How data can affect footbal

Giuseppe Sabino

2022-11-15

introduzione

Football is one of the most popular themes in Italy and in the world. Few people don't have a favorite team or at least don't sympathize for anyone. In recent years various technologies have been introduced in this sport,for example VAR and goal line technology. Furthermore, the use of data for the analysis of individual or team performances is increasingly developing, just think of the De Bruyne case, thanks to a team of data scientists managed to obtain an increase in his contract renewal. with this experiment I will try to give an input of what the data can show us about this sport. It will be used a dataset of the last 5 seasons in Serie A, it contains the dates of the matches, the results and many other in-game statistics.

Import library and list of functions:

```
library(ggplot2)
library(ggridges)
library(hrbrthemes)

#function that calculate the sum of point a squad have in the last 5 years
point5years <- function(df){

df$match[1]<-1
if (df$result[1]== "W"){
  df$point[1] <-3
}
if(df$result[1]=="D"){
  df$point[1] <-1
}
if (df$result[1]== "L"){
  df$point[1] <-0
}


for(i in 2:nrow(df)){
  if (df$result[i]== "W"){
    df$point[i] <- df$point[i-1] +3
  }
  else if(df$result[i]=="D"){
    df$point[i] <- df$point[i-1] +1
  }
  else if(df$result[i]=="L"){
    df$point[i]<- df$point[i-1]
```

```
  }
  df$match[i]<-i
}
  return(df)
}

################################################
#function to plot drbbling and xg
dribbling_plot <- function(df){
  ggplot(df, aes(x=att, y=sh, color=result)) +
    geom_point() + # Show dots
    geom_text(
      label=df_milan$opponent,
      nudge_x = 0.25, nudge_y = 0.25,
      check_overlap = T
    )
}
################################################
#function to plot possess
possess<- function(df){
  ggplot(df, aes(x=xg, y=poss_x, color=result)) +
    geom_point() + # Show dots
    geom_text(
      label=df_milan$opponent,
      nudge_x = 0.25, nudge_y = 0.25,
      check_overlap = T
    )
}

################################################
#Create df for barplot
df_difference <- function(df){
  xg <- sum(df$xg)
  gf <- sum(df$gf)
  df_goal<-array(data= c(xg, gf))
  return(df_goal)

}
##########################################################
#Create df for barplot
df_difference_ag <- function(df){
  ga <- sum(df$ga)
  xga <-sum(df$xga)
  df_goal_ag<-array(data= c(ga, xga))
  return (df_goal_ag)
}

##########################################################
#merge goal scored for all team
merge_goal <- function(){
  df<-
```

```
array(data=c(milan_goal,inter_goal,juve_goal,napoli_goal,roma_goal,lazio_goal))
  return(df)
}
#########################################################
#merge goal concedeed for all team
merge_goal_ag <- function(){
  df<-
array(data=c(milan_goal_ag,inter_goal_ag,juve_goal_ag,napoli_goal_ag,roma_goal_ag,
lazio_goal_ag))
  return(df)
}
#########################################################
#linear regression
lr <- function(df){
  df<-df[c(15,16)]
  lmTemp <- lm(point~match, data = df)
  lmTemp
  plot(df)+
    abline(lmTemp)
  summary(lmTemp)
  df_lr=data.frame(match= c(191:228))
  result<- predict(lmTemp,df_lr)
  total_point<- round(result[38]-result[1])
  return(total_point[1])
}
```

It gives a first view of the dataset on which I am going to work.

```
df_football<- read.csv("seriea-matches.csv")
df_total<- df_football[c(2,5,50,11,7,8,9,10,12,13,14,19,34,35)]
head(df_total)
```

```
##         date       round  team  opponent venue result gf ga  xg xga poss_x sh
## 1 2021-08-23 Matchweek 1 Milan Sampdoria  Away      W  1  0 1.1 1.0     51 11
## 2 2021-08-29 Matchweek 2 Milan  Cagliari  Home      W  4  1 2.7 0.4     57 17
## 3 2021-09-12 Matchweek 3 Milan     Lazio  Home      W  2  0 3.0 0.3     47 20
## 4 2021-09-19 Matchweek 4 Milan  Juventus  Away      D  1  1 0.7 1.0     56 13
## 5 2021-09-22 Matchweek 5 Milan   Venezia  Home      W  2  0 1.7 0.7     67 14
## 6 2021-09-25 Matchweek 6 Milan    Spezia  Away      W  2  1 1.9 0.9     64 18
##   att succ.
## 1  17  35.3
## 2  20  65.0
## 3  26  65.4
## 4  17  52.9
## 5  31  58.1
## 6  25  44.0
```

There are 14 columns :

-date: day the match was played

-round:championship round

-team: team to which the statistics refer

-opponent: opponenent of the team squad

-venue:where the game was played {Home,Away}

-result: match result{W,D,L}

-gf: Number of goals scored

-ga: Number of goals conceded

-xgf: Expected number of goals scored

-xga: Expected number of goals conceded

-poss_x: Percentage of ball possession

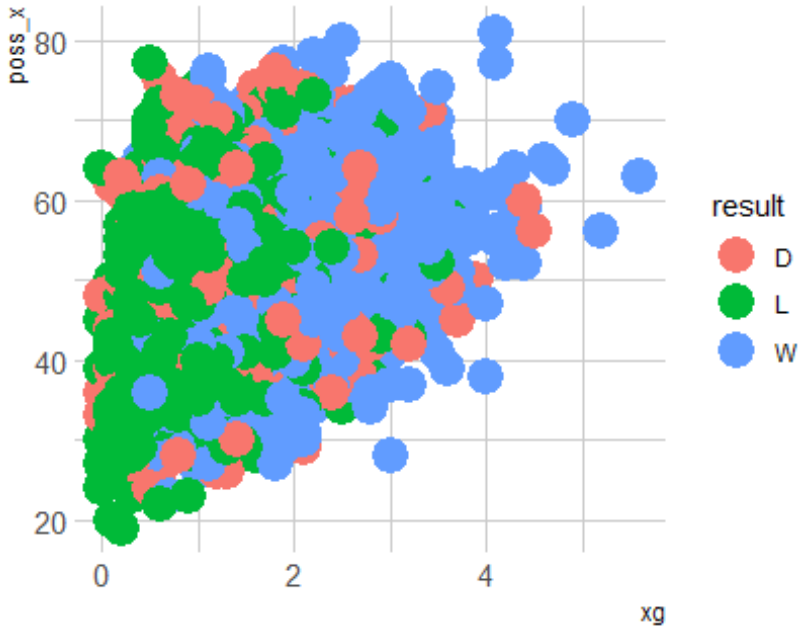
-sh: Number of shoot

-att: Number of attempted dribbling

-succ: Percentage of succesfull dribbling.

A first graph is displayed:

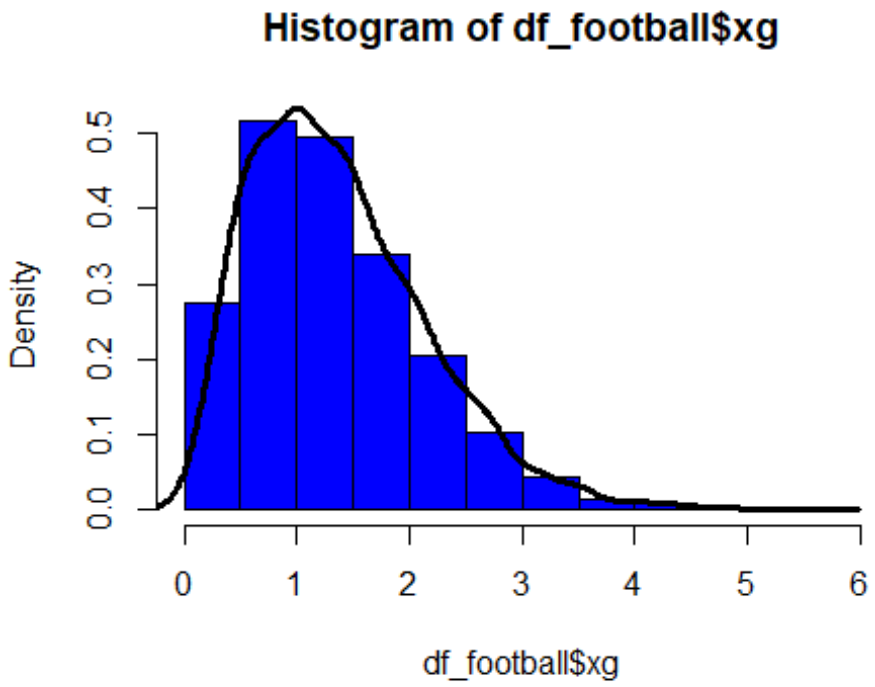
```
ggplot(df_total, aes(x=xg, y=poss_x, color=result)) +
  geom_point(size=6) +
  theme_ipsum()
```



This graph represent the relation between 3 variable : Result,xg,poss_x. it is easy to note a positive correlation between this 3 variable, because if a team create a lot of expected goal and generate more possession, it have more possibility to win the game. On the contrary, if a team creates a few xg and has a low percentage of ball possession, the difficulty of winning the game increases.

But what exactly are xg's? Expected Goals (xG) represent the offensive production of a team or player. XGs are a measure of how many goals a team or player should have scored, regardless of the result. Consquently, more xg -> more goal chances created -> more fun for the spectactors.

```
p1<-hist(df_football$xg,col="blue",freq=F)
kd1<- density(df_football$xg)
lines(kd1,col="black",lwd=3)
```



On average 1-2 xg are created per game.A very low number if you think that a game lasts 90 minutes. This is the reason why numerous proposals have been made to raise the level of competitions and create greater involvement of people(Superleague).

