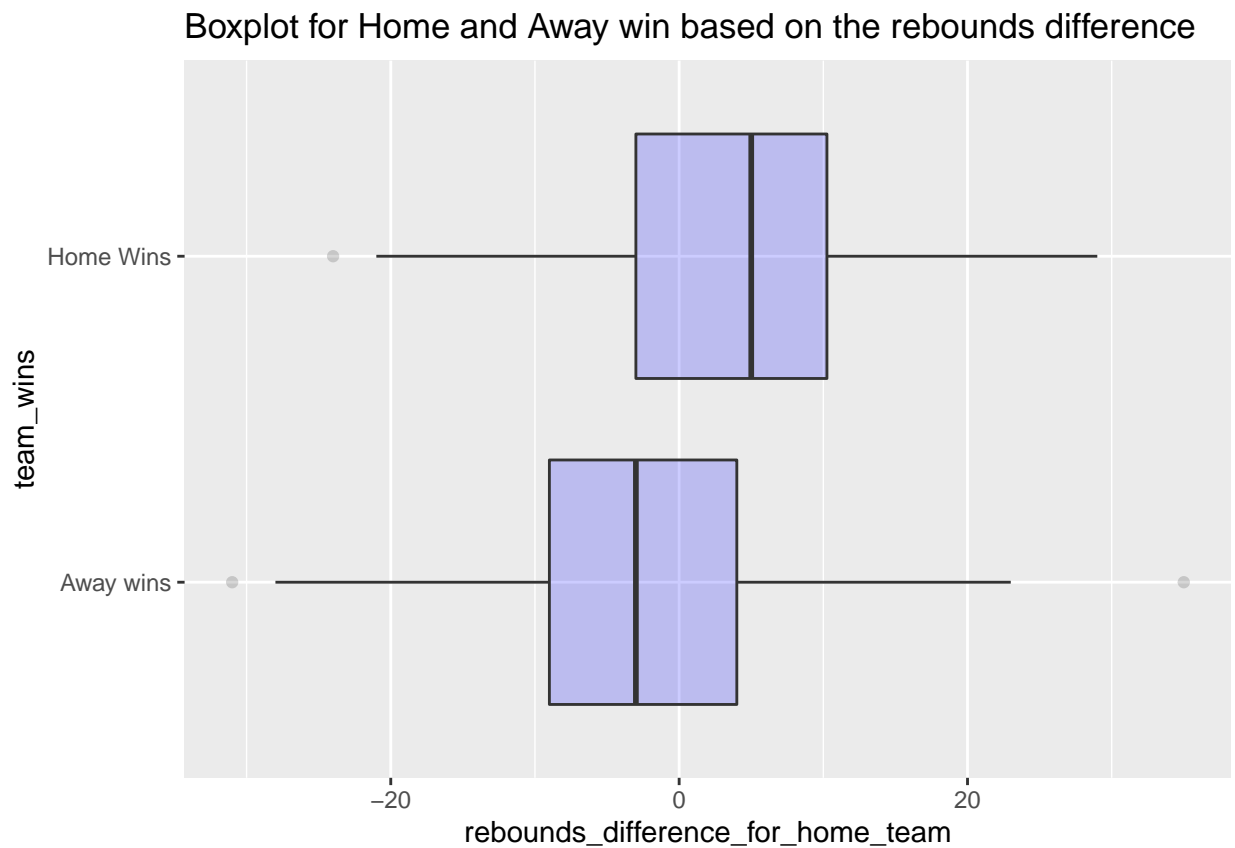
```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
halftime2<-mutate(halftime2,rebounds_difference_for_home_team=total_rebounds_home-total_rebounds_away)
halftime2<-mutate(halftime2, points_difference=home_points-away_points)
halftime2
```

```
## # A tibble: 1,230 x 59
##       id date      home_team home_points away_team away_points
##       <dbl> <date>      <int>      <dbl>      <int>      <dbl>
## 1 400827888 2015-10-27      NA         94      NA        106
## 2 400827889 2015-10-27      NA         97      NA         95
## 3 400827890 2015-10-27      NA        111      NA         95
## 4 400827891 2015-10-28      NA         87      NA         88
## 5 400827892 2015-10-28      NA        112      NA         95
## 6 400827893 2015-10-28      NA        100      NA        115
## 7 400827894 2015-10-28      NA         92      NA         87
## 8 400827895 2015-10-28      NA        104      NA         94
## 9 400827896 2015-10-28      NA        106      NA         99
## 10 400827897 2015-10-28      NA         85      NA        105
## # ... with 1,220 more rows, and 53 more variables: home_half_time_score <int>,
## #   away_half_time_score <int>, fg_made_home <int>, fg_attempts_home <int>,
## #   fg_made_away <int>, fg_attempts_away <int>, field_goal_%_home <dbl>,
## #   field_goal_%_away <dbl>, 3pt_made_home <int>, 3pt_attempts_home <int>,
## #   3pt_made_away <int>, 3pt_attempts_away <int>, three_point_%_home <dbl>,
## #   three_point_%_away <dbl>, ft_made_home <int>, ft_attempts_home <int>,
## #   ft_made_away <int>, ft_attempts_away <int>, free_throw_%_home <dbl>,
## #   free_throw_%_away <dbl>, total_rebounds_home <dbl>,
## #   total_rebounds_away <dbl>, offensive_rebounds_home <dbl>,
## #   offensive_rebounds_away <dbl>, defensive_rebounds_home <dbl>,
## #   defensive_rebounds_away <dbl>, team_rebounds_home <dbl>,
## #   team_rebounds_away <dbl>, assists_home <dbl>, assists_away <dbl>,
## #   steals_home <dbl>, steals_away <dbl>, blocks_home <dbl>, blocks_away <dbl>,
## #   turnovers_home <dbl>, turnovers_away <dbl>, points_home <dbl>,
## #   points_away <dbl>, fast_break_points_home <dbl>,
## #   fast_break_points_away <dbl>, points_in_paint_home <dbl>,
## #   points_in_paint_away <dbl>, personal_fouls_home <dbl>,
## #   personal_fouls_away <dbl>, technical_fouls_home <dbl>,
## #   technical_fouls_away <dbl>, flagrant_fouls_home <dbl>,
## #   flagrant_fouls_away <dbl>, H_or_A_win <int>, H_or_A_halftime <int>,
## #   winning_team <int>, rebounds_difference_for_home_team <dbl>,
## #   points_difference <dbl>
```

