End Sem

Nirav Mahnot

2022-12-09

#Loading Libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Used for data manipulation (creating train-test-valid split)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Loading required package: lattice
```

```
#The caret package (short for Classification And REgression Training) contains functions to stre
amline the model training process for complex regression and classification problems.
library(rpart)
#used for building classification and regression trees.
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.2.2
```

```
#Plot an rpart model, automatically tailoring the plot for the model's response type.
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✓ tibble  3.1.8     ✓ purrr   0.3.5
## ✓ tidyr   1.2.1     ✓ stringr 1.4.1
## ✓ readr   2.1.3     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
```

```
#The tidyverse is an opinionated collection of R packages designed for data science.
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
#It is plotting system based on the grammar of graphics.
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.2.2
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-4
```

```
#Lasso and Elastic-Net Regularized Generalized Linear Models
library(kknn)
```

```
##
## Attaching package: 'kknn'
##
## The following object is masked from 'package:caret':
##
##     contr.dummy
```

```
#R package for Weighted k-Nearest Neighbors Classification, Regression and Clustering.
library(fields)
```

```
## Warning: package 'fields' was built under R version 4.2.2
```

```
## Loading required package: spam
```

```
## Warning: package 'spam' was built under R version 4.2.2
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following object is masked from 'package:Matrix':
##
##     det
##
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package: viridis
```

```
## Warning: package 'viridis' was built under R version 4.2.2
```

```
## Loading required package: viridisLite
##
## Try help(fields) to get started.
```

```
#Tools for Spatial Data
library(cluster)
#Methods for Cluster analysis.
```

Set working directory and load the data into workspace

```
getwd()
```

```
## [1] "C:/Users/Nirav Mahnot/Desktop"
```

```
setwd("C:/Users/Nirav Mahnot/Desktop/Nirav/Endterm")
getwd()
```

```
## [1] "C:/Users/Nirav Mahnot/Desktop/Nirav/Endterm"
```

```
player_data <- read.csv("finalData.csv", stringsAsFactors = TRUE)

#Creating a copy of the original dataset
player_save = player_data
```

Check and validate the type of the variables

```
str(player_data)
```

```
## 'data.frame':    1323 obs. of  29 variables:
##  $ name          : Factor w/ 1293 levels "Cengiz \xdcnder",..: 461 266 1179 1291 39 342 213 8
71 1016 107 ...
##  $ position      : Factor w/ 13 levels "Attacking Midfield",..: 4 12 3 5 4 5 4 3 9 9 ...
##  $ age           : int  21 22 21 22 23 30 22 22 27 30 ...
##  $ market_value  : num  150 35 40 60 55 40 30 70 70 70 ...
##  $ country_from  : Factor w/ 23 levels "Argentina","Austria",..: 11 16 9 15 17 19 19 13 9 9
...
##  $ league_from   : Factor w/ 5 levels "Bundesliga","LaLiga",..: 1 2 4 3 3 2 2 5 4 4 ...
##  $ club_from     : Factor w/ 322 levels "Borussia M\xf6nchengladbach",..: 65 37 180 45 257 23
0 231 168 189 182 ...
##  $ country_to    : Factor w/ 22 levels "Argentina","Austria",..: 8 8 8 18 8 8 8 10 8 10 ...
##  $ league_to     : Factor w/ 5 levels "Bundesliga","LaLiga",..: 4 4 4 2 4 4 4 1 4 1 ...
##  $ club_to       : Factor w/ 314 levels "\xdcmraniyespor",..: 182 183 89 223 177 183 193 55 8
9 55 ...
##  $ fee           : num  60 95 80.4 80 75 ...
##  $ loan          : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ games         : int  15 24 14 21 21 35 37 29 33 13 ...
##  $ time          : int  1061 1022 1213 1183 1183 3096 1474 2454 2678 1092 ...
##  $ goals         : int  13 7 1 3 3 4 9 4 20 1 ...
##  $ xG            : num  8.889 3.728 0.825 1.202 1.202 ...
##  $ assists       : int  2 1 0 3 3 3 1 0 1 2 ...
##  $ xA            : num  1.5265 2.4623 0.0361 1.738 1.738 ...
##  $ shots         : int  33 33 6 8 8 45 44 18 100 5 ...
##  $ key_passes    : int  9 18 2 12 12 14 14 3 48 20 ...
##  $ yellow_cards  : int  0 2 4 1 1 12 4 5 5 2 ...
##  $ red_cards     : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ position.1    : Factor w/ 15 levels "D","D F M S",..: 10 9 6 14 14 13 10 6 9 5 ...
##  $ team_name     : Factor w/ 94 levels "AC Milan","Alaves",..: 14 10 73 10 10 67 68 40 51 65
...
##  $ npg           : int  13 7 1 3 3 4 9 4 20 1 ...
##  $ npxG          : num  8.889 3.728 0.825 1.202 1.202 ...
##  $ xGChain       : num  11.366 8.803 0.914 7.646 7.646 ...
##  $ xGBuildup     : num  2.98 3.98 0.75 5.6 5.6 ...
##  $ League_from_to: Factor w/ 178 levels "Argentina to Argentina",..: 73 119 47 116 129 145 14
5 99 47 49 ...
```

#Reassigning values to reduce factors from 13 to 4

```
player_data <- player_data %>% mutate(position_new = case_when(position =="Centre-Forward" ~"Att
acker",
                                                               position=="Left Winger"~"Attacke
r",
                                                               position=="Right Winger"~"Attacke
r",
                                                               position=="Second Striker"~"Attac
ker",
                                                               position=="Goalkeeper"~"Goalkeepe
r",
                                                               position=="Central Midfield" ~ "M
idfielder",
                                                               position=="Attacking Midfield" ~
"Midfielder",
                                                               position=="Defensive Midfield" ~
"Midfielder",
                                                               position=="Left Midfield" ~ "Midf
ielder",
                                                               position=="Right Midfield" ~ "Mid
fielder",
                                                               position=="Right-Back" ~ "Defende
r",
                                                               position=="Centre-Back" ~ "Defend
er",
                                                               position=="Left-Back" ~ "Defende
r"))