What will be done now will be to show the 6 most important teams in Italy. For simplicity, AC Milan, winner of the last Scudetto, will be analyzed, but for the sake of completeness all the teams will be shown at the end.

```
df_milan<-df_total
milan <- which(df_milan$team == "Milan")
df_milan<-df_milan[milan,]
summary(df_milan)
```

```
##      date              round              team              opponent        
##  Length:190         Length:190         Length:190         Length:190        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##     venue              result                gf              ga        
##  Length:190         Length:190         Min.   :0.000   Min.   :0.000  
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.000  
##  Mode  :character   Mode  :character   Median :2.000   Median :1.000  
##                                        Mean   :1.668   Mean   :1.032  
##                                        3rd Qu.:2.000   3rd Qu.:2.000  
##                                        Max.   :7.000   Max.   :5.000  
##        xg              xga            poss_x            sh       
##  Min.   :0.300   Min.   :0.100   Min.   :30.00   Min.   : 4.00  
##  1st Qu.:1.000   1st Qu.:0.600   1st Qu.:49.00   1st Qu.:12.00  
##  Median :1.500   Median :1.000   Median :54.00   Median :15.00  
##  Mean   :1.560   Mean   :1.087   Mean   :54.15   Mean   :15.55  
##  3rd Qu.:2.075   3rd Qu.:1.400   3rd Qu.:59.00   3rd Qu.:18.00  
##  Max.   :3.900   Max.   :3.200   Max.   :78.00   Max.   :39.00  
##       att             succ.       
##  Min.   : 4.00   Min.   :  0.00  
##  1st Qu.:12.00   1st Qu.: 54.50  
##  Median :17.00   Median : 64.30  
##  Mean   :17.34   Mean   : 63.27  
##  3rd Qu.:22.00   3rd Qu.: 72.70  
##  Max.   :35.00   Max.   :100.00
```

```
var(df_milan$gf)
```

```
## [1] 1.524394
```

```
cor(df_milan$gf,df_milan$xg)
```

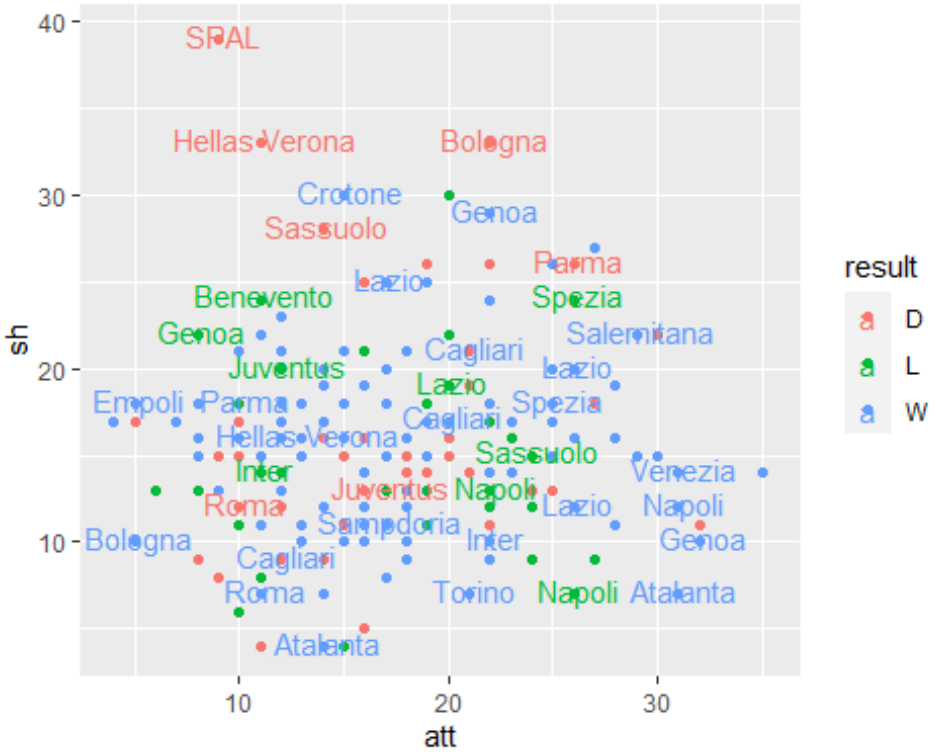
```
## [1] 0.5659132
```

```
cor(df_milan$ga,df_milan$xga)
```

```
## [1] 0.5452363
```

On average, Milan scores 1,668 goals per game, creating just fewer chances (1.56). He has an average possession percentage of 54.15% and attempts to dribble 17.34 per game, 63.27 of which are successful. The following graph shows the number of dribbles and the number of shots attempted

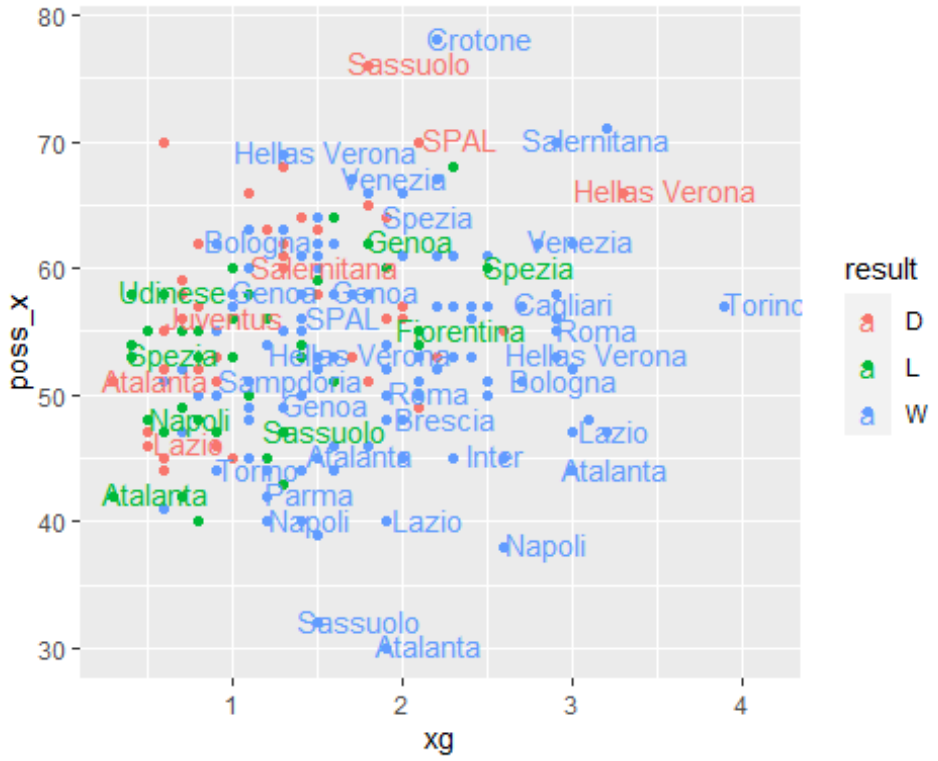
```
dribbling_plot(df_milan)
```



There is no clear and visible relationship, but the games in which Milan have shot more, have dribbled less.Conversely, when he tried more dribbles, the number of shots dropped. It can be said that the first case is due to very closed games, against weaker teams on paper, which have focused on defending the result.

As previously done, ball possession and xg are correlated

```
possess(df_milan)
```



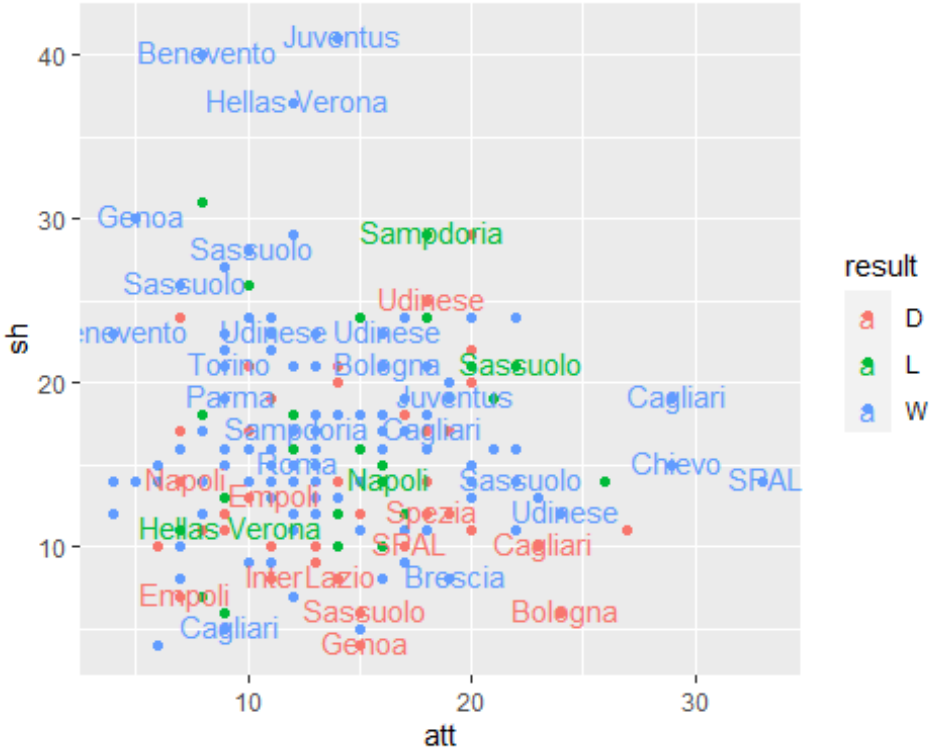
It is immediately evident that with the growth of the xg, the possibility of victory increases, in fact when Milan created 2 or more scoring chances, they lost only 4 times. It is also clear that to win you have to know how to suffer, in fact, Milan never lost when their possession was less than 40%, and usually it happens against big team. For all teams the previous graphs will be shown:

```
df_inter<-df_total
inter <- which(df_inter$team == "Internazionale")
df_inter<-df_inter[inter,]

summary(df_inter)
```

