STMO Final Project

Victor Garcia Rubio

19 de diciembre de 2018

1 DATASET

For the final project, I have choosen a public dataset from *https://datahub.io/* . In this case, the selected data comes from **Spanish LaLiga**, the organization which manages the main football competition in Spain. The dataset contains different categories of data, regarding two clearly separeted types: Football statistics and betting statistics. As I have more interest in football statistics than bets, I have only selected the categories regarding this type. The url of the dataset is: *https://datahub.io/sports-data/spanish-la-liga* . In this case, we have selected the matches correspoding to **season 2017-2018**, in order to have the most complete and recent data from teams. The different information of each match collected on the dataset is described in the following table:

Label	**Description**
Date	Date of the match
HomeTeam	Home Team of the match
AwayTeam	Away Team of the match
FTHG	Full Time Home Team Goals
FTAG	Full Time Away Team Goals
FTR	Full Time Result (H=Home Win, D=Draw, A=Away Win)
HTHG	Half Time Home Team Goals
HTAG	Half Time Away Team Goals
HTR	Half Time Result (H=Home Win, D=Draw, A=Away Win)
HS	Home Team Shots
AS	Away Team Shots
HST	Home Team Shots on Target
AST	Away Team Shots on Target
HF	Home Team Fouls Committed
AF	Away Team Fouls Committed
HC	Home Team Corners
AC	Away Team Corners
HY	Home Team Yellow Cards
AY	Away Team Yellow Cards
HR	Home Team Red Cards
AR	Away Team Red Cards

First of all, I had to make the selection of the categories to work with. To do so, the subset function is used along with the corresponding names of the categories listed above.

```
data <- read.csv('~/data/datasets/spanish-laliga/archive/season-1718.csv')
football_data <- subset(data,select = c(Date,HomeTeam,AwayTeam,FTHG,FTAG,FTR,HTHG,HTAG,HTR,
                                        HS,AS,HST,AST,HF,AF,HC,AC,HY,AY,HR,AR))
```

An interesting list to have in order to manage the data is the list of teams. This is extracted using the unique function as follows:

```
teams <- as.character(unique(football_data[,"HomeTeam"]))
```

2 Team Analysis

To begin with the analysis, I have decided to start with only one team to simplify the operations. In further parts of the project I will be managing data from all teams. In this case I have elected **S.D Eibar** as my team to analyse. The information is splitted in two different dataframes: one for the matches played as *Home Team* and the other for the matches played as *Away Team*. For sake of order, I have splitted each operation in different functions.

2.1 Descriptive Statistics

The first part of the team analysis is to obtain descriptive statistics from each feature (excluding date and team names). Using the following computeStats function, the next features are obtained an grouped on a new dataframe:

1. Mean
2. Standard Deviation
3. Variance
4. Maximum
5. Minimum

The code of the function is as follows:

```
computeStats <- function(x){
  stats <- data.frame(matrix(ncol = 16, nrow = 6))
  int_cols <- c("FTHG","FTAG","HTHG","HTAG","HS","AS","HST",
                "AST","HF","AF","HC","AC","HY","AY","HR","AR")
  colnames(stats) <- int_cols
  for (col in int_cols){
    mean <- mean(x[,col])
    std <- sd(x[,col])
    var <- var(x[,col])
    max <- max(x[,col])
    min <- min(x[,col])
    median <- median(x[,col])
    stats[,col] <- c(mean,std,var,max,min,median)
  }
  return(stats)
}
```

The mentioned function is **scalable to the complete dataset** with all teams and will be used on further parts of the project.

2.2 Confidence intervals

From this statistics, a more extensive analysis can be performed. In this case, several interesting confidence intervals are obtained using different methods. Firstly, we have checked the normality of the data usign the saphiro test. The data retrieves high values of normality, which permits the assumption of normal distributions. In homeStatsAnalysis and awayStatsAnalysis functions, the following statistis are obtained:

1. Confidence Interval(95%) of the yellow cards means, assuming a normal distribution
2. Confidence Interval(95%) of the yellow cards means, without assuming any distribution by using boostrap method
3. Correlation between away team yellow cards,away team fouls, home team yellow cards, and home team fouls.