#Centre-Forward,Left Winger, Right Winger, Second Striker reassigned to single factor - "Attacke
r"
#Central Midfield, Attacking Midfield, Defensive Midfield, Left Midfield, Right Midfield reassig
ned to - "Midfielder"
#Centre-Back, Right-Back, Left-Back reasigned to - "Defender"
#Goalkeeper remains the same.
```

Removing unwanted columns

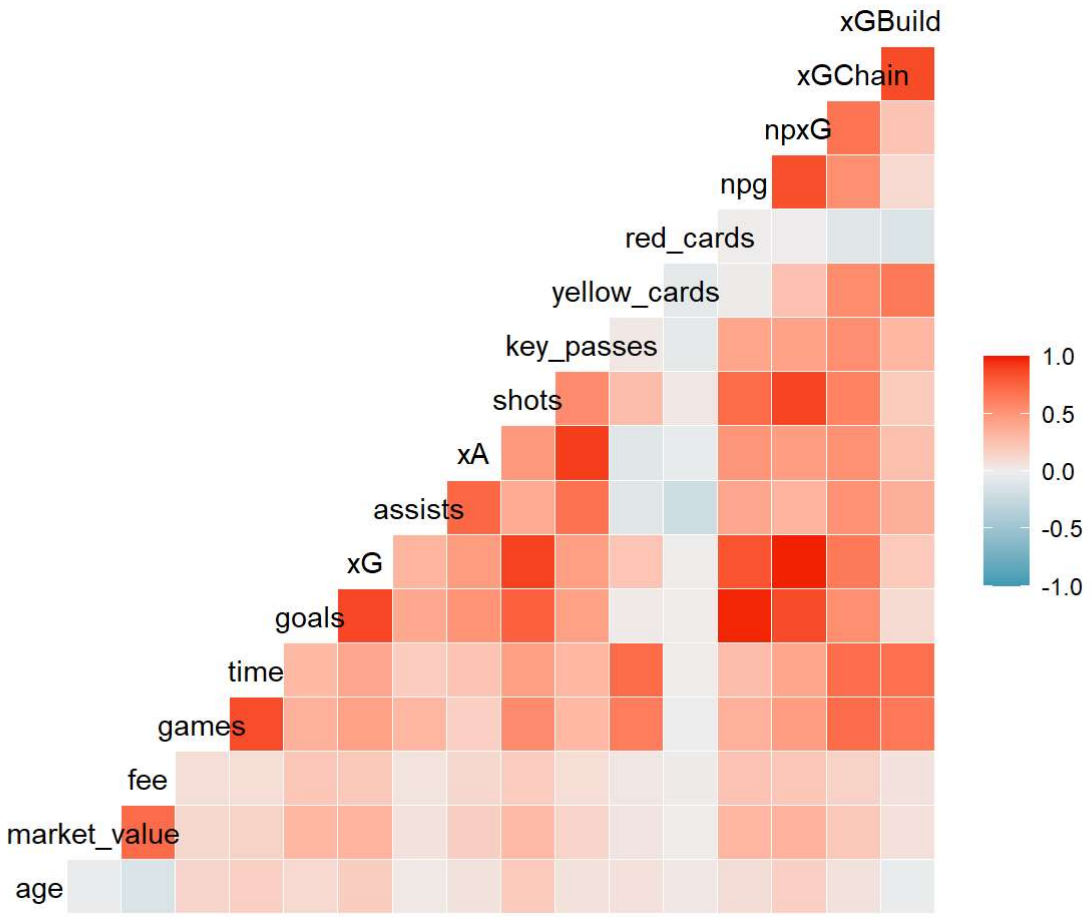
```
player_data$position_new <- as.factor(player_data$position_new)

player_data<- subset(player_data, select = -c(club_from,club_to,position.1,name,team_name))
str(player_data)
```

```
## 'data.frame':    1323 obs. of  25 variables:
##  $ position      : Factor w/ 13 levels "Attacking Midfield",..: 4 12 3 5 4 5 4 3 9 9 ...
##  $ age           : int  21 22 21 22 23 30 22 22 27 30 ...
##  $ market_value  : num  150 35 40 60 55 40 30 70 70 70 ...
##  $ country_from  : Factor w/ 23 levels "Argentina","Austria",..: 11 16 9 15 17 19 19 13 9 9
...
##  $ league_from   : Factor w/ 5 levels "Bundesliga","LaLiga",..: 1 2 4 3 3 2 2 5 4 4 ...
##  $ country_to    : Factor w/ 22 levels "Argentina","Austria",..: 8 8 8 18 8 8 8 10 8 10 ...
##  $ league_to     : Factor w/ 5 levels "Bundesliga","LaLiga",..: 4 4 4 2 4 4 4 1 4 1 ...
##  $ fee           : num  60 95 80.4 80 75 ...
##  $ loan          : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ games         : int  15 24 14 21 21 35 37 29 33 13 ...
##  $ time          : int  1061 1022 1213 1183 1183 3096 1474 2454 2678 1092 ...
##  $ goals         : int  13 7 1 3 3 4 9 4 20 1 ...
##  $ xG            : num  8.889 3.728 0.825 1.202 1.202 ...
##  $ assists       : int  2 1 0 3 3 3 1 0 1 2 ...
##  $ xA            : num  1.5265 2.4623 0.0361 1.738 1.738 ...
##  $ shots         : int  33 33 6 8 8 45 44 18 100 5 ...
##  $ key_passes    : int  9 18 2 12 12 14 14 3 48 20 ...
##  $ yellow_cards  : int  0 2 4 1 1 12 4 5 5 2 ...
##  $ red_cards     : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ npg           : int  13 7 1 3 3 4 9 4 20 1 ...
##  $ npxG          : num  8.889 3.728 0.825 1.202 1.202 ...
##  $ xGChain       : num  11.366 8.803 0.914 7.646 7.646 ...
##  $ xGBuildup     : num  2.98 3.98 0.75 5.6 5.6 ...
##  $ League_from_to: Factor w/ 178 levels "Argentina to Argentina",..: 73 119 47 116 129 145 14
5 99 47 49 ...
##  $ position_new  : Factor w/ 4 levels "Attacker","Defender",..: 1 1 2 4 1 4 1 2 1 1 ...
```

correlation Matrix