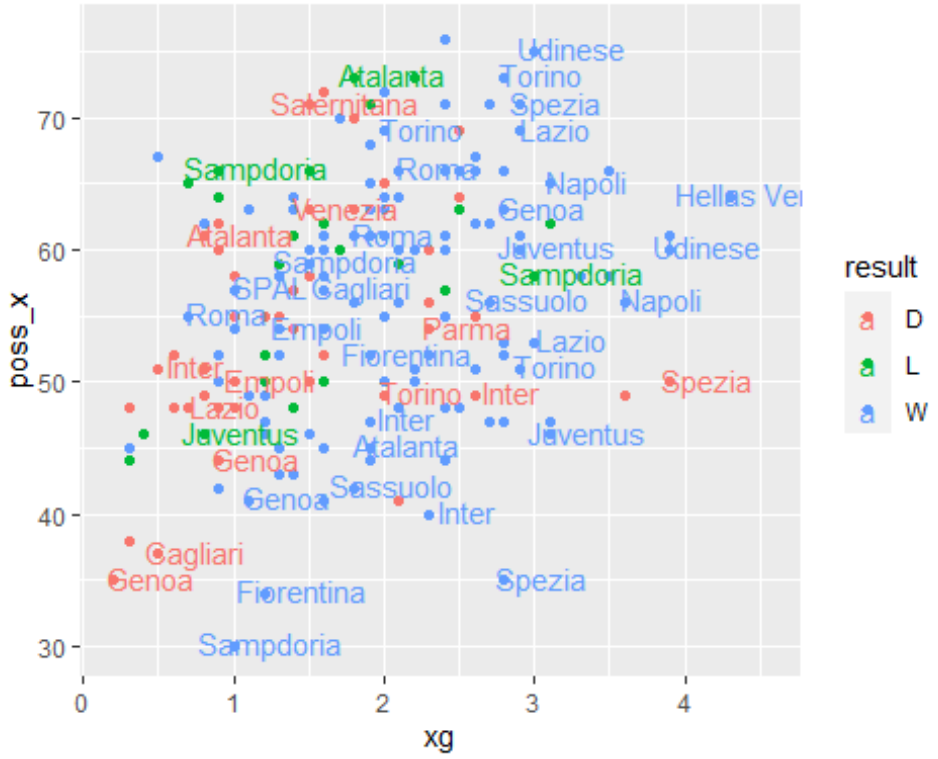
```
##      date               round              team              opponent
##  Length:190         Length:190         Length:190         Length:190
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     venue              result                gf              ga
##  Length:190         Length:190         Min.   :0.000   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :2.000   Median :1.0000
##                                        Mean   :1.984   Mean   :0.8737
##                                        3rd Qu.:3.000   3rd Qu.:1.0000
##                                        Max.   :6.000   Max.   :4.0000
##        xg              xga            poss_x            sh
##  Min.   :0.200   Min.   :0.000   Min.   :30.00   Min.   : 4.00
##  1st Qu.:1.225   1st Qu.:0.600   1st Qu.:49.00   1st Qu.:12.00
```

```
##  Median :1.800   Median :0.900   Median :56.00   Median :15.00
##  Mean   :1.861   Mean   :1.009   Mean   :55.85   Mean   :15.95
##  3rd Qu.:2.475   3rd Qu.:1.375   3rd Qu.:62.75   3rd Qu.:19.00
##  Max.   :4.300   Max.   :2.800   Max.   :76.00   Max.   :41.00
##       att              succ.
##  Min.   : 4.00   Min.   : 12.50
##  1st Qu.:10.00   1st Qu.: 50.00
##  Median :13.00   Median : 60.00
##  Mean   :13.79   Mean   : 60.09
##  3rd Qu.:17.00   3rd Qu.: 70.45
##  Max.   :33.00   Max.   :100.00
```

```
dribbling_plot(df_inter)
```




```
possess(df_inter)
```

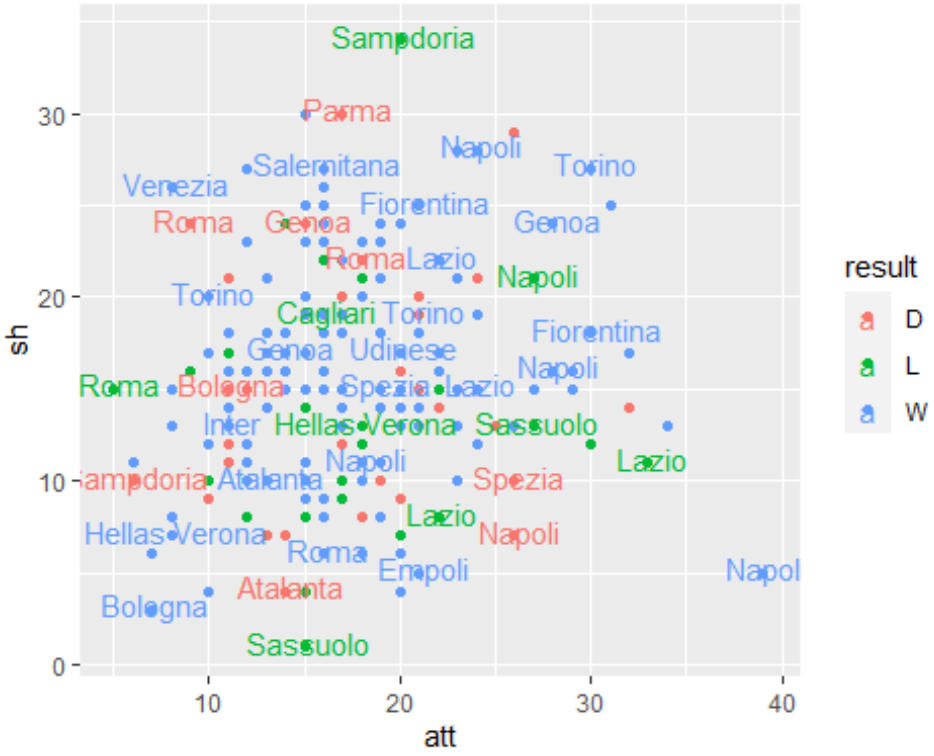



```
df_juve<-df_total
juve <- which(df_juve$team == "Juventus")
df_juve<-df_juve[juve,]

summary(df_juve)
```

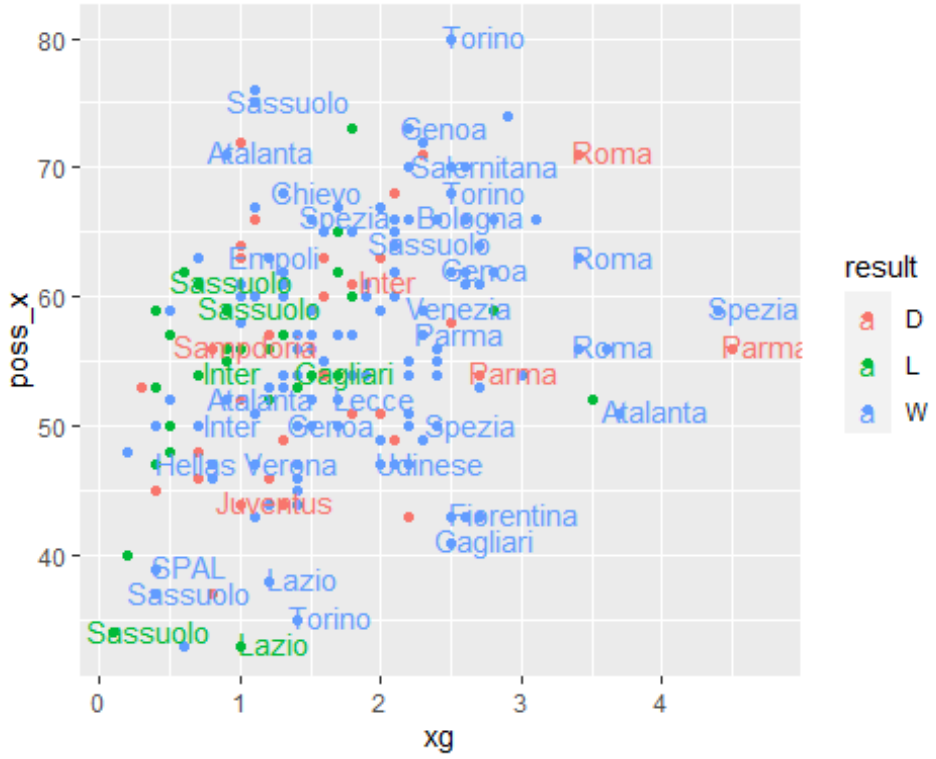
```
##      date               round              team              opponent
##  Length:190         Length:190         Length:190         Length:190
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     venue              result                gf              ga
##  Length:190         Length:190         Min.   :0.000   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :2.000   Median :1.0000
##                                        Mean   :1.926   Mean   :0.9053
##                                        3rd Qu.:3.000   3rd Qu.:1.0000
##                                        Max.   :7.000   Max.   :4.0000
##        xg              xga             poss_x            sh
##  Min.   :0.100   Min.   :0.0000   Min.   :33.00   Min.   : 1.00
##  1st Qu.:1.100   1st Qu.:0.5000   1st Qu.:50.00   1st Qu.:11.00
##  Median :1.500   Median :0.8000   Median :56.00   Median :15.00
##  Mean   :1.642   Mean   :0.9632   Mean   :55.89   Mean   :15.47
##  3rd Qu.:2.200   3rd Qu.:1.3000   3rd Qu.:62.00   3rd Qu.:19.00
##  Max.   :4.500   Max.   :2.8000   Max.   :80.00   Max.   :34.00
##       att             succ.
```

```
##  Min.   : 5.00   Min.   : 14.30
##  1st Qu.:14.00   1st Qu.: 52.25
##  Median :17.00   Median : 61.30
##  Mean   :17.51   Mean   : 61.46
##  3rd Qu.:21.00   3rd Qu.: 71.25
##  Max.   :39.00   Max.   :100.00
```

```
dribbling_plot(df_juve)
```



