# xG(Expected Goals) Model

**Name:** Siddhant Thakur
**Reg No.:** RA1811027010004

## Project Description

To devise a model which can help quantify the shot quality and analyse the team's attacking efficiency.

## About the Dataset

Even though the event/tracking dataset for every football match is mostly commercialised and not openly available, Wyscout and a few other professional football platforms have made parts of this data available to the upcoming sports analytics newbies.
The dataset was in the form of a json file consisting of all the event data from the English Premier League for the seasons 2008-2018. Tags in the event data tells us about the type of event, whether it was a pass or a shot or a header.
The above dataset was processed using a script to convert it into usable data consisting of coordinates of the shot and the normalised data of y coordinate.

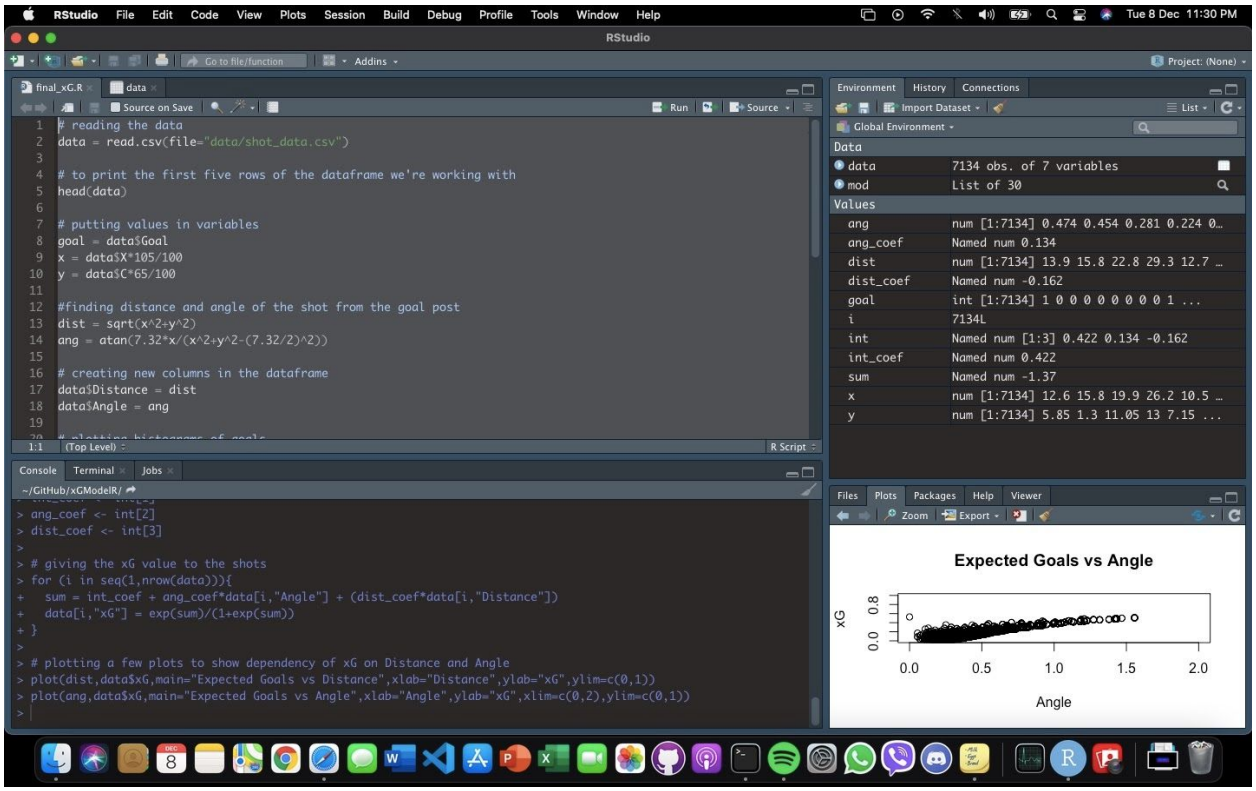
## Code

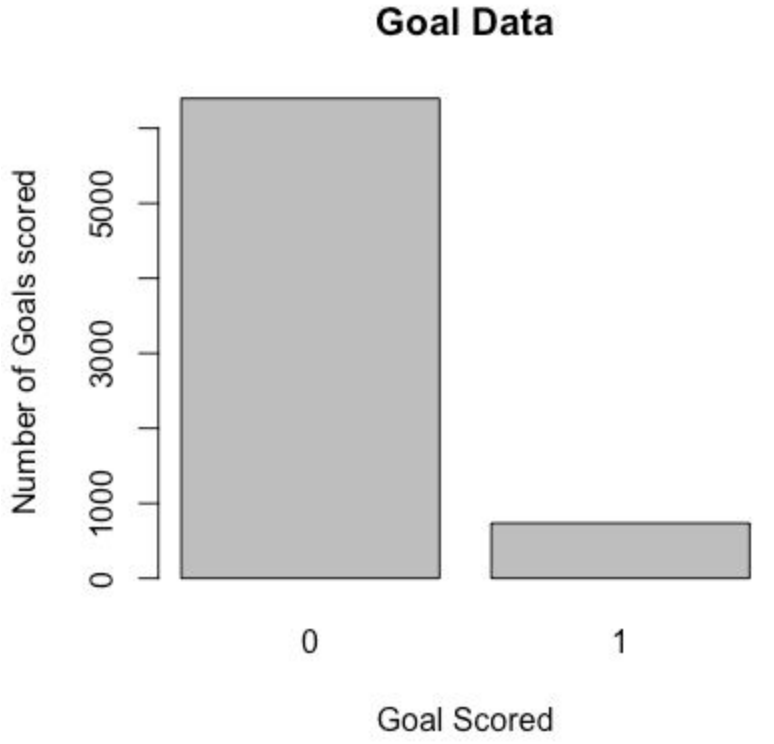
```
# reading the data
data = read.csv(file="data/shot_data.csv")

# to print the first five rows of the dataframe we're working with
head(data)

# putting values in variables
goal = data$Goal
x = data$X*105/100
y = data$C*65/100
```