```
ggcorr(player_data)
```

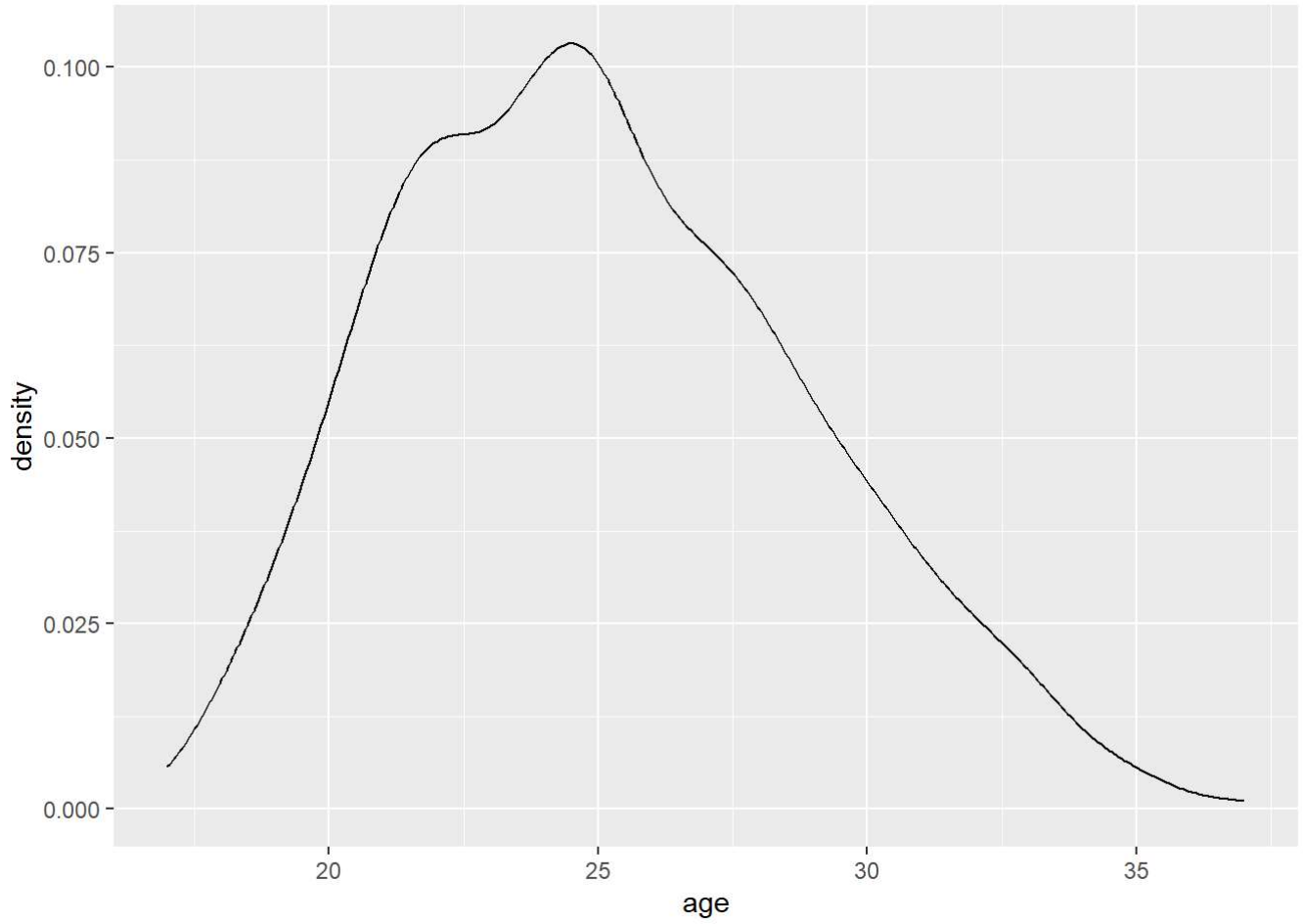
```
## Warning in ggcorr(player_data): data in column(s) 'position', 'country_from',
## 'league_from', 'country_to', 'league_to', 'loan', 'League_from_to',
## 'position_new' are not numeric and were ignored
```



Ages of football players

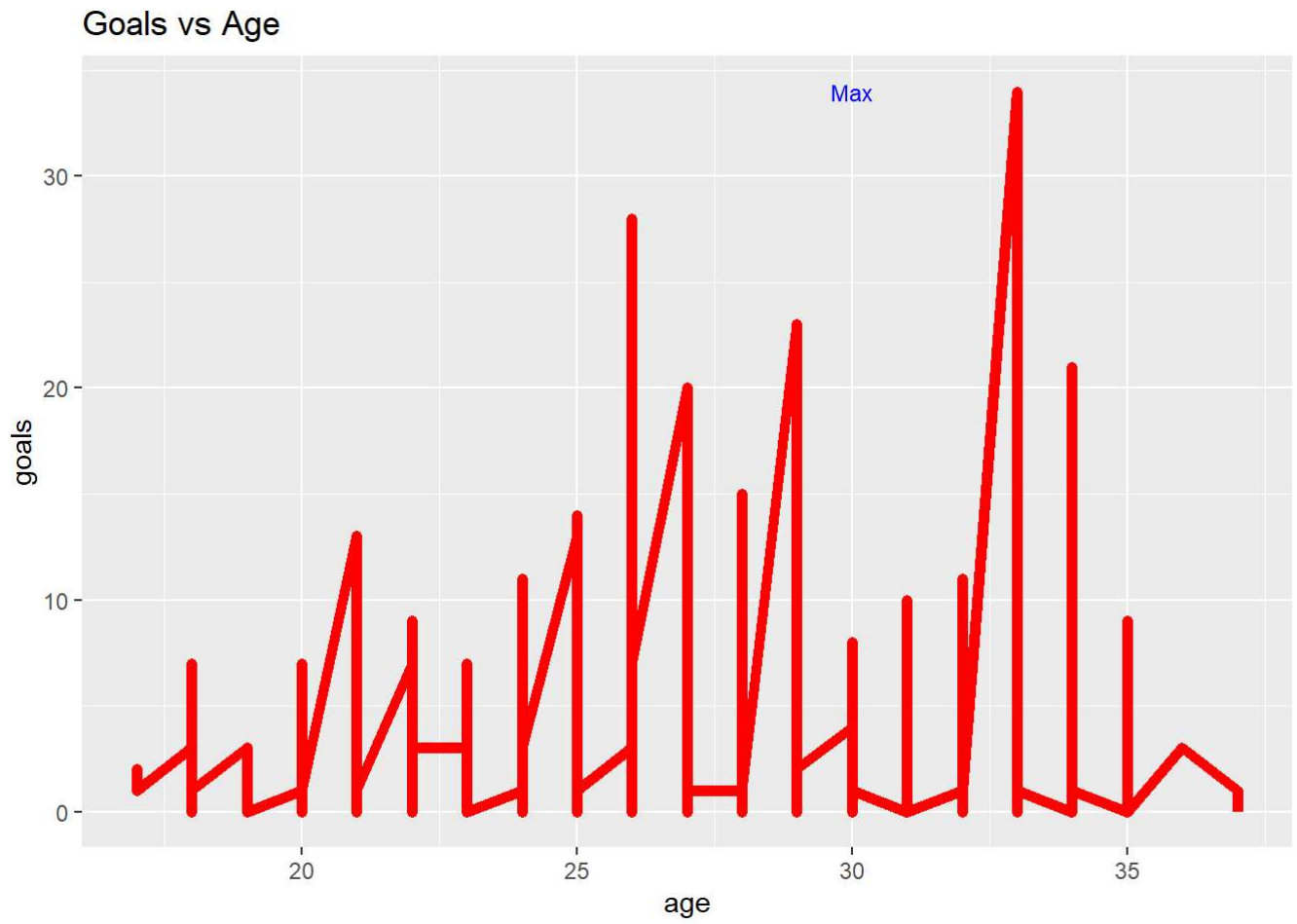
```
ggplot(player_data, aes(age, fill = age)) +
  geom_density(position = "stack")
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