The code to obtain the CIs, as well as the mentioned functions:

```
#Calculate CI from a normal distribution
getCINormal <- function(mean,std,CI,n){
  interval = CI+(1-CI)/2
  error <- qnorm(interval)*std/sqrt(n)
  left <- mean-error
  right <- mean+error
  return(c(left,right))
}

#Calculate CI from an unknown distribution using bootstrap method
getCIUnknown <- function(data){
  iterations = 40
  #generate resamples
  estimated_means <- c(1:iterations)
  sample_mean <- mean(data)
  for (it in seq(1,iterations,by=1)){
    #resample
    it_samples = sample(data,size=length(data),replace=TRUE)
    mean_it_samples = mean(it_samples)
    #resample mean. add to estimations array
    estimation <- mean_it_samples - sample_mean
    estimated_means[it] <- estimation
  }
}

#Compute CIs of home data
homeStatsAnalysis <- function(data_home,stats_home){
  # Check normality of data
  shapiro.test(data_home["HY"])
  shapiro.test(data_home["HF"])

  ci_hy_normal <- getCINormal(stats_home[1,"HY"],stats_home[2,"HY"],.95,nrow(stats_home))
  mean_test <- stats_home[1,"HY"]
  ci_hy_unknown <- getCIUnknown(data_home[,"HY"])
  home_yf <- cbind(data_home[,"HY"],data_home[,"HF"],data_home[,"AY"],data_home[,"AF"])
  cor_home_yf <- cor(home_yf)
  returnList <- list("CINormal" = ci_hy_normal, "CIUnk" = ci_hy_unknown, "Corr" = cor_home_yf)
  return(returnList)
}

#Compute CIs of home data
awayStatsAnalysis <- function(data_away,stats_away){
  # Check normality of data
  shapiro.test(data_home["AY"])
  shapiro.test(data_home["AF"])

  ci_ay_normal <- getCINormal(stats_away[1,"AY"],stats_away[2,"AY"],.95,nrow(stats_away))
  mean_test <- stats_away[1,"AY"]
  ci_ay_unknown <- getCIUnknown(data_away[,"AY"])
  away_yf <- cbind(data_away[,"HY"],data_away[,"HF"],data_away[,"AY"],data_away[,"AF"])
  cor_away_yf <- cor(away_yf)
  returnList <- list("CINormal" = ci_ay_normal, "CIUnk" = ci_ay_unknown, "Corr" = cor_away_yf)
```

```
  return(returnList)
}
```

The obtained results are:

Home

CI Normal: 1.110637 2.994627

CI Bootstrap: 2.263158 2.631579

Correlation Table:

	HY	HF	AY	AF
HY	1.0000000	0.53214226	-0.28270042	-0.26382339
HF	0.5321423	1.00000000	-0.05310801	-0.09279231
AY	-0.2827004	-0.05310801	1.00000000	0.52659360
AF	-0.2638234	-0.09279231	0.52659360	1.00000000

Away

CI Normal: 1.436416 3.300426

CI Bootstrap: 2.368421 2.421053

Correlation Table:

	HY	HF	AY	AF
HY	1.0000000	0.515619209	0.203687987	0.1857249
HF	0.5156192	1.000000000	0.005637495	0.4892399
AY	0.2036880	0.005637495	1.000000000	0.2742541
AF	0.1857249	0.489239877	0.274254073	1.0000000

2.3 Linear Regression

As one of the most concurrent methods nowadays, linear regression is applied in this project in order to obtain the relation and predict one of the most important things in football: goals. To do so, we have obtained the two interesting linear models along with descriptive values (histograms): Shoots on Target ~ Shoots Attempted and Full Time Goals ~ Shoots on Target. With these model, the accuracy of our players can be easily obtained. Both models are applied to the away data and home data as in previous sections. The plots along with the plots are described next:

HOME

The coefficients obtained are:

Shoots on Target ~ Shoots Attempted

lm(formula = data_home[, "HST"] ~ data_home[, "HS"], data = data_home)