If the point difference is greater than 0 then the Home team wins. The median of the Boxplot when its towards the positive side of the graph then the chances of winning the game are high.

```
halftime2<-halftime2 %>%
  group_by(points_difference,rebounds_difference_for_home_team) %>%
  mutate(team_wins=ifelse(points_difference>0,'Home Wins','Away wins'))
ggplot(data=halftime2,aes(x=team_wins,y=rebounds_difference_for_home_team))+geom_boxplot(fill='blue',al
labs(title = 'Boxplot for Home and Away win based on the rebounds difference')+coord_flip()
```



Three pointer

```
three_pt <- lm(`three_point_%_home`) ~ (H_or_A_win), data = nba.data.clean)
summary(three_pt)$coef
```

```
##           Estimate Std. Error t value    Pr(>|t|)
## (Intercept) 31.629249  0.4218076  74.98502 0.000000e+00
## H_or_A_winH   6.901138  0.5497911  12.55229 4.329793e-34
```

Feature Analysis

Cleaning Data

```

df <- df %>%
  separate(`3pt_made_attempted_home`, into = c("3pt_attempts_home", "3pt_made_home")) %>%
  separate(`3pt_made_attempted_away`, into = c("3pt_attempts_away", "3pt_made_away"))
df$`3pt_made_home` <- as.numeric(df$`3pt_made_home`)
df$`3pt_made_away` <- as.numeric(df$`3pt_made_away`)
df = df %>%
  separate(ft_made_attempted_home, into = c("ft_attempts_home", "ft_made_home")) %>%
  separate(ft_made_attempted_away, into = c("ft_attempts_away", "ft_made_away"))
df <- df %>%
  mutate(Winning_freethrow_attempts = ifelse(ft_attempts_home > ft_attempts_away, ft_attempts_home, ft_attempts_away))
df <- df %>%
  mutate(Winning_Team = ifelse(home_points > away_points, home_team, away_team))
df <- df %>%
  mutate(Won = ifelse(home_points > away_points, 1, 0))
df <- df %>%
  mutate(Winning_points = ifelse(home_points > away_points, home_points, away_points))
ft_attempts_home <- as.double(df$ft_attempts_home)
ft_attempts_away <- as.double(df$ft_attempts_away)
#2
df <- df %>%
  mutate(Losing_freethrow_attempts = ifelse(ft_attempts_home < ft_attempts_away, ft_attempts_home, ft_attempts_away))
df <- df %>%
  mutate(Winning_points = ifelse(home_points > away_points, home_points, away_points))
df <- df %>%
  mutate(Losing_points = ifelse(home_points < away_points, home_points, away_points))
df <- df %>%
  mutate(Losing_Team = ifelse(home_points < away_points, home_team, away_team))
df <- df %>%
  mutate(Lost = ifelse(home_points < away_points, 0, 1))
df$Winning_freethrow_attempts <- as.double(df$Winning_freethrow_attempts)
df$Losing_freethrow_attempts <- as.double(df$Losing_freethrow_attempts)