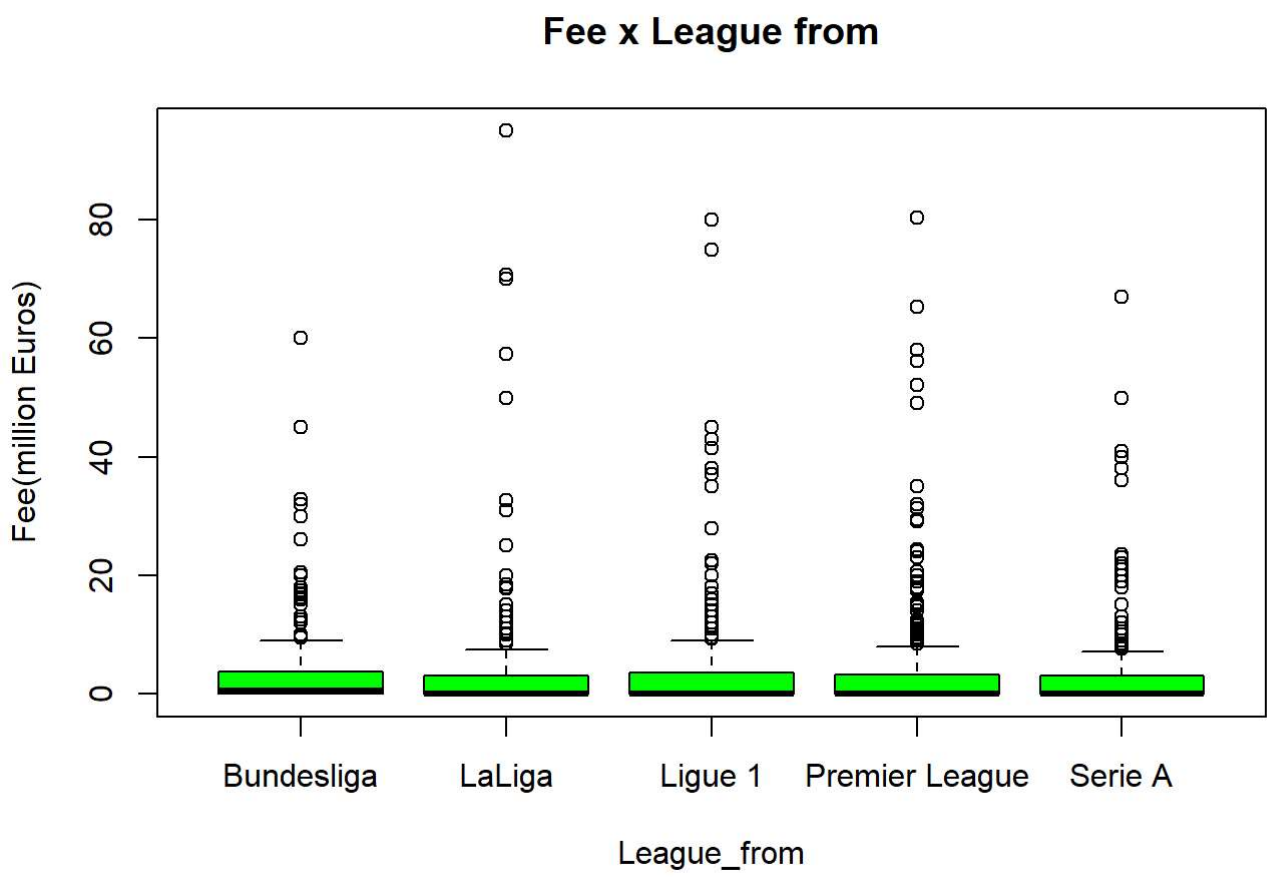
#Goals x Age

```
ggplot(data = player_data,aes(x=age,y=goals))+
  geom_line(color="red",linewidth=2)+labs(title="Goals vs Age")+
  annotate("text", x = 30, y = max(player_data$goals),color="blue", label = "Max", parse = TRUE,
size = 3)
```



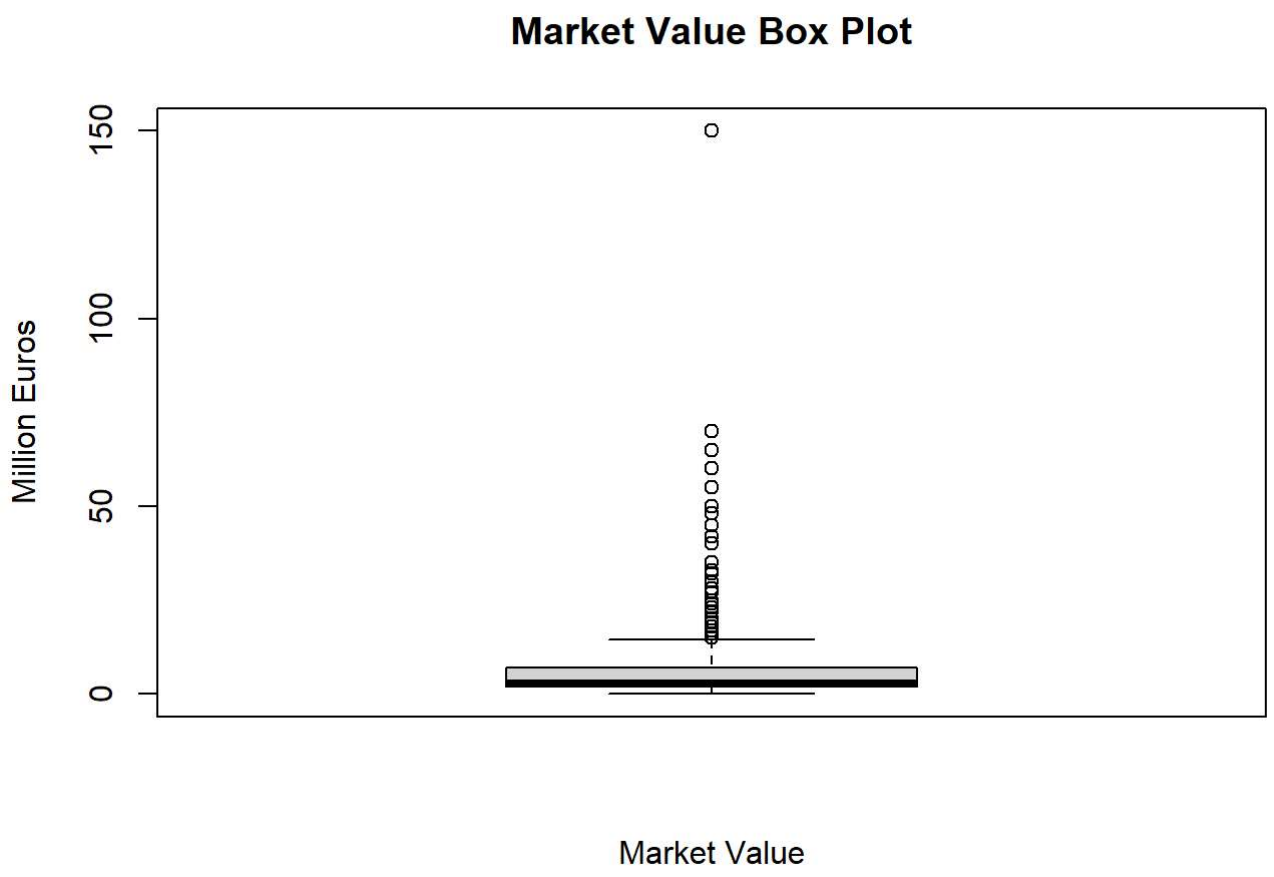
#Fee x Leagu_from

```
plot(player_data$league_from,player_data$fee, xlab = 'League_from', ylab = 'Fee(million Euros)',
main = 'Fee x League from', col = 'green')
```