Coefficients: (Intercept): 0.1768 data_home[, "HS"] : 0.2860

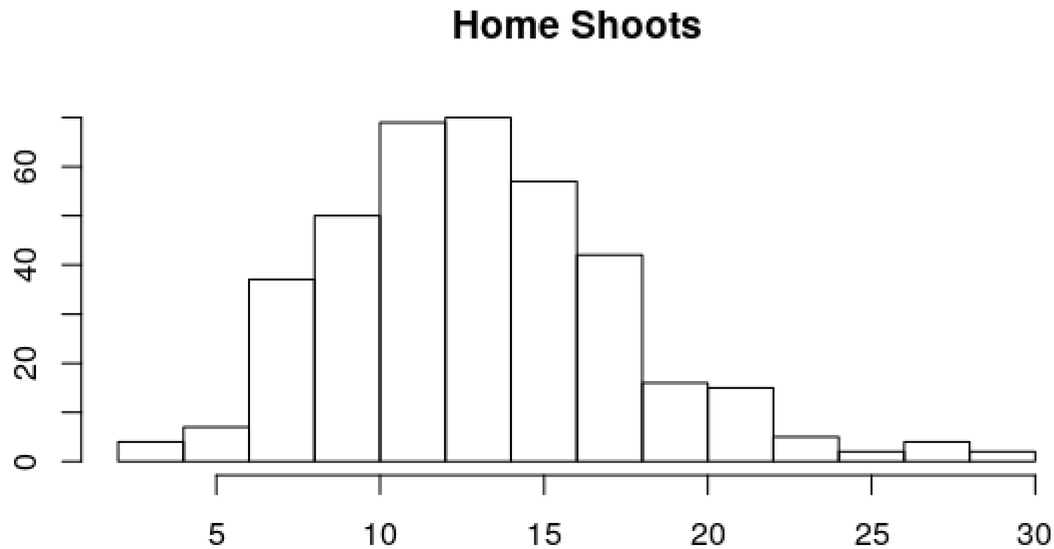

Figure 1: Home Shoots

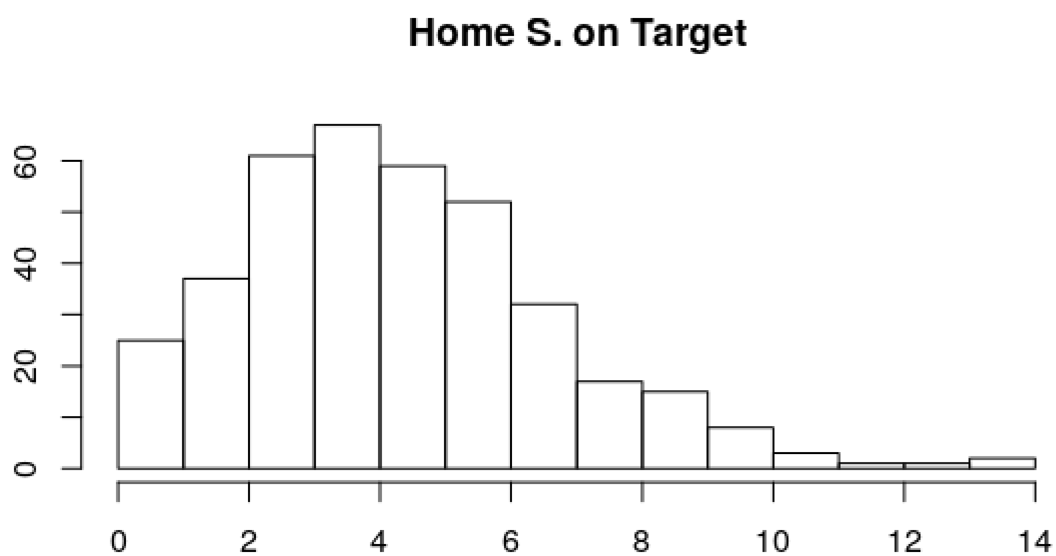

Figure 2: Home Shoots on Target

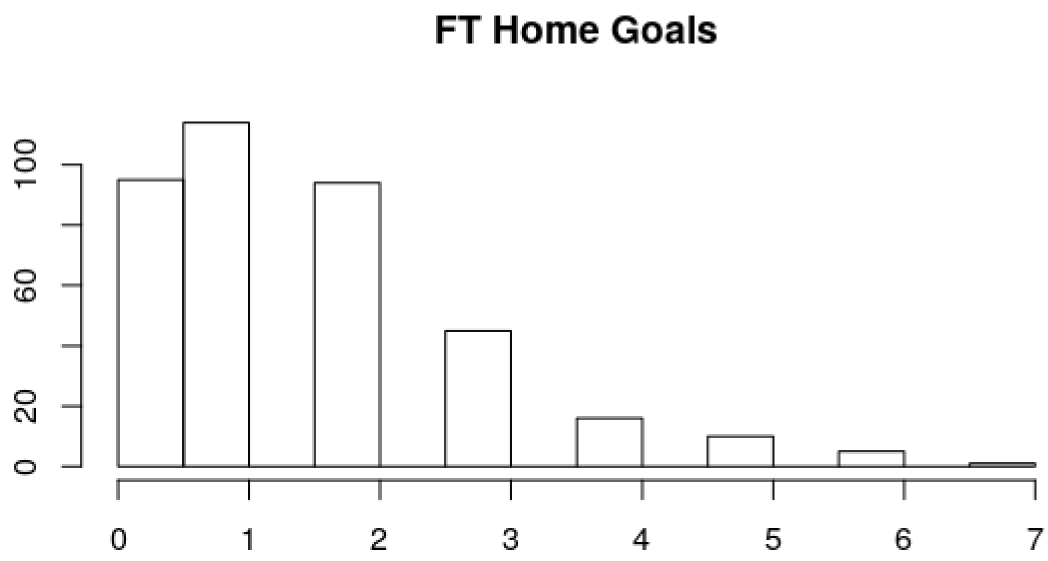

Figure 3: Full Time Home Goals

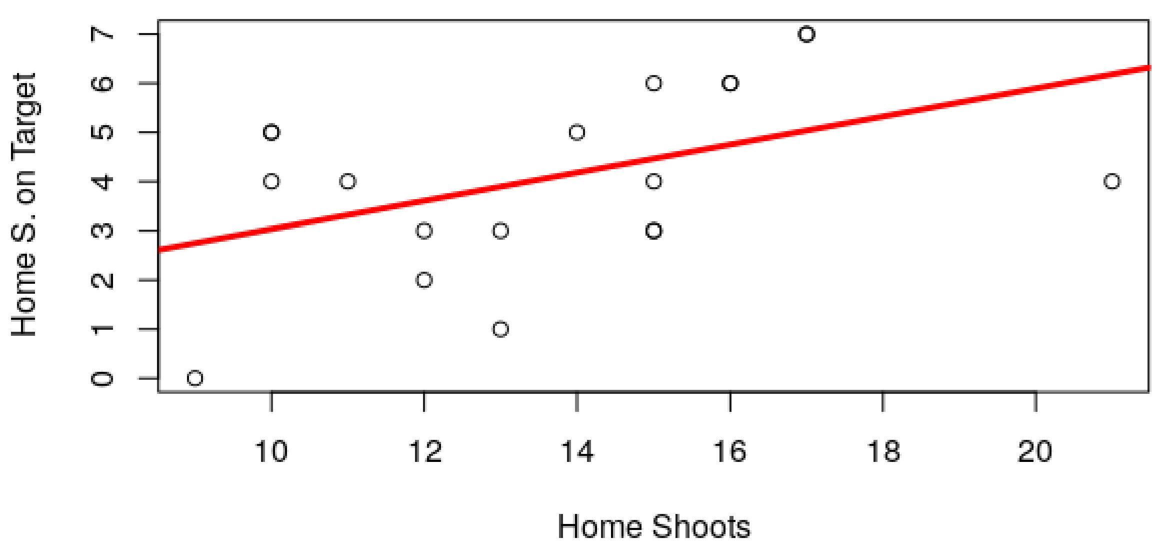