```

Free Throw Attempts By Team

```
df[c('Winning_Team', 'Winning_freethrow_attempts', 'Losing_Team', 'Losing_freethrow_attempts')]
```

```

## # A tibble: 1,230 x 4
##   Winning_Team Winning_freethrow_attempts Losing_Team Losing_freethrow_attempts
##   <chr>                <dbl> <chr>                <dbl>
## 1 DET                    20 ATL                    12
## 2 CHI                    16 CLE                    10
## 3 GS                     20 NO                     19
## 4 WSH                     8 ORL                    15
## 5 BOS                    26 PHI                    20
## 6 CHI                    22 BKN                    17
## 7 DET                    25 UTAH                   15
## 8 MIA                    22 CHA                    20
## 9 TOR                    27 IND                    26
## 10 DEN                   17 HOU                    12
## # ... with 1,220 more rows

```

Visualization for free throws attempted ## The Graph represents win or loss by taking the mean of winning free throw attempts and losing free throw attempts.


```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

```
df2 <- df %>%
  group_by(Losing_Team) %>%
  filter(Losing_Team == c('GS', 'SA', 'CLE', 'TOR', 'OKC', 'MIA', 'LAC')) %>%
  summarise(Losing_Free = mean(Losing_freethrow_attempts)) %>%
  as.data.frame()
```

```
## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

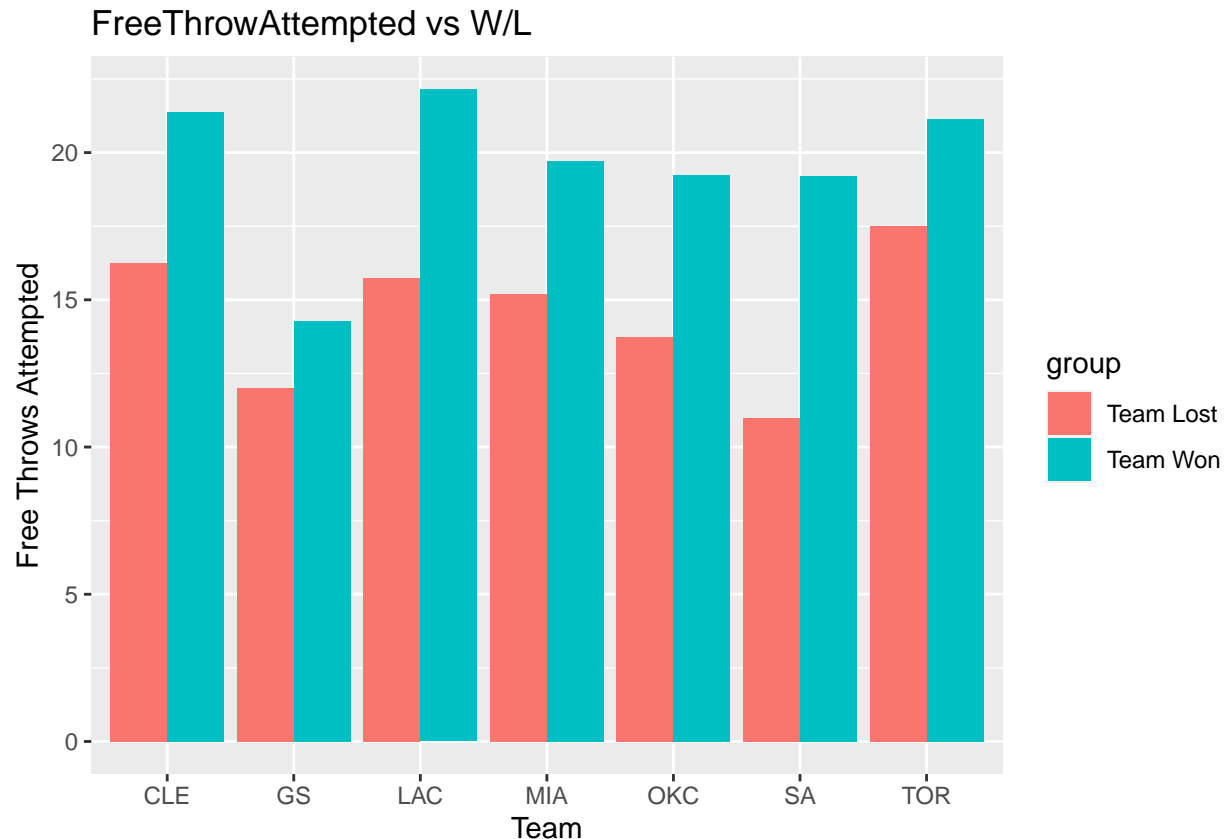
## Warning in Losing_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```



```

colnames(df2) = c('Winning_Team', 'Win_Free')
v3 <- rbind(df1, df2)
v3$group <- c(rep("Team Won", nrow(df1)), rep("Team Lost", nrow(df2)))
v3 %>%
  ggplot(aes(x=Winning_Team, y=Win_Free, fill = group)) +
  geom_bar(stat="identity", position = 'dodge') + labs(x = 'Team', y = 'Free Throws Attempted', title =