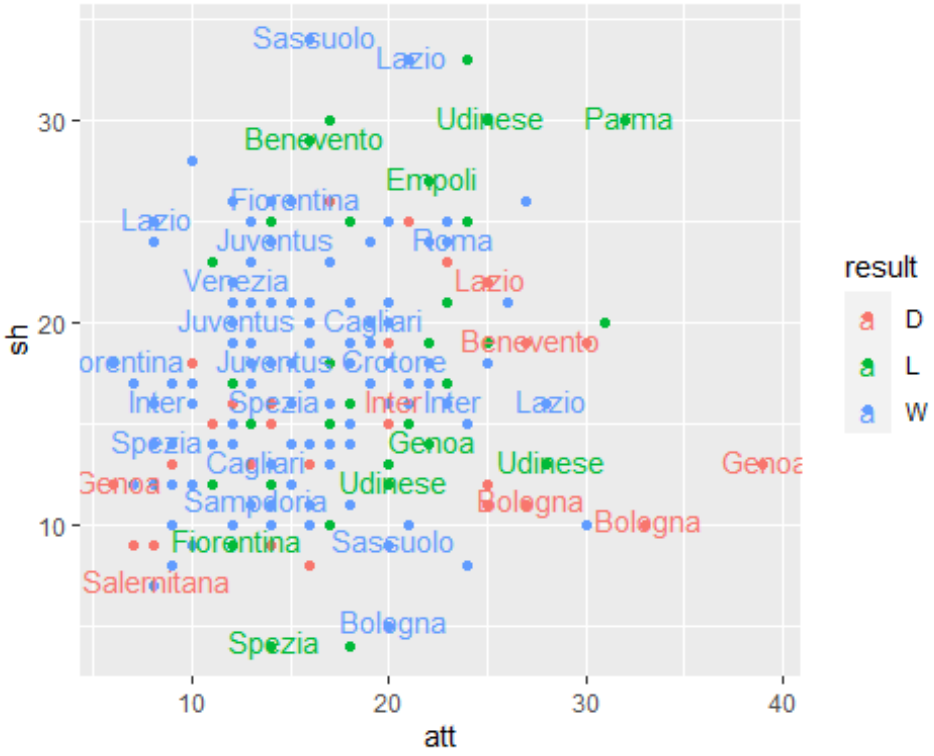

```
possess(df_juve)
```



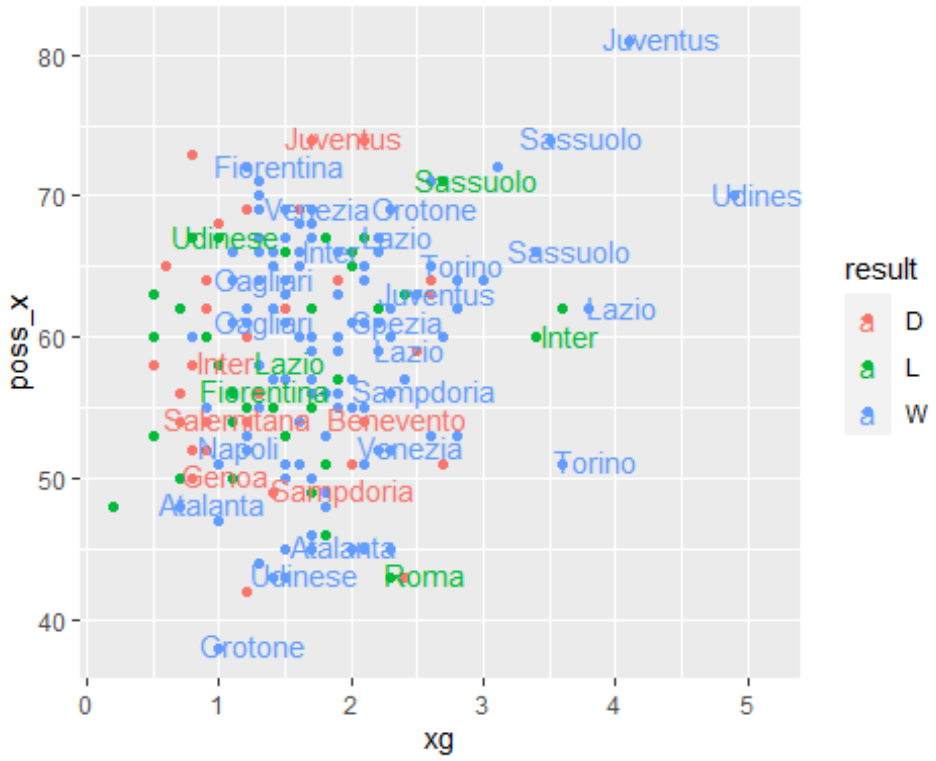

```
df_napoli<-df_total
napoli <- which(df_napoli$team == "Napoli")
df_napoli<-df_napoli[napoli,]

summary(df_napoli)
```

```
##      date               round              team              opponent
##  Length:190         Length:190         Length:190         Length:190
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     venue              result                gf              ga
##  Length:190         Length:190         Min.   :0.000   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :2.000   Median :1.0000
##                                        Mean   :1.958   Mean   :0.9842
##                                        3rd Qu.:3.000   3rd Qu.:1.0000
##                                        Max.   :6.000   Max.   :4.0000
##        xg              xga             poss_x            sh
##  Min.   :0.200   Min.   :0.1000   Min.   :38.00   Min.   : 4.00
##  1st Qu.:1.200   1st Qu.:0.5000   1st Qu.:53.00   1st Qu.:13.00
##  Median :1.700   Median :0.9000   Median :60.00   Median :17.00
##  Mean   :1.707   Mean   :0.9089   Mean   :58.98   Mean   :17.06
##  3rd Qu.:2.100   3rd Qu.:1.2000   3rd Qu.:65.00   3rd Qu.:20.00
##  Max.   :4.900   Max.   :2.8000   Max.   :81.00   Max.   :34.00
##       att             succ.
```

```
##  Min.   : 6.00   Min.   : 23.50
##  1st Qu.:13.00   1st Qu.: 50.00
##  Median :16.00   Median : 60.00
##  Mean   :16.68   Mean   : 59.84
##  3rd Qu.:20.00   3rd Qu.: 70.00
##  Max.   :39.00   Max.   :100.00

dribbling_plot(df_napoli)
```