Figure 4: Linear Regression: Home Shoots on Target ~ Home Shoots

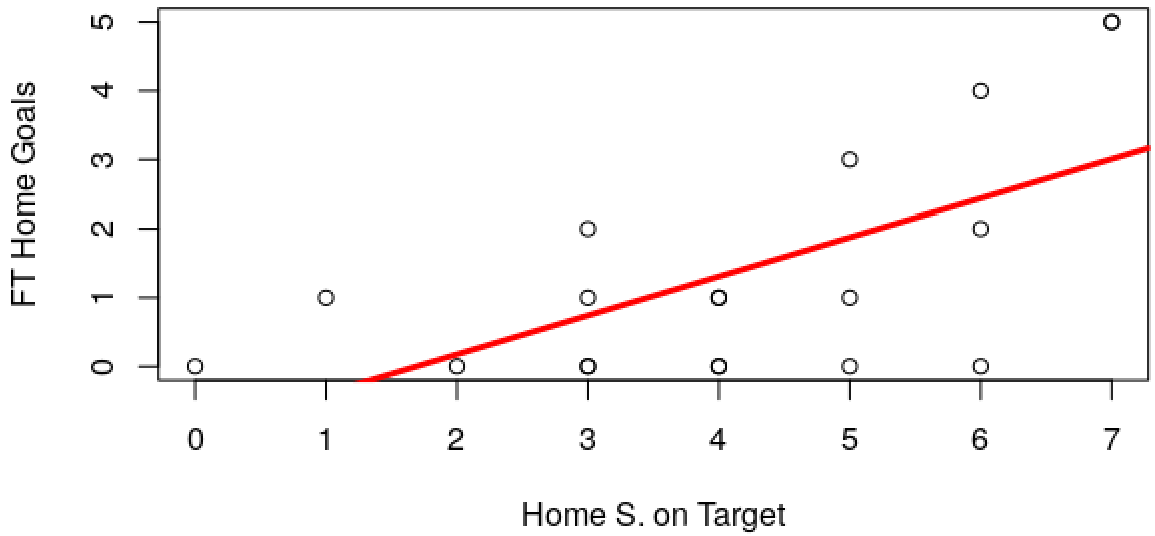

Figure 5: Linear Regression: Full Time Home Goals ~ Home Shoots on Target

Full Time Goals ~ Shoots on Target

lm(formula = data_home[, “FTHG”] ~ data_home[, “HST”], data = data_home)