```



Visualization for home vs away points scored # The Graph Represents Win or Loss from the Home points and away points.

```

dp1 <- df %>%
  group_by(Winning_Team) %>%
  filter(Winning_Team == c('GS', 'SA', 'CLE', 'TOR', 'OKC', 'MIA', 'LAC')) %>%
  summarise(Home_Points = mean(home_points)) %>%
  as.data.frame()

```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```


[illegible]

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
dp2 <- df %>%  
  group_by(Winning_Team) %>%  
  filter(Winning_Team == c('GS', 'SA', 'CLE', 'TOR', 'OKC', 'MIA', 'LAC')) %>%  
  summarise(away_points = mean(away_points)) %>%  
  as.data.frame()
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

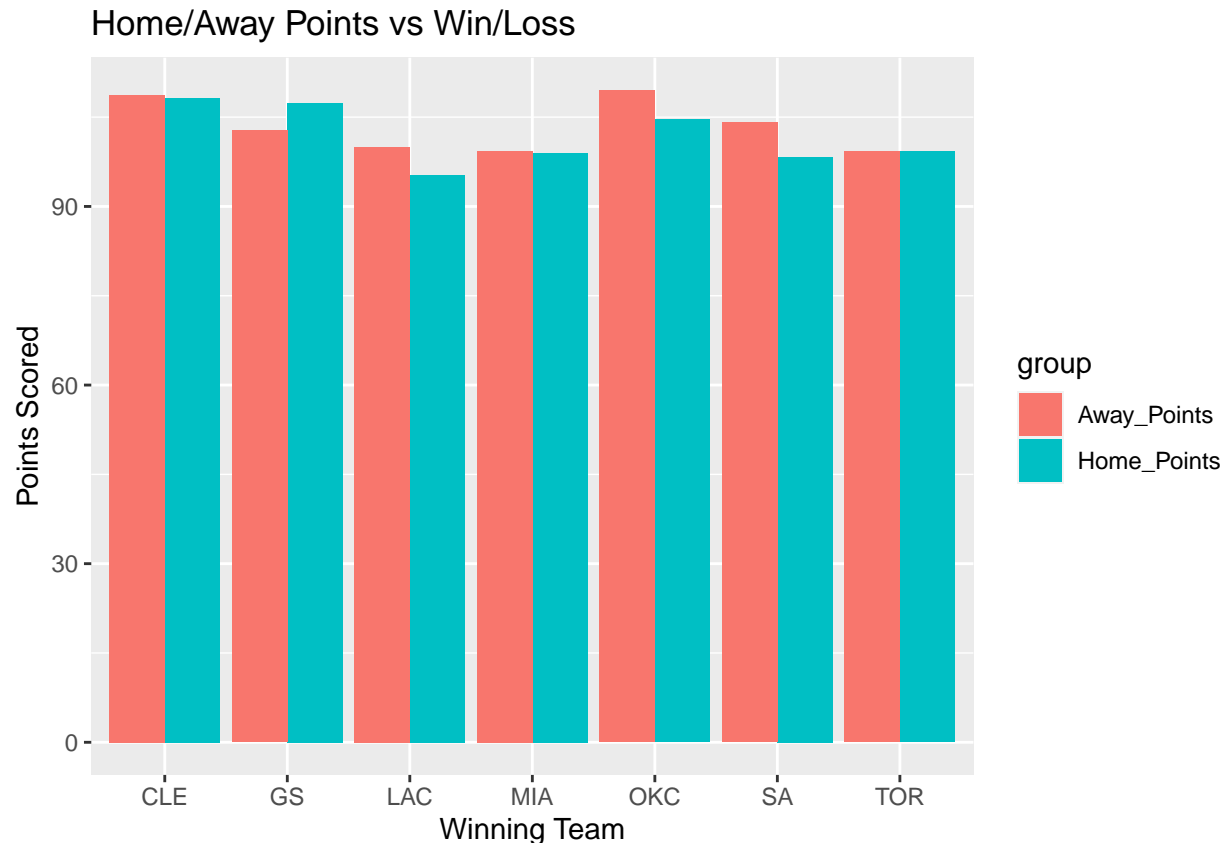
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

```
colnames(dp2) = c('Winning_Team', 'Home_Points')
v3 <- rbind(dp1, dp2)
v3$group <- c(rep("Home_Points", nrow(dp1)), rep("Away_Points", nrow(dp2)))
v3 %>%
  ggplot(aes(x=Winning_Team, y=Home_Points, fill = group)) +
  geom_bar(stat="identity", position = 'dodge') + labs(x = 'Winning Team', y = 'Points Scored', title=
```



Visualization for 3 pointers scored in home vs away

```
df <- df %>%
  mutate(Won = ifelse(home_points > away_points, 1, 0))

del <- df %>%
  group_by(Winning_Team) %>%
  filter(Winning_Team == c('GS', 'SA', 'CLE', 'TOR', 'OKC', 'MIA', 'LAC')) %>%
  summarise(ThreeP_Home = mean(`3pt_made_home`))
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

[illegible]

```
## longer object length is not a multiple of shorter object length
```

```
de2 <- df %>%  
  group_by(Winning_Team) %>%  
  filter(Winning_Team == c('GS', 'SA', 'CLE', 'TOR', 'OKC', 'MIA', 'LAC')) %>%  
  summarise(ThreeP_Away = mean(`3pt_made_away`))
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):  
## longer object length is not a multiple of shorter object length
```