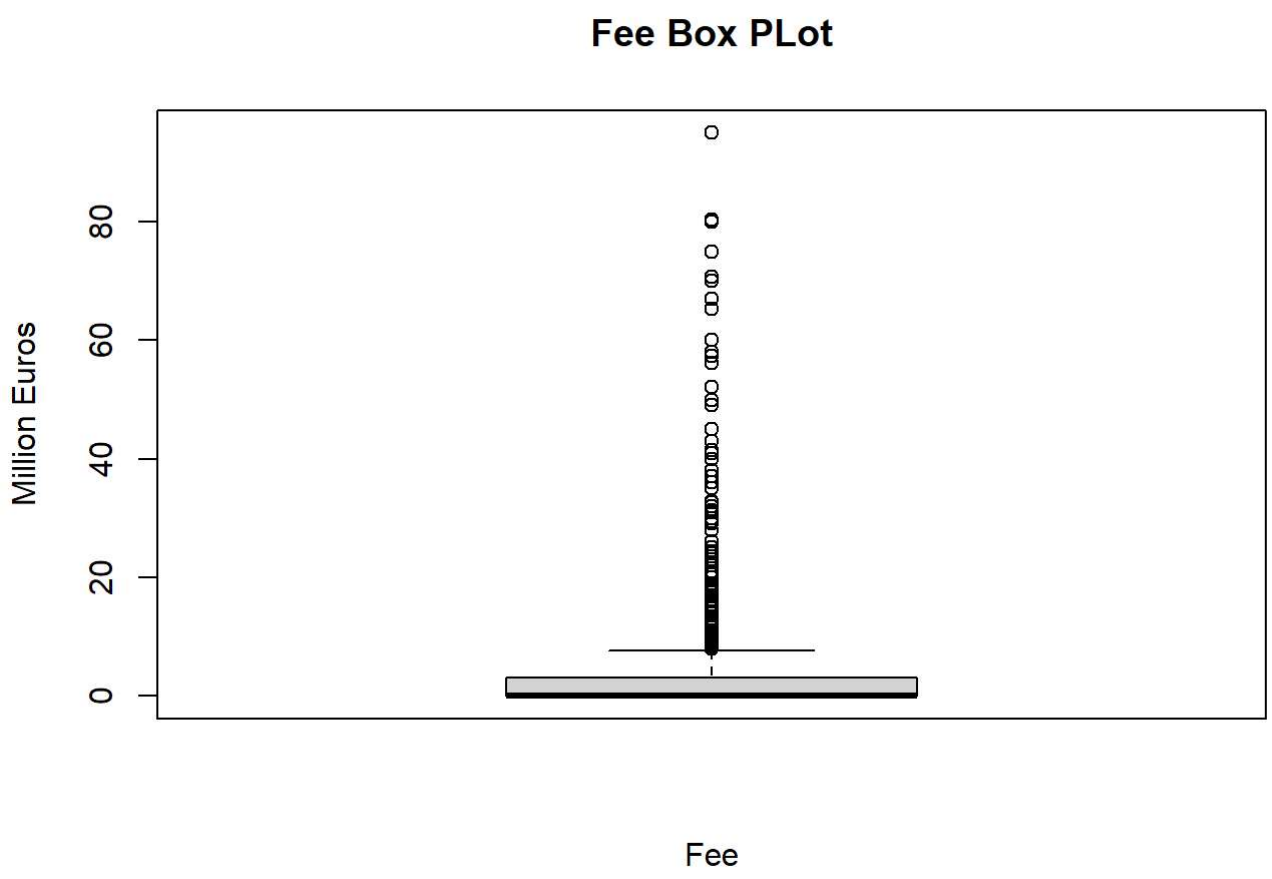
#Market Value Box Plot

```
boxplot(player_data$market_value,xlab = 'Market Value',ylab='Million Euros', main='Market Value Box Plot')
```



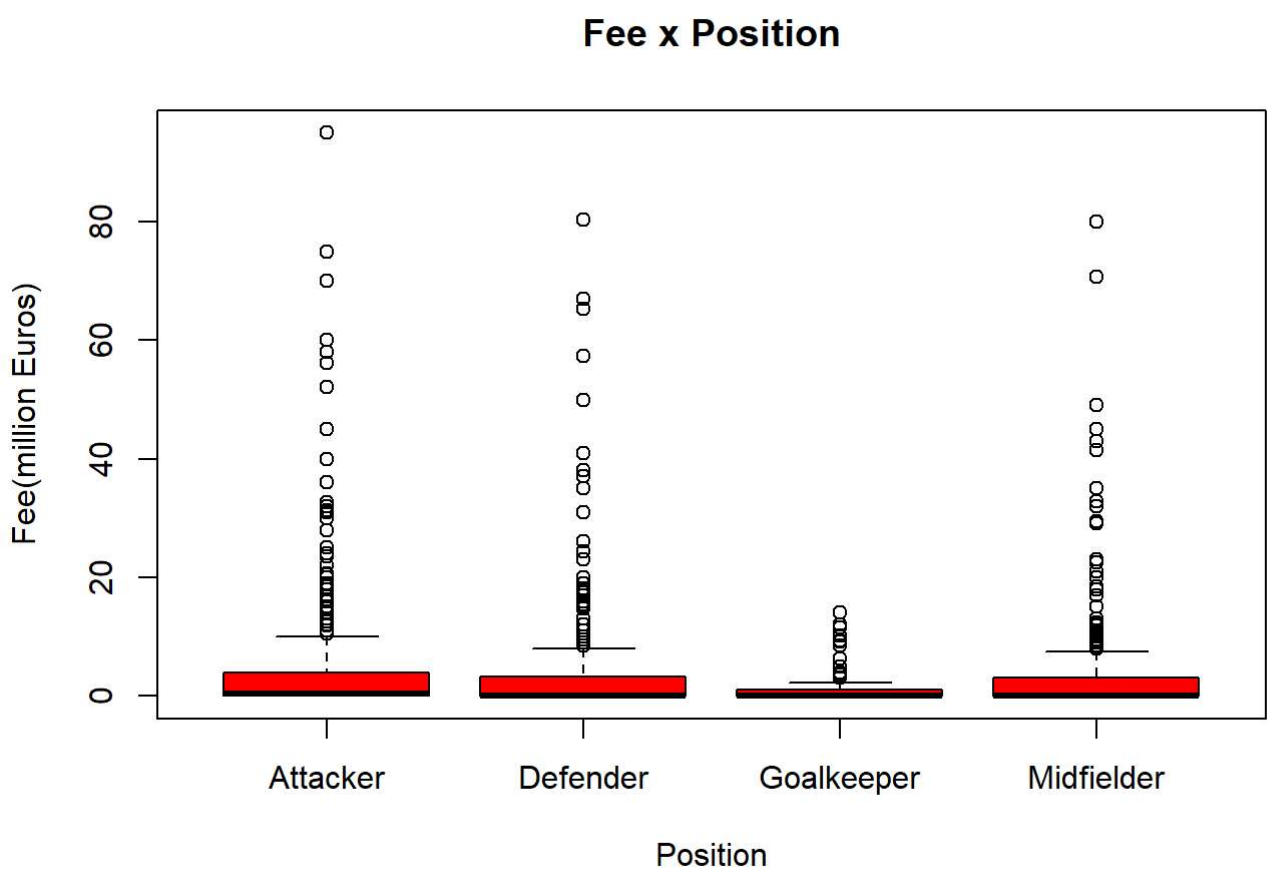
#Fee Box PLot

```
boxplot(player_data$fee,xlab = 'Fee',ylab='Million Euros', main='Fee Box PLot')
```