Coefficients: (Intercept): -0.9568 data_home[, “HST”] 0.5664

AWAY

The coefficients obtained are:

Shoots on Target ~ Shoots Attempted

lm(formula = data_away[, “AST”] ~ data_away[, “AS”], data = data_away)

Coefficients: (Intercept): 0.3917 data_away[, “AS”] 0.2631

Full Time Goals ~ Shoots on Target

lm(formula = data_away[, “FTAG”] ~ data_away[, “AST”], data = data_away)

Coefficients: (Intercept):-0.3423 data_away[, “AST”]: 0.3829

3 League Analysis

As mentioned on previous section, a individual team analysis is performed in order to simplify some concepts and reduce code size. However, the purpose of this project is to analyse as depth as possible the teams involved and use as much relevant information as possible.

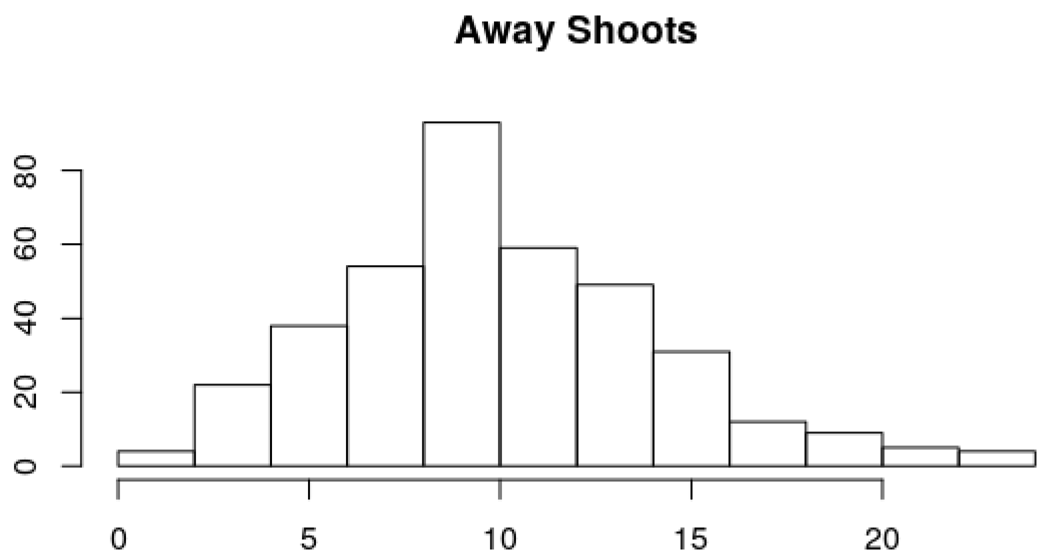

Figure 6: Away shoots

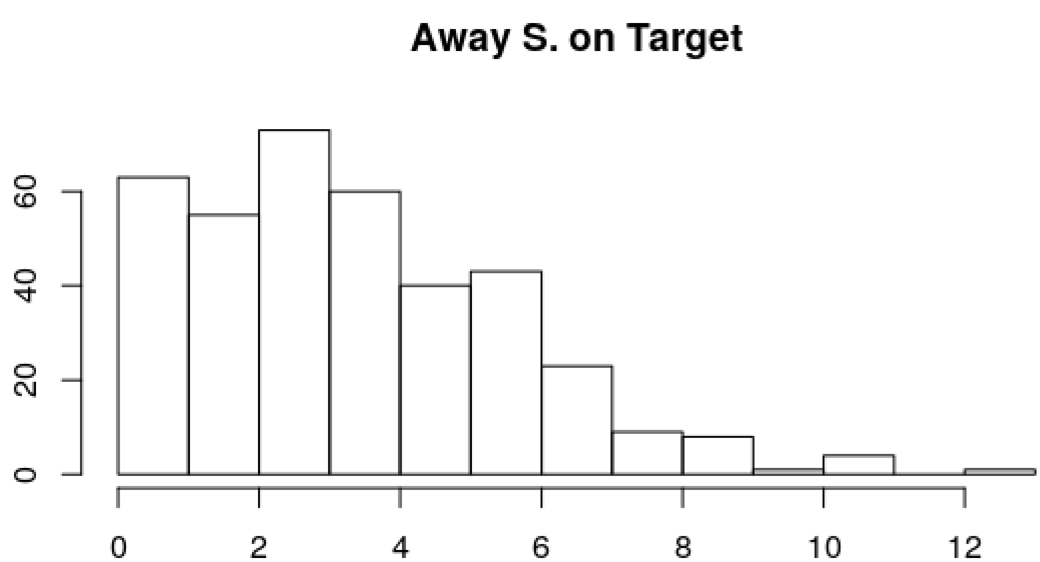

Figure 7: Away shoots on target