```
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

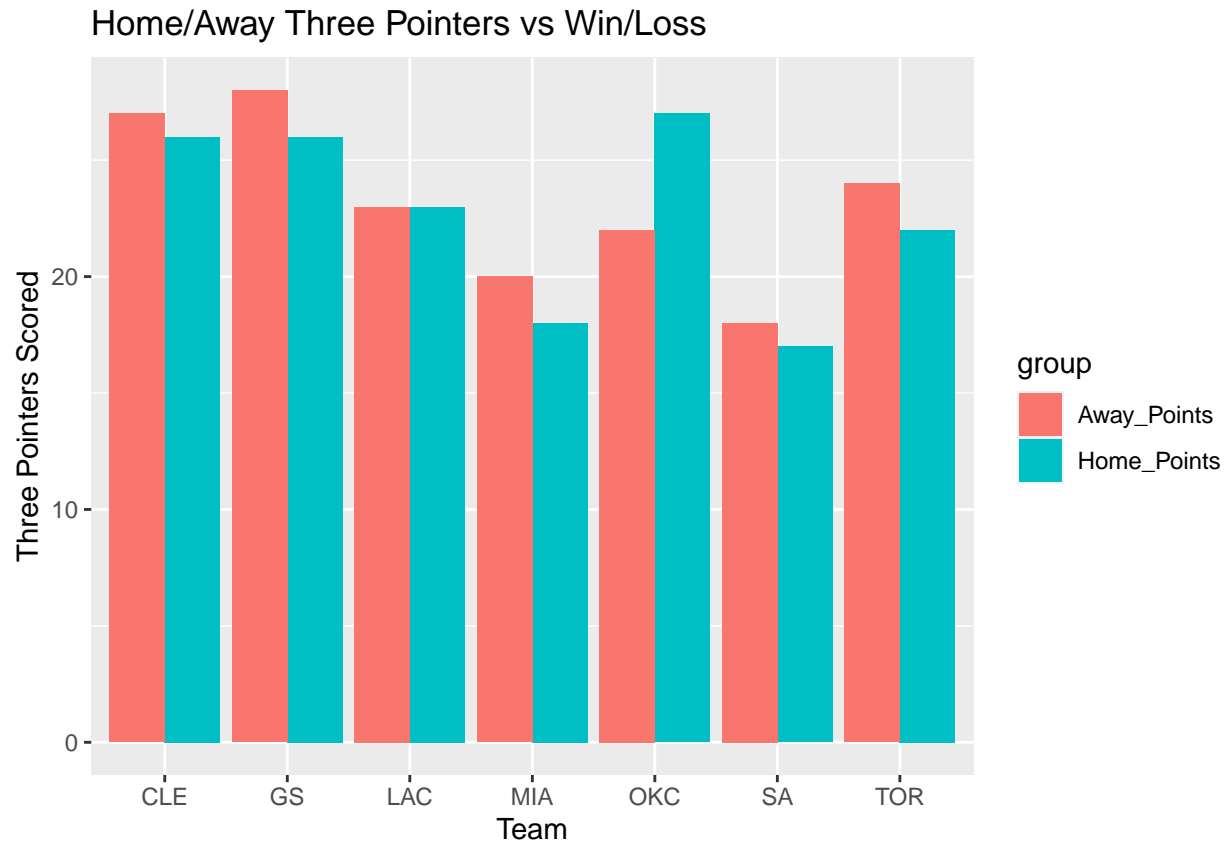
## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length

## Warning in Winning_Team == c("GS", "SA", "CLE", "TOR", "OKC", "MIA", "LAC"):
## longer object length is not a multiple of shorter object length
```

```
colnames(de2) = c('Winning_Team', 'ThreeP_Home')
v3 <- rbind(de1, de2)
v3$ThreeP_Home <- as.integer(v3$ThreeP_Home)
v3$group <- c(rep("Home_Points", nrow(dp1)), rep("Away_Points", nrow(dp2)))
v3 %>%
  ggplot(aes(x=Winning_Team, y=ThreeP_Home, fill = group)) +
  geom_bar(stat="identity", position = 'dodge') + labs(x = 'Team', y = 'Three Pointers Scored', title =
```



Logistic Regression Model ## Using Logistic Regression Model we received an Accuracy of 86% for the Home Team winning.

```
smp_size <- floor(0.75 * nrow(df))
set.seed(123)
train_ind <- sample(seq_len(nrow(df)), size = smp_size)
train <- df[train_ind, ]
test <- df[-train_ind, ]
#train$fast_break_points_home
#test$points_in_paint_home
train$Won <- factor(train$Won)
model <- glm(Won ~ home_points + `3pt_made_home` + fast_break_points_home + offensive_rebounds_home + V
predictions <- model %>% predict(test)
predicted.classes <- ifelse(predictions > 0.5, 1, 0)

test$Predicted_Win_Loss <- predicted.classes
mean(test$Predicted_Win_Loss == test$Won) * 100
```

[1] 84.09091

Conclusion:-

The important parameters we considered for the Analysis are :- In Home Game, Home team has higher chances of winning the game. In Half time Analysis the winning team in high time has more percentage of winning the game. In rebound If the boxplot has median towards the positive side the winning chances are more for the team. The Accuracy of Logistic Regression of Home Team winning is 84%.

Bias:-

We have done our analysis for only 2015-16 season. For a good analysis, we should consider the trend of these variables and also for the latest seasons to predict more accurately.

Also we have considered three point percentage made by the home team and predicted that if they convert 28% of their 3 pointers, they might win the game. It also depends on how many three pointer attempts were made by the team. If the team attempts only 3 three pointers and convert one of them, would this lead to a win? These questions can be answered through further analysis.