#Fee x Position

```
plot(player_data$position_new,player_data$fee, xlab = 'Position', ylab = 'Fee(million Euros)', m
ain = 'Fee x Position', col = 'red')
```



Train - Test - Valid Split (80-10-10)

```
#Train-Test split
set.seed(123)
train_sample <- createDataPartition(player_data$fee, p = 0.8, list = FALSE)
str(train_sample)
```

```
##  int [1:1059, 1] 2 3 5 8 9 10 11 13 14 15 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "Resample1"
```

```
train <- player_data[train_sample, ]
testdata  <- player_data[-train_sample, ]

#Test-valid split
set.seed(123)
train_sample1 <- createDataPartition(testdata$fee, p = 0.5, list = FALSE)
str(train_sample)
```

```
##  int [1:1059, 1] 2 3 5 8 9 10 11 13 14 15 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "Resample1"
```

```
valid <- testdata[train_sample1, ]
test  <- testdata[-train_sample1, ]
```

#SLR - Predicting fee model

```
model <- lm(fee~position_new+age+market_value+league_from+games+time+goals+assists+loan+xG+xA, d
ata =train)
summary(model)
```

```
##
## Call:
## lm(formula = fee ~ position_new + age + market_value + league_from +
##     games + time + goals + assists + loan + xG + xA, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.637  -1.967  -0.120   1.746  66.406
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.3042587  1.4253907   6.528 1.04e-10 ***
## position_newDefender      -0.1008374  0.4657070  -0.217   0.8286
## position_newGoalkeeper    -0.2679579  0.8143229  -0.329   0.7422
## position_newMidfielder    -0.5424150  0.4665400  -1.163   0.2452
## age                       -0.3460460  0.0498666  -6.939 6.91e-12 ***
## market_value               0.7296855  0.0213453  34.185  < 2e-16 ***
## league_fromLaLiga          0.5704206  0.6308225   0.904   0.3661
## league_fromLigue 1         0.6444602  0.5553357   1.160   0.2461
## league_fromPremier League  0.1604655  0.5750426   0.279   0.7803
## league_fromSerie A         0.5137439  0.5662030   0.907   0.3644
## games                      0.0179522  0.0505126   0.355   0.7224
## time                      -0.0002367  0.0005955  -0.398   0.6911
## goals                      0.3429117  0.1527513   2.245   0.0250 *
## assists                   -0.1775143  0.2331499  -0.761   0.4466
## loanTRUE                  -4.7933899  0.4003930 -11.972  < 2e-16 ***
## xG                        -0.3207237  0.1754193  -1.828   0.0678 .
## xA                        -0.1716015  0.2953330  -0.581   0.5613
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.792 on 1042 degrees of freedom
## Multiple R-squared:  0.6117, Adjusted R-squared:  0.6057
## F-statistic: 102.6 on 16 and 1042 DF,  p-value: < 2.2e-16
```

```
model0 <- lm(fee~position_new+age+market_value+league_from+games+time+goals+assists+xG+loan+shot
s+npg+npxG+xGChain+xGBuildup, data =train)
summary(model0)
```

```
##
## Call:
## lm(formula = fee ~ position_new + age + market_value + league_from +
##     games + time + goals + assists + xG + loan + shots + npg +
##     npxG + xGChain + xGBuildup, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.524  -2.005  -0.201   1.706  65.653
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                8.4592971  1.4224150   5.947 3.72e-09 ***
## position_newDefender      -0.0845459  0.4624068  -0.183  0.85496
## position_newGoalkeeper     0.0162920  0.8229229   0.020  0.98421
## position_newMidfielder    -0.4490312  0.4646038  -0.966  0.33403
## age                       -0.3161254  0.0507666  -6.227 6.89e-10 ***
## market_value               0.7251067  0.0212692  34.092  < 2e-16 ***
## league_fromLaLiga          0.6445908  0.6264846   1.029  0.30377
## league_fromLigue 1         0.6263453  0.5510258   1.137  0.25593
## league_fromPremier League  0.1528185  0.5718563   0.267  0.78934
## league_fromSerie A         0.6369834  0.5633201   1.131  0.25841
## games                      0.0210559  0.0471121   0.447  0.65502
## time                      -0.0007876  0.0005880  -1.340  0.18069
## goals                     -4.7265721  1.6187706  -2.920  0.00358 **
## assists                   -0.2705610  0.2079074  -1.301  0.19343
## xG                         4.3591244  1.9447287   2.242  0.02520 *
## loanTRUE                  -4.7220277  0.3968935 -11.897  < 2e-16 ***
## shots                     -0.0102589  0.0313589  -0.327  0.74362
## npg                        5.2403994  1.6396410   3.196  0.00144 **
## npxG                      -4.1752297  1.9987239  -2.089  0.03696 *
## xGChain                   -0.4139630  0.3828926  -1.081  0.27988
## xGBuildup                  0.5245529  0.3857928   1.360  0.17423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.743 on 1038 degrees of freedom
## Multiple R-squared:  0.6197, Adjusted R-squared:  0.6123
## F-statistic: 84.56 on 20 and 1038 DF,  p-value: < 2.2e-16
```