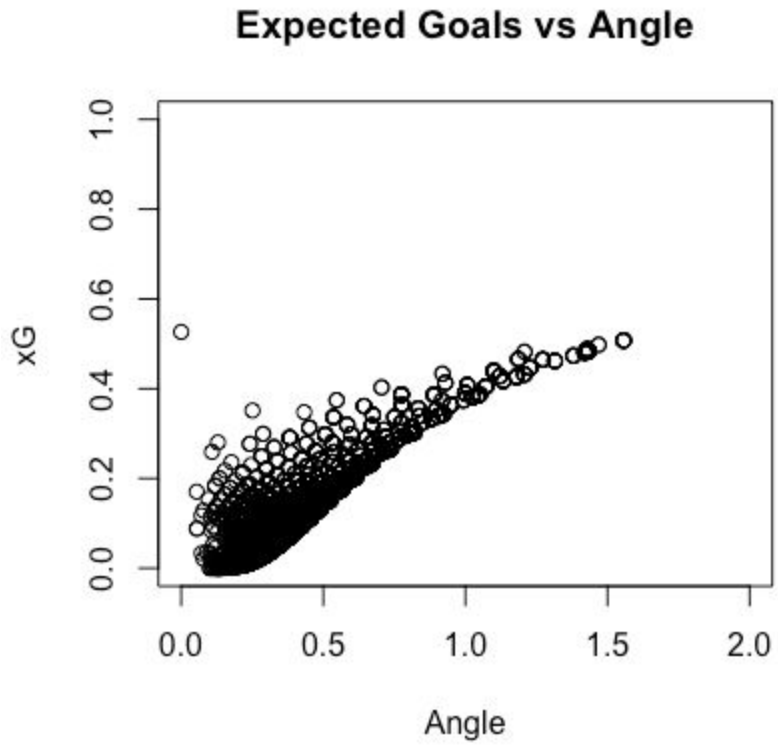
```
#finding distance and angle of the shot from the goal post
dist = sqrt(x^2+y^2)
ang = atan(7.32*x/(x^2+y^2-(7.32/2)^2))

# creating new columns in the dataframe
data$Distance = dist
data$Angle = ang

# plotting histograms of goals
barplot(table(goal), xlab="Goal Scored", ylab="Number of Goals
scored",main="Goal Data")

# modelling the logistic regression with the
# independent features as Angle and Distance
# dependent features as Goal
mod <- glm(Goal ~ Angle + Distance, data=data, family=binomial)
summary(mod)

# getting the intercepts of the model
int <- coef(mod)
int_coef <- int[1]
ang_coef <- int[2]
dist_coef <- int[3]

# giving the xG value to the shots
for (i in seq(1,nrow(data))){
  sum = int_coef + ang_coef*data[i,"Angle"] + (dist_coef*data[i,"Distance"])
  data[i,"xG"] = exp(sum)/(1+exp(sum))
}

# plotting a few plots to show dependency of xG on Distance and Angle
plot(dist,data$xG,main="Expected Goals vs
Distance",xlab="Distance",ylab="xG",ylim=c(0,1))
```

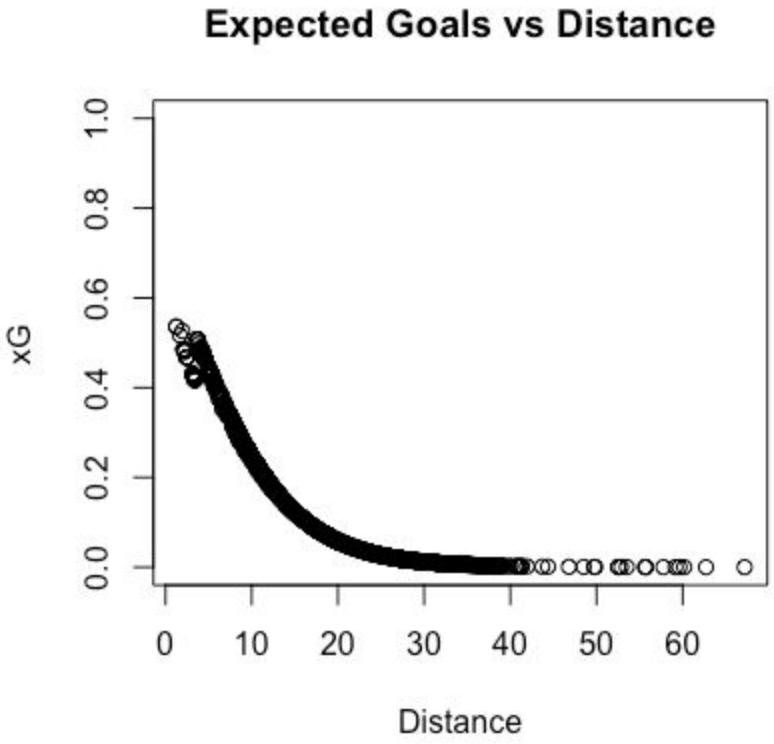
plot(ang,data$xG,main="Expected Goals vs Angle",xlab="Angle",ylab="xG",xlim=c(0,2),ylim=c(0,1))

## Snapshot




Goal Data
Number of Goals scored
0
1000
3000
5000
0
1
Goal Scored


Expected Goals vs Angle
xG
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Angle




**Result**

A successful xG model was implemented by using R.