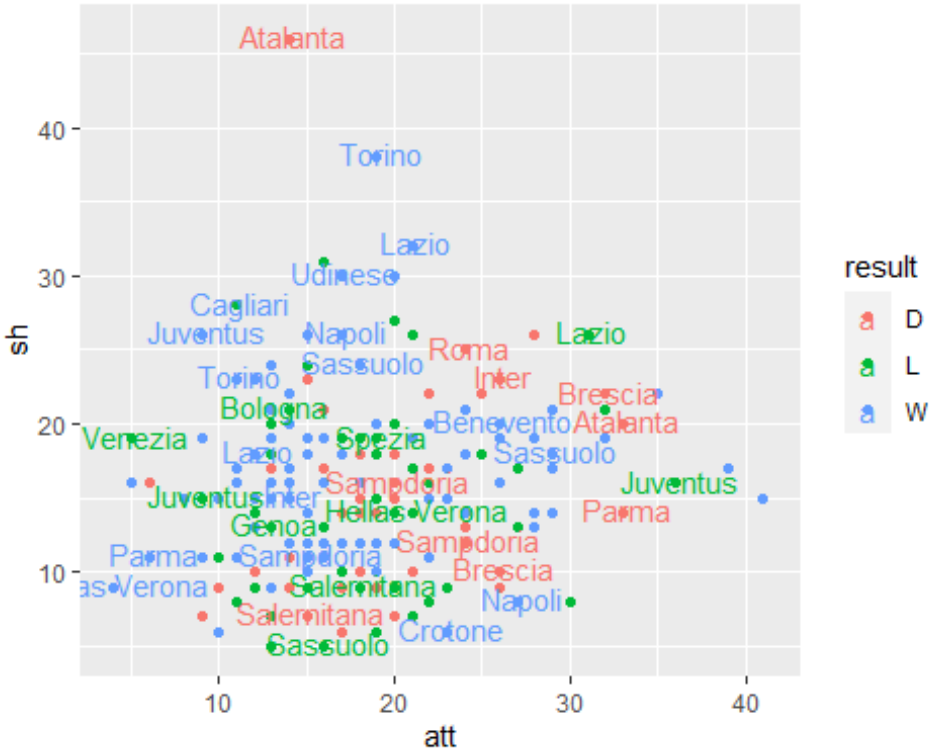

```
possess(df_napoli)
```



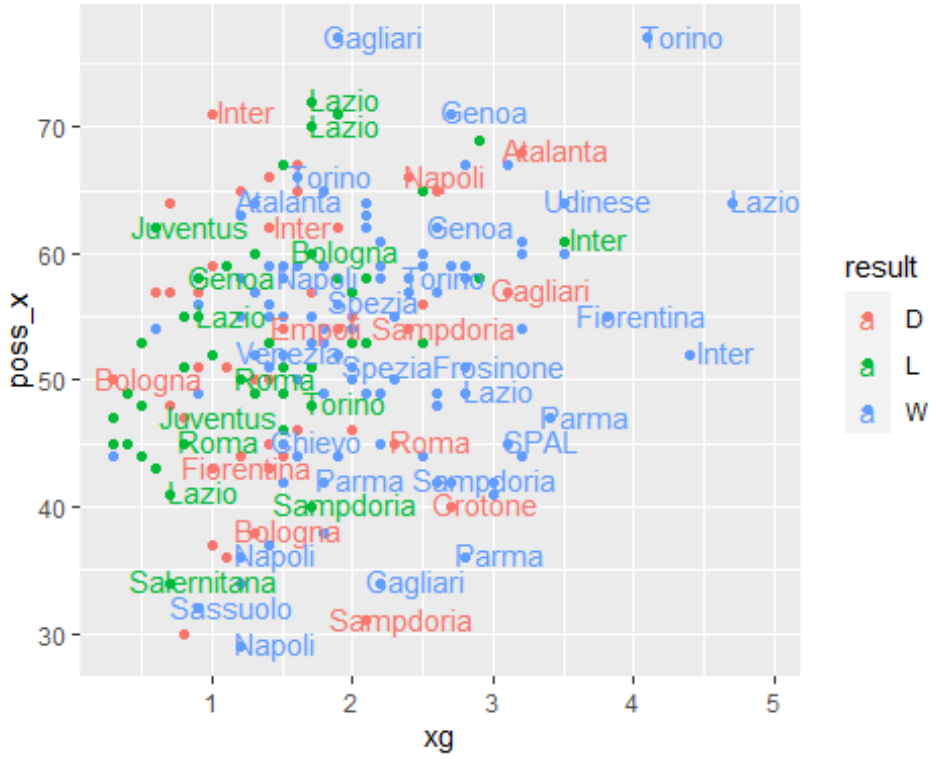

```
df_roma<-df_total
roma <- which(df_roma$team == "Roma")
df_roma<-df_roma[roma,]

summary(df_roma)
```

```
##      date               round              team              opponent
##  Length:190         Length:190         Length:190         Length:190
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     venue              result                gf              ga
##  Length:190         Length:190         Min.   :0.000   Min.   :0.0
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.0
##  Mode  :character   Mode  :character   Median :2.000   Median :1.0
##                                        Mean   :1.742   Mean   :1.2
##                                        3rd Qu.:3.000   3rd Qu.:2.0
##                                        Max.   :6.000   Max.   :4.0
##        xg              xga            poss_x            sh
##  Min.   :0.300   Min.   :0.000   Min.   :29.00   Min.   : 5.00
##  1st Qu.:1.200   1st Qu.:0.700   1st Qu.:47.00   1st Qu.:12.00
##  Median :1.700   Median :1.100   Median :54.00   Median :16.00
##  Mean   :1.785   Mean   :1.221   Mean   :53.19   Mean   :16.09
##  3rd Qu.:2.300   3rd Qu.:1.600   3rd Qu.:59.00   3rd Qu.:19.00
##  Max.   :4.700   Max.   :3.700   Max.   :77.00   Max.   :46.00
##       att             succ.
```

```
##  Min.   : 4.00   Min.   :10.00
##  1st Qu.:13.00   1st Qu.:50.00
##  Median :17.00   Median :58.30
##  Mean   :18.17   Mean   :57.69
##  3rd Qu.:22.00   3rd Qu.:68.40
##  Max.   :41.00   Max.   :87.50

dribbling_plot(df_roma)
```




```
possess(df_roma)
```

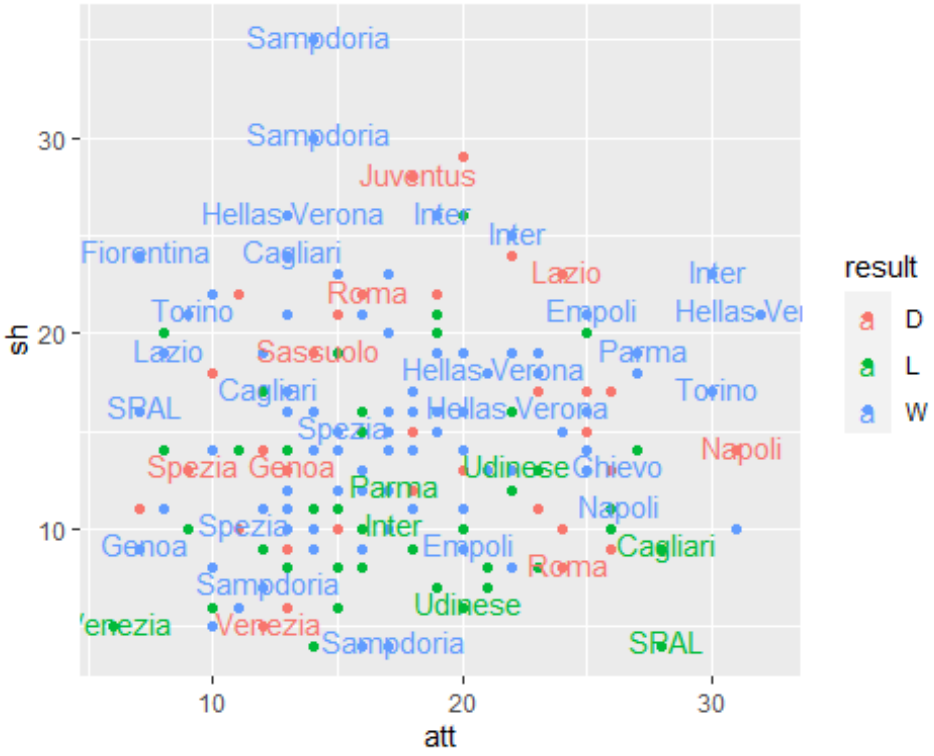



```
df_lazio<-df_total
lazio <- which(df_lazio$team == "Lazio")
df_lazio<-df_lazio[lazio,]

summary(df_lazio)
```

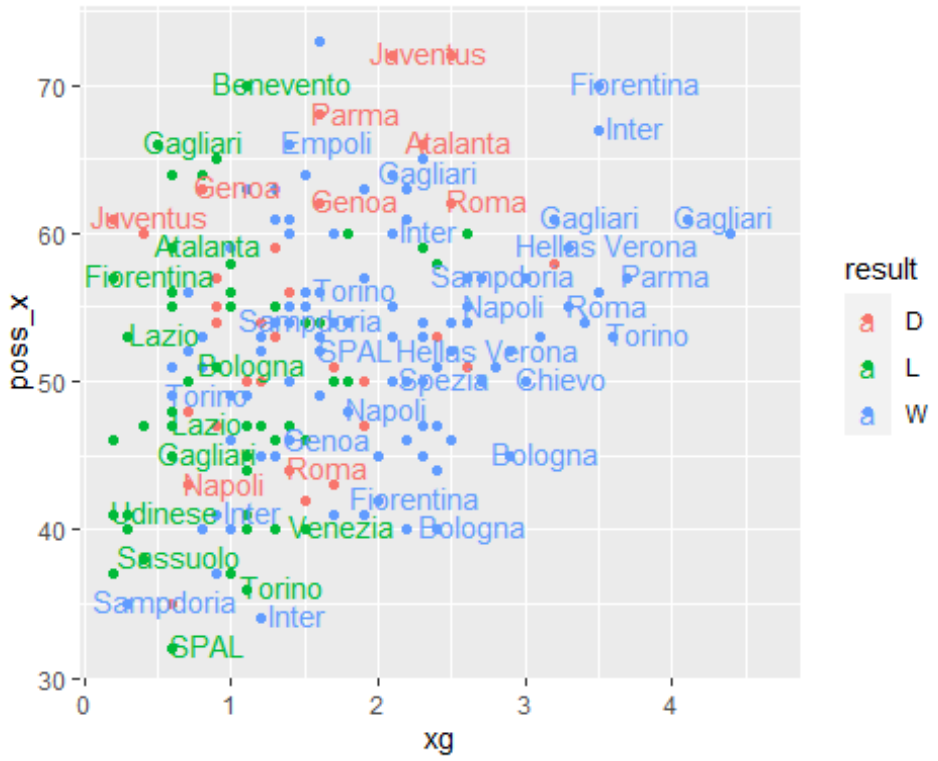
```
##      date               round              team             opponent
##  Length:190         Length:190         Length:190         Length:190
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     venue              result                gf              ga
##  Length:190         Length:190         Min.   :0.000   Min.   :0.000
##  Class :character   Class :character   1st Qu.:1.000   1st Qu.:0.000
##  Mode  :character   Mode  :character   Median :2.000   Median :1.000
##                                        Mean   :1.905   Mean   :1.316
##                                        3rd Qu.:3.000   3rd Qu.:2.000
##                                        Max.   :6.000   Max.   :5.000
##        xg              xga            poss_x           sh
##  Min.   :0.200   Min.   :0.200   Min.   :32.00   Min.   : 4.00
##  1st Qu.:1.000   1st Qu.:0.700   1st Qu.:46.25   1st Qu.:10.00
##  Median :1.400   Median :1.150   Median :52.50   Median :14.00
##  Mean   :1.602   Mean   :1.223   Mean   :52.11   Mean   :14.21
##  3rd Qu.:2.200   3rd Qu.:1.600   3rd Qu.:57.00   3rd Qu.:18.00
##  Max.   :4.400   Max.   :3.000   Max.   :73.00   Max.   :35.00
##       att             succ.
```

```
##  Min.   : 6.00   Min.   :21.40
##  1st Qu.:13.00   1st Qu.:48.00
##  Median :16.00   Median :60.00
##  Mean   :17.09   Mean   :58.84
##  3rd Qu.:21.00   3rd Qu.:69.10
##  Max.   :32.00   Max.   :92.90
```

```
dribbling_plot(df_lazio)
```