Best SLR model for predicting Fee

```
model1 <- lm(fee~position_new+age+market_value+league_from+games+assists+xG+loan+npg+npxG+goals+
key_passes, data =train)
summary(model1)
```

```
##
## Call:
## lm(formula = fee ~ position_new + age + market_value + league_from +
##     games + assists + xG + loan + npg + npxG + goals + key_passes,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.219  -1.986  -0.196   1.716  65.869
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                8.900199   1.383414   6.434  1.9e-10 ***
## position_newDefender      -0.082485   0.458827  -0.180  0.85737
## position_newGoalkeeper    -0.269824   0.795244  -0.339  0.73445
## position_newMidfielder    -0.457538   0.464041  -0.986  0.32437
## age                       -0.333753   0.049327  -6.766  2.2e-11 ***
## market_value               0.722693   0.021125  34.211  < 2e-16 ***
## league_fromLaLiga          0.604928   0.625542   0.967  0.33374
## league_fromLigue 1         0.645210   0.550935   1.171  0.24182
## league_fromPremier League  0.140896   0.570519   0.247  0.80499
## league_fromSerie A         0.593872   0.563290   1.054  0.29199
## games                     -0.002255   0.024199  -0.093  0.92577
## assists                   -0.284869   0.191446  -1.488  0.13706
## xG                         3.950224   1.931138   2.046  0.04105 *
## loanTRUE                  -4.759662   0.396167 -12.014  < 2e-16 ***
## npg                        4.958597   1.634993   3.033  0.00248 **
## npxG                      -4.165707   1.970192  -2.114  0.03472 *
## goals                     -4.531778   1.617766  -2.801  0.00518 **
## key_passes                -0.009517   0.028730  -0.331  0.74052
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.746 on 1041 degrees of freedom
## Multiple R-squared:  0.6182, Adjusted R-squared:  0.612
## F-statistic: 99.15 on 17 and 1041 DF,  p-value: < 2.2e-16
```

```
predictions1 <- predict(model1, valid)
summary(predictions1)
```

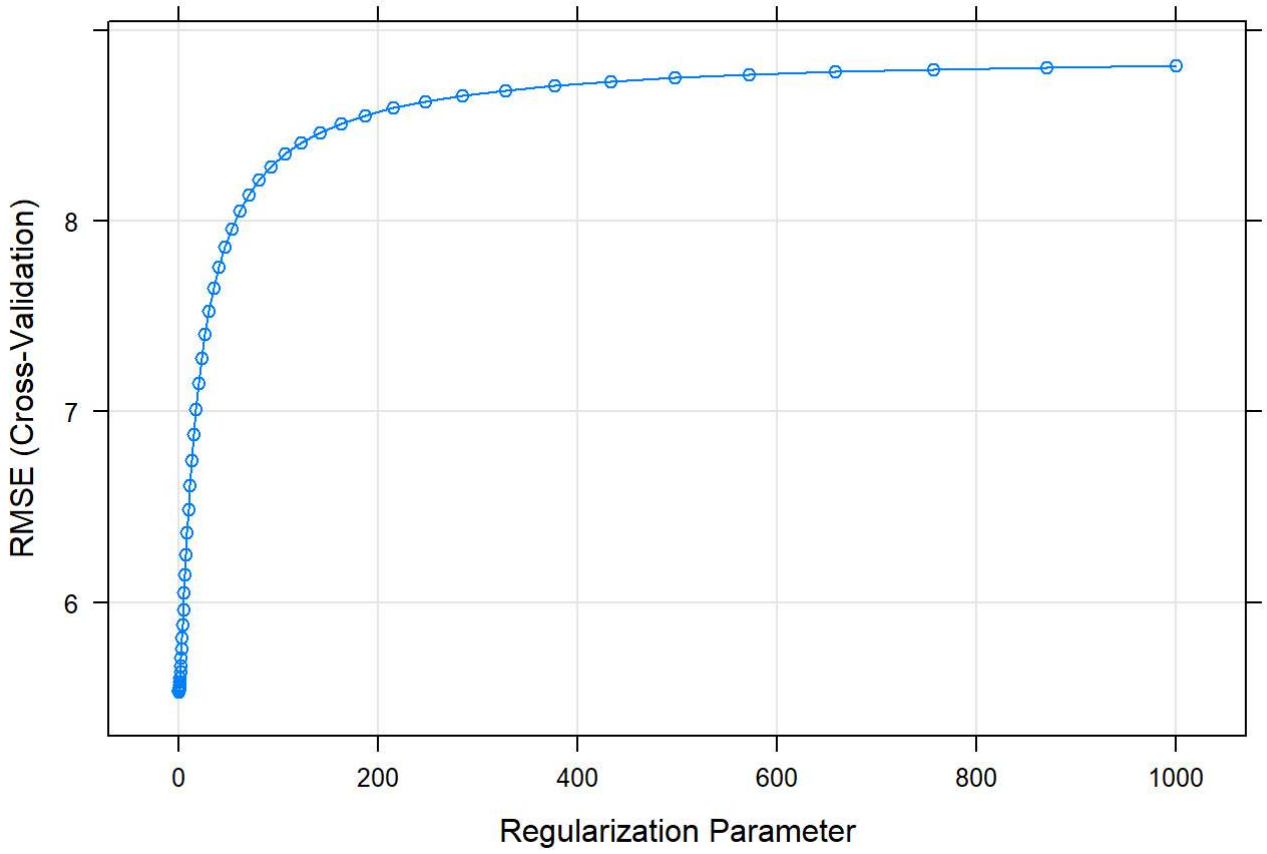
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.4546  0.2735  2.1375  3.6676  4.5933 37.0921
```

```
data.frame( R2 = R2(predictions1, valid$fee),
            RMSE = RMSE(predictions1, valid$fee),
            MAE = MAE(predictions1, valid$fee))
```

```
##          R2     RMSE      MAE
## 1 0.4746328 5.925423 3.194644
```

#Ridge model

```
lambda <- 10^seq(-3, 3, length = 100)
train_control <- trainControl(method = "cv", number = 10)
ridge_model_cv = train(fee~position_new+age+market_value+league_from+league_to+loan+games+time+goals+xG+assists+xA+shots+key_passes+yellow_cards+red_cards, data = train, method = "glmnet", trControl = train_control,tuneGrid = expand.grid(alpha = 0, lambda = lambda))
plot(ridge_model_cv)
```




```
summary(ridge_model_cv)
```

```
##             Length Class      Mode
## a0           100   -none-     numeric
## beta        2400   dgCMatrix  S4
## df           100   -none-     numeric
## dim            2   -none-     numeric
## lambda       100   -none-     numeric
## dev.ratio    100   -none-     numeric
## nulldev        1   -none-     numeric
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        24   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          0   -none-     list
```

```
varImp(ridge_model_cv, scaled=TRUE ,plot=TRUE, plotType="boxplot")
```

```
## glmnet variable importance
##
##   only 20 most important variables shown (out of 24)
##
##                             Overall
## loanTRUE                   100.0000
## league_toPremier League     84.5716
## red_cards                   29.6439
## league_fromPremier League   27.4407
## market_value                15.5998
## position_newMidfielder      14.1213
## league_fromLigue 1          13.0145
## league_toSerie A            10.2038
## league_fromLaLiga            9.6344
## league_fromSerie A           9.3314
## age                          7.4111
## goals                        6.5477
## position_newGoalkeeper       5.6762
## league_toLaLiga              4.5639
## yellow_cards                 2.9793
## xA                           2.3664
## league_toLigue 1             2.3279
## xG                           1.1625
## position_newDefender         0.9306
## assists                      0.6913
```

Final Ridge model for dependent variable “fee”