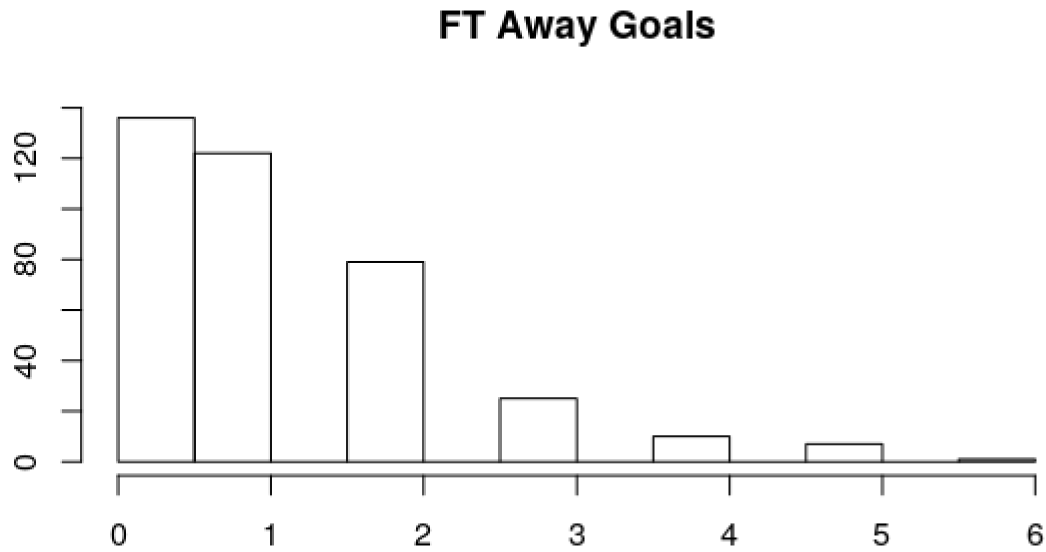

Figure 8: Full Time away goals

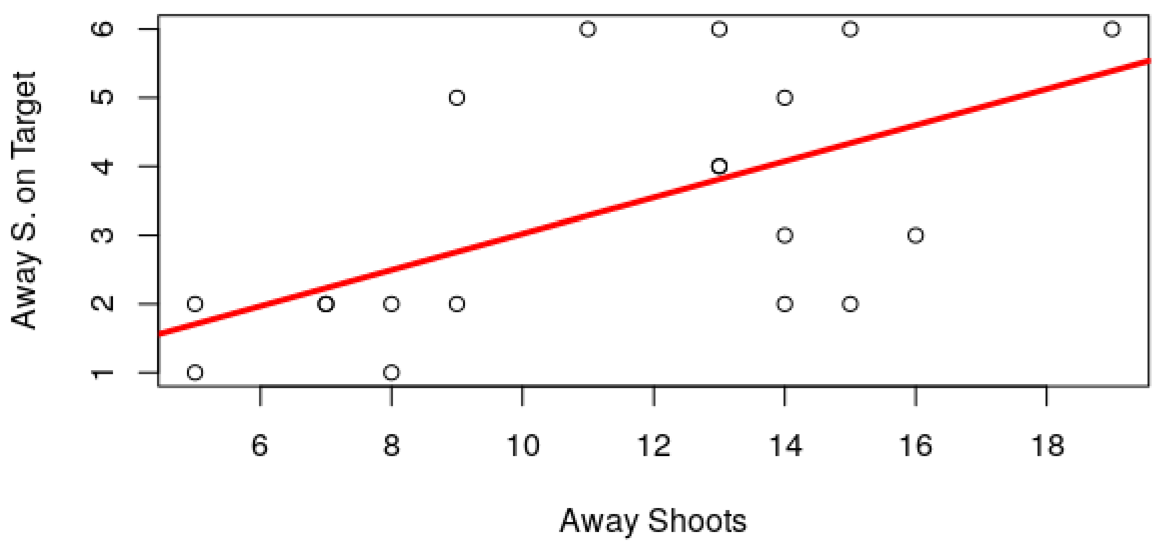

Figure 9: Linear Regression: Away Shoots on Target ~ Away Shoots

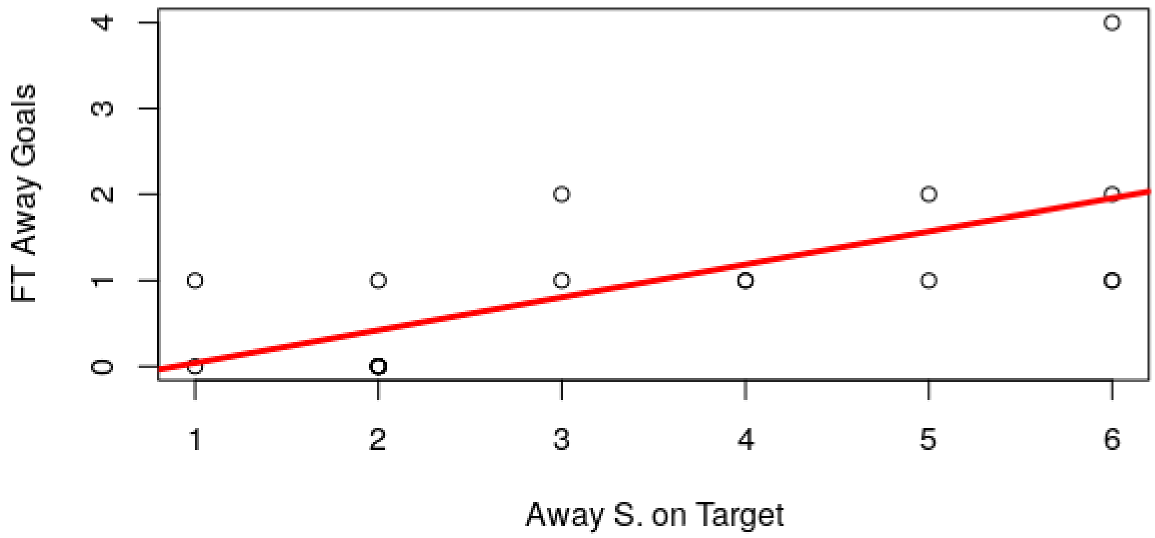

Figure 10: Linear Regression: Full Time Away Goals ~ Away Shoots on Target

3.1 Hypothesis Tests

To apply the studied concepts on the dataset, I have selected to make an hypothesis test to determine if the mean of the yellow cards is equal playing as a home team than as away team. Therefore, our **H0** is *mean_home = mean_away* and our **H1** is *mean_home not equal to mean_away*. First of all we obtain the mean yellow cards of each team, splitted in two rows: **Home mean and Away Mean**. Each column represents a team. The function getYellows performs the mentioned action and retrives the dataframe. The code of the function is described below:

```
#Get mean of yellow cards
getYellows <- function(teams,data){
  mean_cards <- data.frame(matrix(ncol = 0, nrow = 2))
  for (team in teams){
    hy <- football_data[football_data$"HomeTeam" == team,][,'HY']
    ay <- football_data[football_data$"AwayTeam" == team,][,'AY']
    mean_hy <- mean(hy)
    mean_ay <- mean(ay)
    mean_cards[,team] <- c(mean_hy,mean_ay)
  }
  return(mean_cards)
}
```

Secondly, a barplot is made to illustrate the mean yellow cards obtained by each team. Not all team names are plotted due to the fact that they are too many for the image size. Darkgray indicates *Home Yellow Cards Mean* and LightGray the *Away Yellow Cards Mean*.

We have selected is the welch test method to perform the hyphotesis tests. After that, a *Welch test* is applied to the obtained dataframe to solve the hypothesis test. The results are:

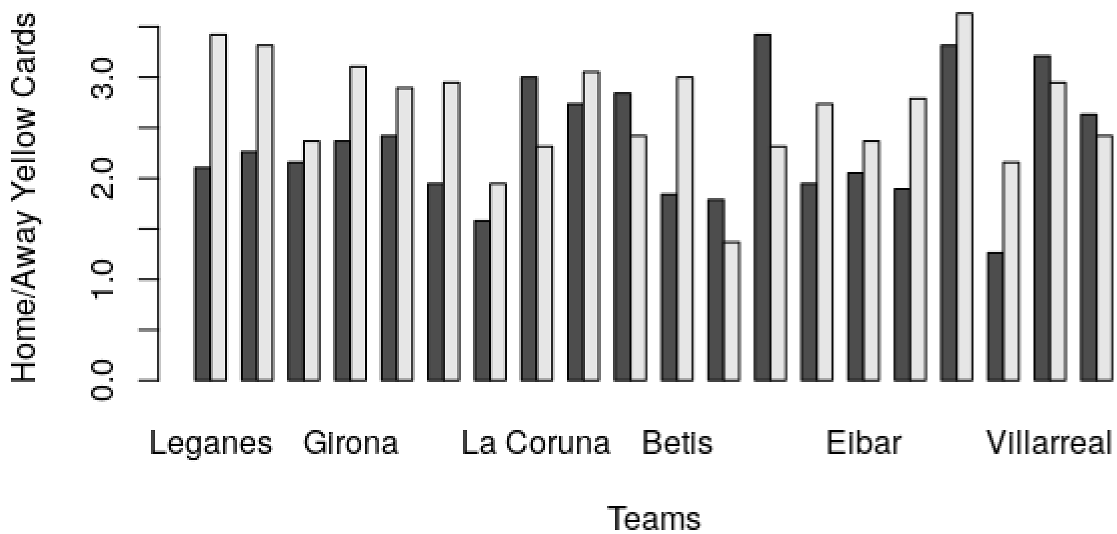

Figure 11: Yellow Cards Means Barplot

..* Statistic: -1.86488 ..* Parameter: 37.65817 ..* P-Value: 0.07000533 ..* Conf. Lvl: 0.99 ..* Conf. Int: -0.8268478 0.1531636 ..* Estimate: 2.339474 2.676316 ..* Method: "Welch Two Sample t-test" ..* Type: "two.sided"

The p-value is slightly greather than 0.05, which indicates that the H0 hyphotesis should not be rejected at first. However, it is close to the mentioned threshold.

3.2 Montecarlo Method

As one of the most recurrently used methods in Statistical Modelling's homeworks, I have made an implementation of the Montecarlo method from the information collected at the dataset. To do so, we have to generate a *Bernoulli* distribution from the dataset. This is made applicating the following criteria:

1 if the **number of corners are equal** between HomeTeam and AwayTeam 0 Otherwise

Therefore, applying this for: 500,1000,5000,10000 samples, we have generated the mean value of the distribution. Finally, we have applied a confidence interval in order to know how frequent is the mentioned event. The results of the application could be used for betting purposes.

The results of the application along with the needed code to generate them is described bellow:

..* 12312 ..* 1232 #Mean from Montecarlo Method: 0.194075

CI of 95% with 0.001 length: 0.1940624 0.1940876

Detailed Functions

```
#Yellow cards montecarlo
montecarlo<- function(data){

  mc_data <- 1 * (data[,1] == data[,2])
  hc_values <- seq(range(data[,1])[1],range(data[,1])[2],by=1)
  ac_values <- seq(range(data[,2])[1],range(data[,2])[2],by=1)

  home_probs <- numeric()
  away_probs <- numeric()

  for (home_value in hc_values){
    prob_value <- sum(data[,1] == home_value) / nrow(data)
    home_probs <- c(home_probs, prob_value)
  }
  for(away_value in ac_values){
    prob_value <- sum(data[,2] == away_value) / nrow(data)
    away_probs <- c(home_probs, prob_value)
  }
  mean_vector <- numeric()
  vector_samples <- c(500,1000,5000,10000)
  for (nsamples in vector_samples) {
    for (num in 1:nsamples){
      hc_sample <- sample(max(hc_values)+1, size = nsamples, replace = TRUE, prob = home_probs)
      hc_sample <- hc_sample - 1
      ac_sample <- sample(max(ac_values)+1, size = nsamples, replace = TRUE, prob = away_probs)
      ac_sample <- ac_sample - 1
      joined_sample <- cbind(hc_sample,ac_sample)
      num_duplicates <- 1 * (joined_sample[,1] == joined_sample[,2])
    }
    mean_duplicates <- mean(as.numeric(num_duplicates))
    mean_vector <- c(mean_duplicates, mean_vector)
  }
  return(mean(mean_vector))
}


#Montecarlo CI
getMontecarloCI95001 <- function(mean_montecarlo){

  n_samples_CI95_001 <- (1.96/.005) **2*(1-mean_montecarlo)*mean_montecarlo
  error1 <- qt(0.975,df=n_samples_CI95_001)*0.001/sqrt(n_samples_CI95_001)
  upper<- mean_montecarlo + error1
  lower <- mean_montecarlo - error1
  return(c(lower,upper))
}
```

Main Sequence

corner_data <- subset(football_data,select=c("HC", "AC")) mean_montecarlo <- montecarlo(corner_data) montecarlo_CI95_001 <- getMontecarloCI95001(mean_montecarlo)

4 Conclusions

We have analysed and interesting dataset from sports, more concretly the Spanish first division league of football. We have applied concepts susch as confidence interval, Montecarlo, Boostrap or hyphotesis test to the dataset in order to achieve relevan results to different areas: Information for matches, bets, and team relevant information for more professional issues.