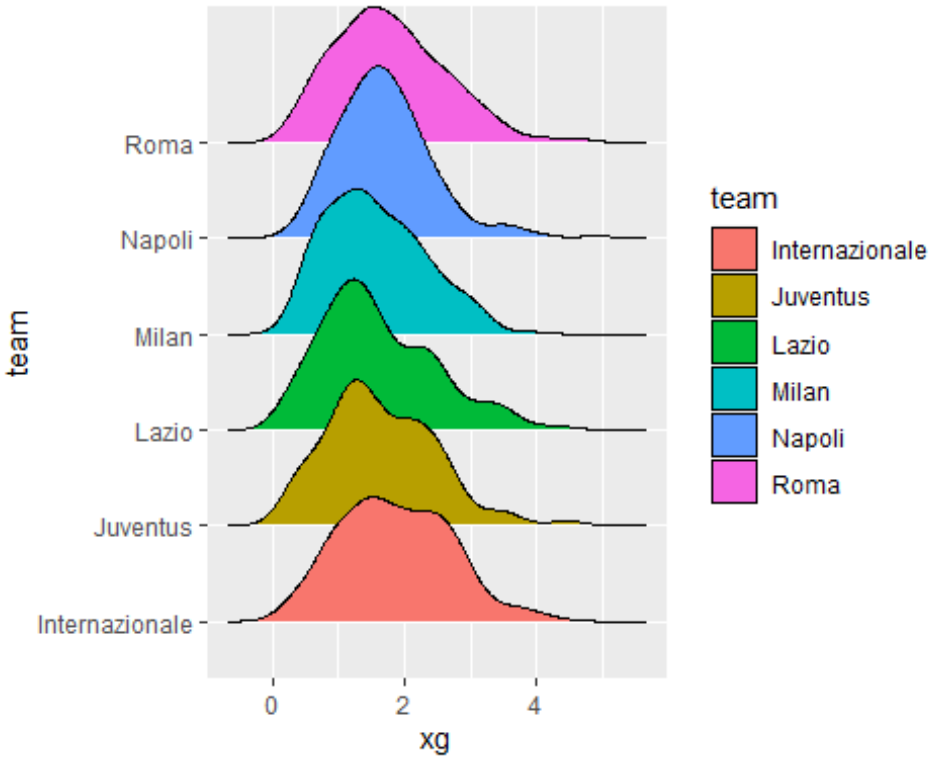



```
possess(df_lazio)
```



The following graph compares the distribution of Xg for each analyzed team

```
df_ridge<- df_football
ridge<-which(df_ridge$team == "Milan" | df_ridge$team=="Internazionale"|
               df_ridge$team == "Lazio" | df_ridge$team=="Roma"|
               df_ridge$team == "Napoli" | df_ridge$team=="Juventus")
df_ridge<-df_ridge[ridge,]
ggplot(df_ridge, aes(x=xg, y=team,fill = team))+
  geom_density_ridges()
```



Napoli and Roma are the only team that has a similar distribution to the normal one with a peak at 1.6 Xg and 1.5 Xg. Inter is the team that has created more Xg as a second peak corresponds to 2.5 Xg.

From this graph it would seem that Napoli produces more Xg than Roma To prove this hypothesis, we work with the t-test on the difference of the mean of two samples independent. Having N = 190 greater than 30, by the central limit theorem, the data is distribute as a normal standard. so the two hypothesis are :

H0: xg_Roma >= xg_Napoli vs H1: xg_Roma < xg_Napoli

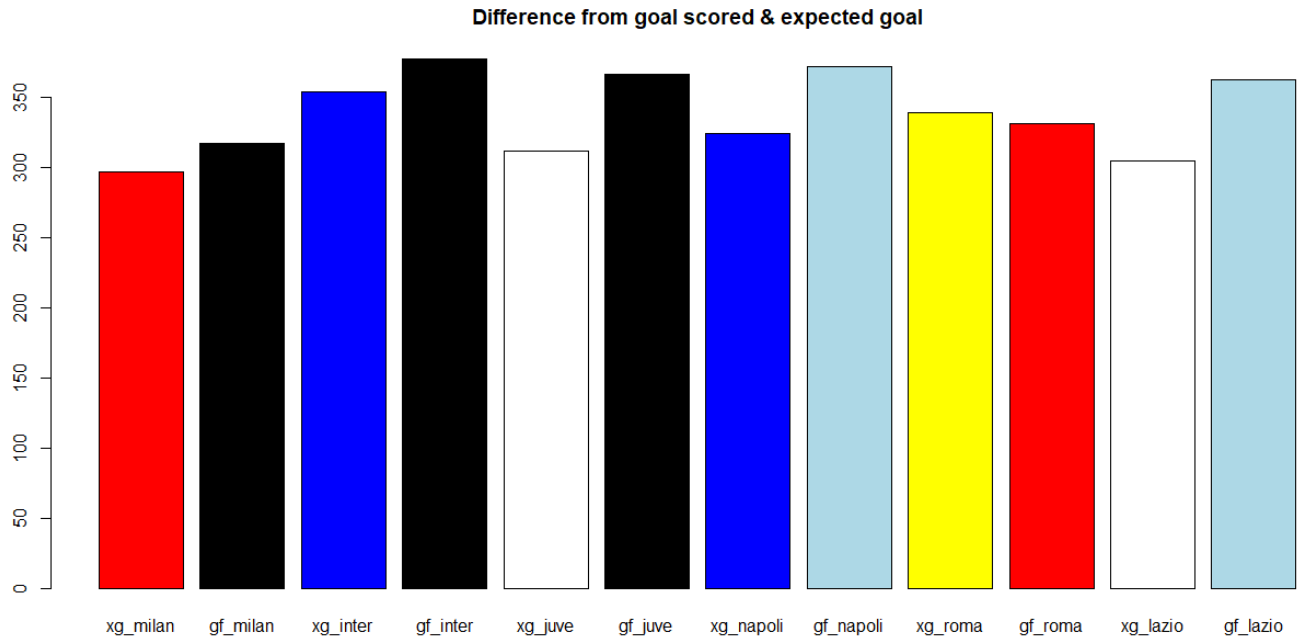
```
t.test(df_point_roma$xg, df_point_napoli$xg, alternative = "less", conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  df_point_roma$xg and df_point_napoli$xg
## t = 0.96432, df = 368.87, p-value = 0.8322
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.2110947
## sample estimates:
## mean of x mean of y
##  1.784737  1.706842
```

The test yields a p-value = 0.8322. Having alpha = 0.05 p-value> alpha then the test accepts H0. Consequently, it cannot be said that Naples produces more than Rome.

This can be also seen from the following graph. This barplot is displayed in pairs. For each team the first bar represents the Xg while the second represents the goals scored

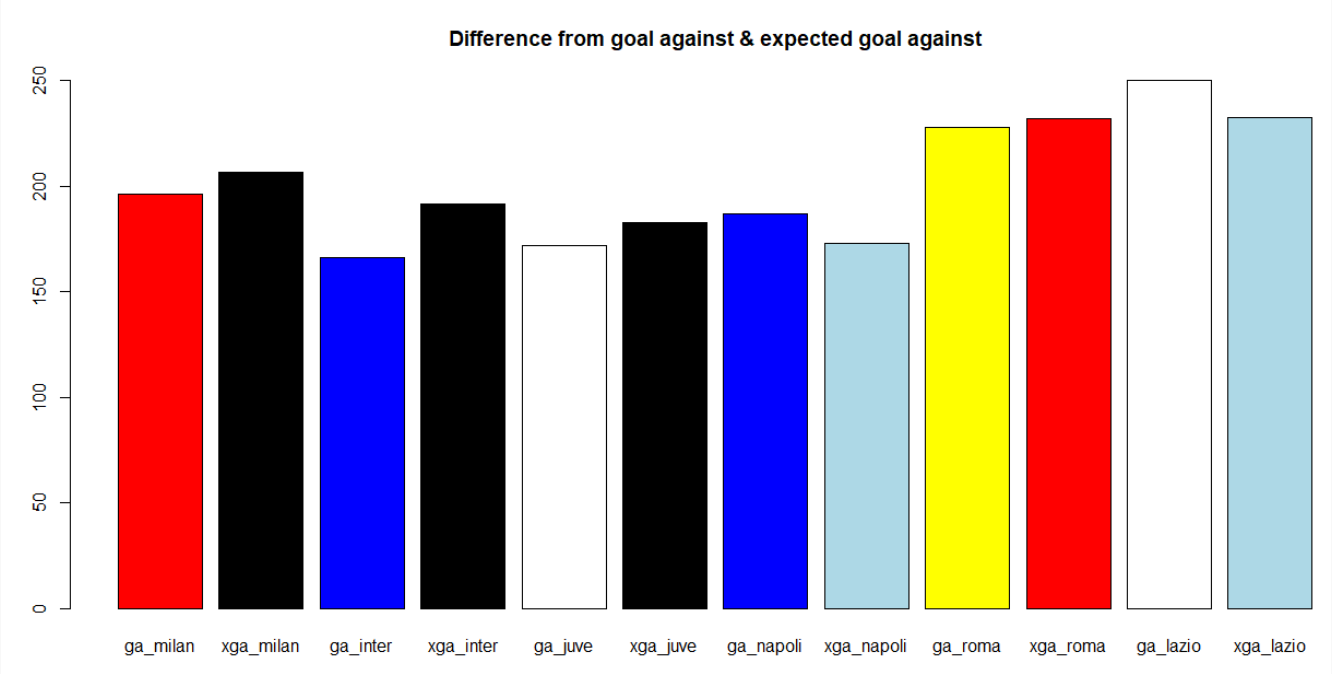
```
df_goal<-merge_goal()
df_goal_ag<-merge_goal_ag()
barplot.default(df_goal,border = "black",  col=c("red","black","blue","black",
                                                 "white","black","blue","light
blue",
                                                 "yellow","red","white","light
blue"),
                main="Difference from goal scored & expected goal",
                names.arg=c("xg_milan","gf_milan","xg_inter","gf_inter",
                            "xg_juve","gf_juve","xg_napoli","gf_napoli",
                            "xg_roma","gf_roma","xg_lazio","gf_lazio" ))
```



As mentioned before, Inter is the team that creates the most scoring chances and consequently the team that scores the most. Juve and Lazio are the teams that take advantage of the occasional goals most by having the greatest difference between goals scored and Xg. This is thanks to the cinecity of the forwards who take advantage of the most complicated occasions to score, just think of last year's top scorer, Immobile, who scored 27 goals, having 22.36 as Xg. The opposite happens for Rome which is the only one among the teams analyzed to have scored fewer goals than expected. This is due to the scarce ability of strikers like dzeko who in his last 3 seasons at Roma has produced respectively 14.25, 21.15, 15.17 Xg scoring 7, 16, 9.

Similarly, the goals conceded and the expected goals conceded will be analyzed.

```
barplot.default(df_goal_ag,border = "black",  col=c("red","black","blue","black",
                                                    "white","black","blue","light
blue",
                                                    "yellow","red","white","light
blue"),
                main="Difference from goal against & expected goal against",
                names.arg=c("ga_milan","xga_milan","ga_inter","xga_inter",
                            "ga_juve","xga_juve","ga_napoli","xga_napoli",
                            "ga_roma","xga_roma","ga_lazio","xga_lazio"))
```



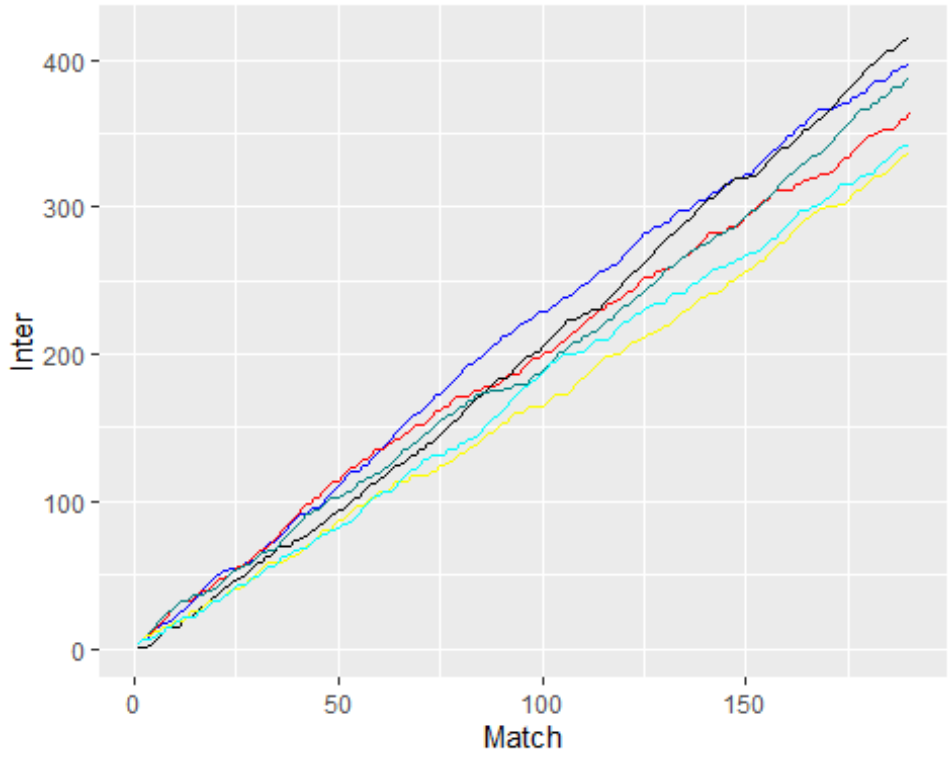
To win the championships you need a good defense, in fact Inter, Milan and Juve, the last 3 winners of the championship, are the only ones to have a clear difference between goals conceded and expected goals conceded. Scoring is important, but not conceding a goal is even more important.

Lastly, I would like to try to predict the point obtained at the end of the year by the 6 teams (the trend of this start of the championship was not taken into consideration, the input data correspond to the last 5 championships).

First of all, I plot the progress of the teams over the 5 years, each team is therefore represented by its own jersey color Juve = black; Inter = blue; Milan = red; Rome = yellow; lazio = heavenly; Naples = teal;

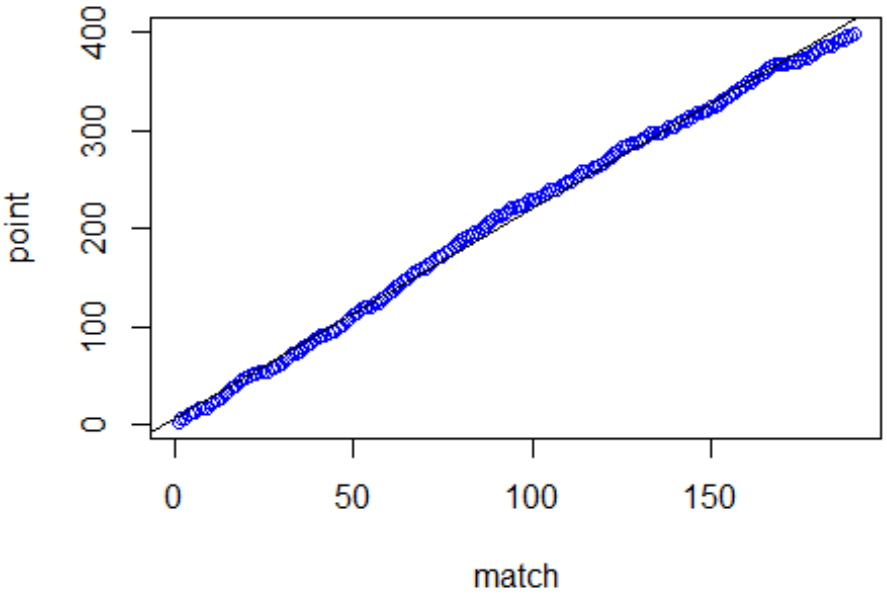
```
df_point_all<- data.frame(Match=df_point_inter$match,
                          Inter=df_point_inter$point,Milan=df_point_milan$point,
                          Juve=df_point_juve$point,Napoli=df_point_napoli$point,
                          Roma=df_point_roma$point,Lazio=df_point_lazio$point)

ggplot(df_point_all, aes(x=Match)) +
```

```
  geom_line( aes(y=Inter),color="#0000FF") +
  geom_line( aes(y=Milan),color="#FF0000")+
  geom_line( aes(y=Juve),color="#000000")+
  geom_line( aes(y=Napoli),color="#008080")+
  geom_line( aes(y=Roma), color="#FFFF00")+
  geom_line( aes(y=Lazio), color="#00FFFF")
```



Then we give an explanation of the linear regression model, using the points obtained by Inter as an example

```
reg_inter<-df_point_inter[c(15,16)]
lmTemp <- lm(point~match, data = reg_inter)
lmTemp

##
## Call:
## lm(formula = point ~ match, data = reg_inter)
##
## Coefficients:
## (Intercept)        match
##       6.960        2.131

plot(reg_inter, col = "blue")+
abline(lmTemp)
```

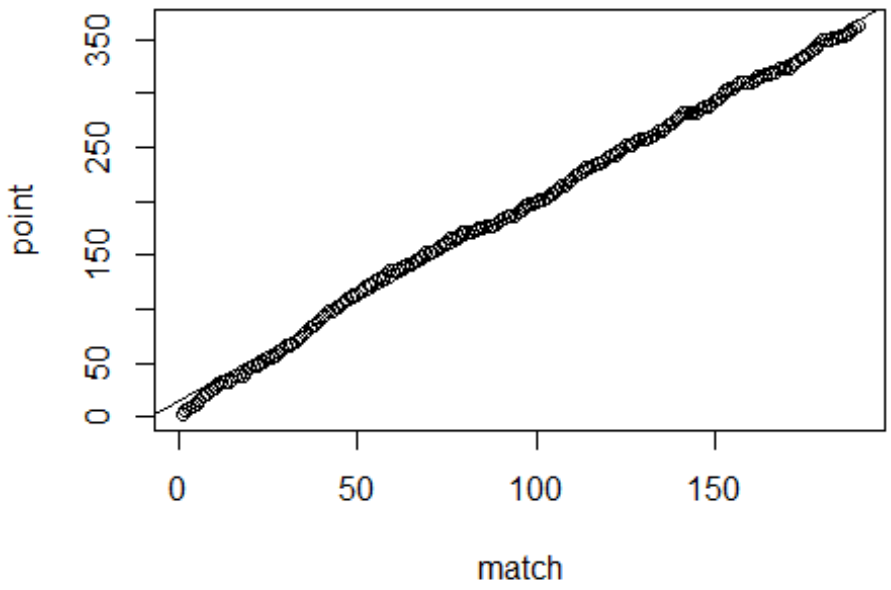



```
## integer(0)

summary(lmTemp)

##
## Call:
## lm(formula = point ~ match, data = reg_inter)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6837  -5.1932  -0.8233   5.5634  13.2668
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.959900   0.959833   7.251 1.05e-11 ***
## match       2.130814   0.008715 244.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.589 on 188 degrees of freedom
## Multiple R-squared:  0.9969, Adjusted R-squared:  0.9968
## F-statistic: 5.977e+04 on 1 and 188 DF,  p-value: < 2.2e-16
```

Among the information that summary provides, there is the R-squared value, which is the proportion of variance of the dependent variable that is explained by the predictor. This value is between 0 and 1. Values close to 1 indicate a good model. On average, an increase of 2,130814 points is expected for each match.

Now it's time to give a prediction of the six teams in analysis:

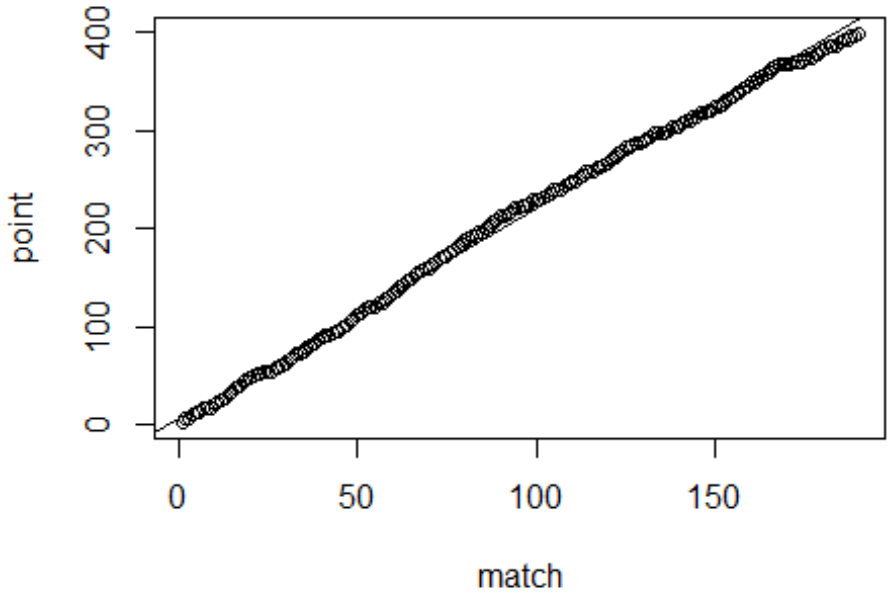
```
prediction_milan<-lr(df_point_milan)
```



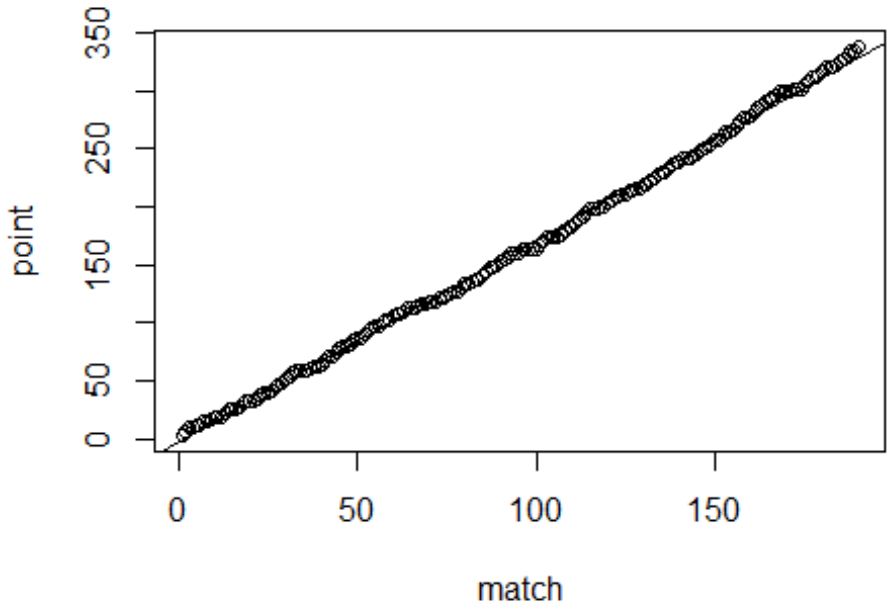

```
prediction_milan
```

```
## 38
## 69
```

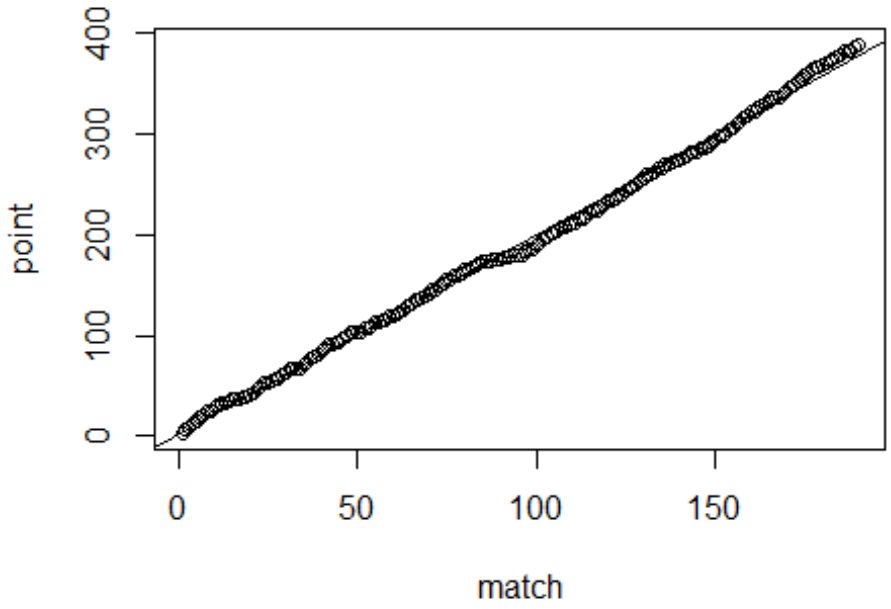
```
prediction_inter<-lr(df_point_inter)
```



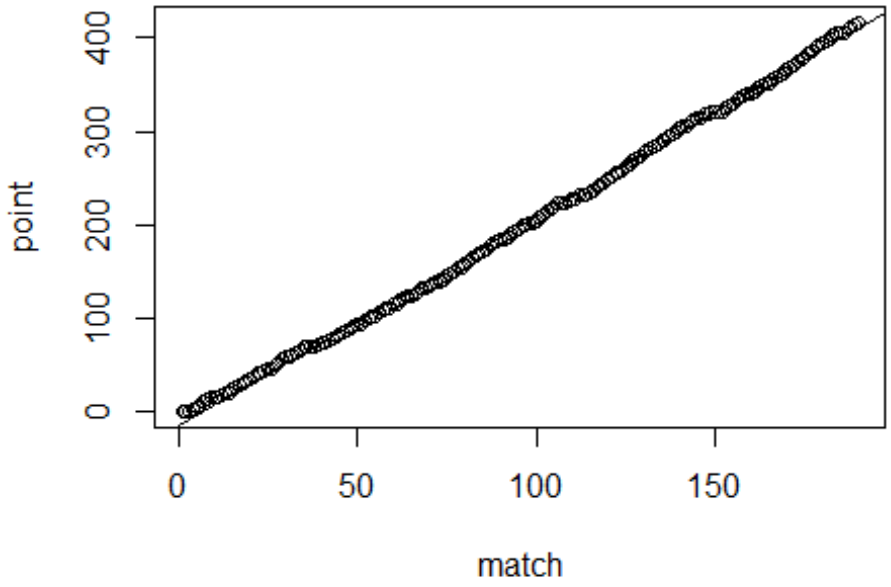

```
prediction_inter

## 38
## 79

prediction_roma<-lr(df_point_roma)
```



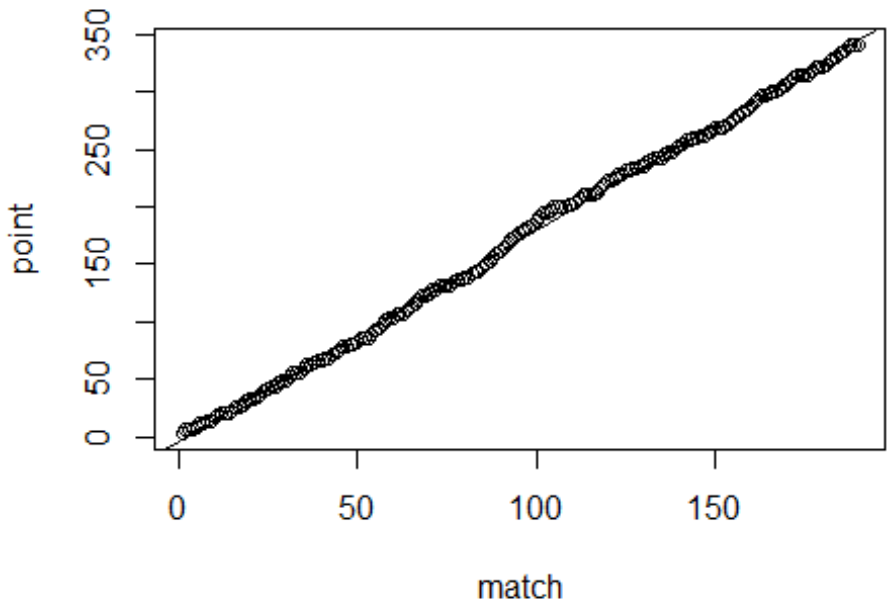

```
prediction_roma

## 38
## 65

prediction_napoli<-lr(df_point_napoli)
```




```
prediction_napoli

## 38
## 73

prediction_juve<-lr(df_point_juve)
```



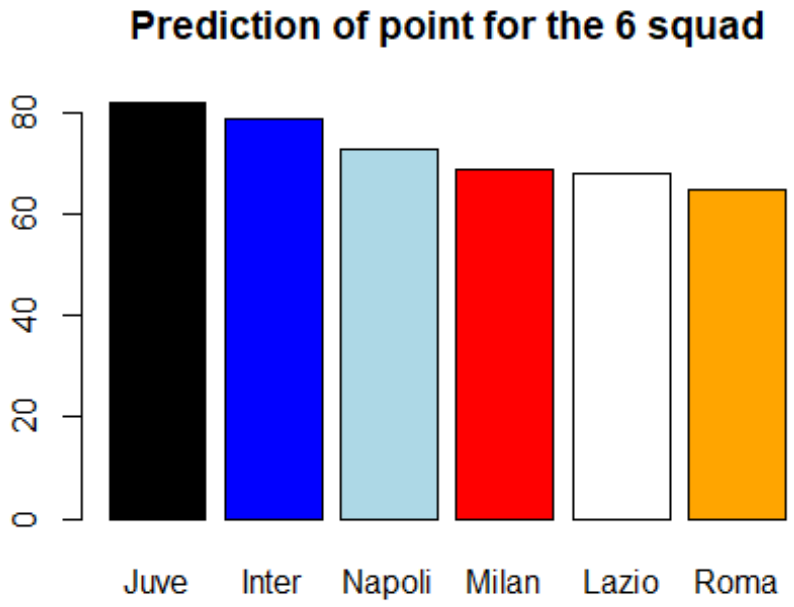

```
prediction_juve

## 38
## 82

prediction_lazio<-lr(df_point_lazio)
```




```
prediction_lazio
```

```
## 38
## 68
```

```
total_prediction <- c(prediction_juve,prediction_inter,prediction_napoli,
                      prediction_milan,prediction_lazio,prediction_roma)
barplot.default(total_prediction,border = "black",  col=c("black","blue","light
blue",
                                                          "red","white","orange"),
                main="Prediction of point for the 6 squad",
                names.arg=c("Juve","Inter","Napoli","Milan","Lazio","Roma" ))
```



The model at the beginning of the championship would have expected Juventus to win the championship, with a champions zone composed of Inter, Naples and Milan, ending with Lazio and Rome.

This is a small example of how data can be used on a sport, and how in general, data analysis is becoming increasingly important in any field.