```
lambda <- 10^seq(-3, 3, length = 100)
ridge1 <- train(
  fee ~position_new+market_value+league_from+league_to+loan+games+time+assists+shots+key_passes+
red_cards, data =train, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneGrid = expand.grid(alpha = 0, lambda = lambda)
)
# Model coefficients
coef(ridge1$finalModel, ridge1$bestTune$lambda)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                      s1
## (Intercept)                0.2239975847
## position_newDefender      -0.0273689497
## position_newGoalkeeper    -1.1272364491
## position_newMidfielder    -0.7813312260
## market_value               0.6801082459
## league_fromLaLiga          0.1391225609
## league_fromLigue 1         0.5971141430
## league_fromPremier League -1.3257988858
## league_fromSerie A         0.2596504634
## league_toLaLiga            0.0867568871
## league_toLigue 1          -0.0264629692
## league_toPremier League    3.5878618204
## league_toSerie A           0.4291244036
## loanTRUE                  -3.7953729373
## games                      0.0341763788
## time                      -0.0005717452
## assists                    0.0254452674
## shots                     -0.0195818530
## key_passes                -0.0111602422
## red_cards                  1.0979955215
```

```
# Make predictions
predictions1 <- predict(ridge1, valid)
# Model prediction performance
data.frame(
  RMSE = RMSE(predictions1, valid$fee),
  Rsquare = R2(predictions1, valid$fee)
)
```

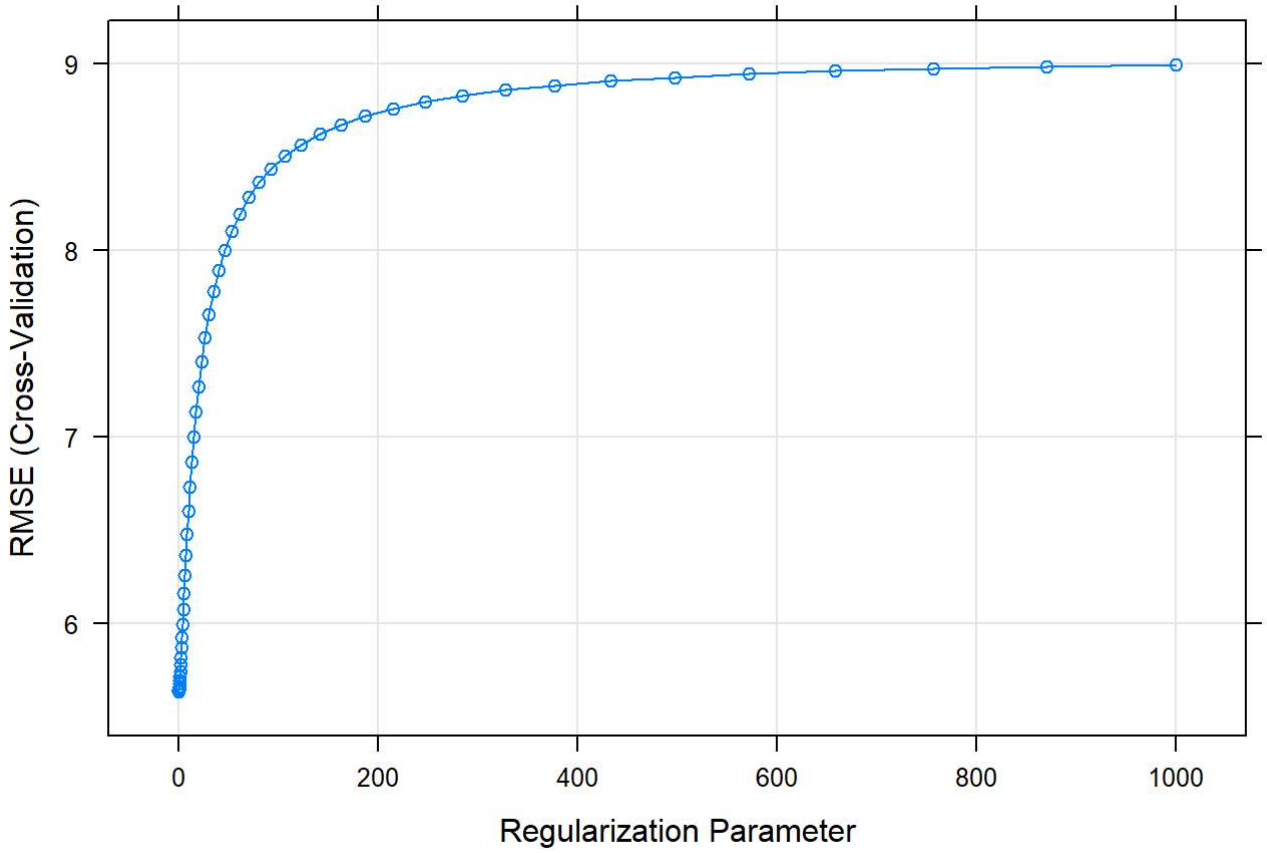
```
##       RMSE   Rsquare
## 1 5.759169 0.4977907
```

```
print(ridge1$bestTune$lambda)
```

```
## [1] 0.6135907
```

#Lasso model

```
lambda <- 10^seq(-3, 3, length = 100)
train_control <- trainControl(method = "cv", number = 10)
lasso_model_cv = train(fee~position_new+age+market_value+league_from+league_to+loan+games+time+goals+xG+assists+xA+shots+key_passes+yellow_cards+red_cards, data = train, method = "glmnet", trControl = train_control,tuneGrid = expand.grid(alpha = 0, lambda = lambda))
plot(lasso_model_cv)
```




```
summary(lasso_model_cv)
```

```
##             Length Class      Mode
## a0           100   -none-     numeric
## beta        2400   dgCMatrix  S4
## df           100   -none-     numeric
## dim            2   -none-     numeric
## lambda       100   -none-     numeric
## dev.ratio    100   -none-     numeric
## nulldev        1   -none-     numeric
## npasses        1   -none-     numeric
## jerr           1   -none-     numeric
## offset         1   -none-     logical
## call           5   -none-     call
## nobs           1   -none-     numeric
## lambdaOpt      1   -none-     numeric
## xNames        24   -none-     character
## problemType    1   -none-     character
## tuneValue      2   data.frame list
## obsLevels      1   -none-     logical
## param          0   -none-     list
```

```
varImp(lasso_model_cv, scaled=TRUE ,plot=TRUE, plotType="boxplot")
```

```
## glmnet variable importance
##
##   only 20 most important variables shown (out of 24)
##
##                             Overall
## loanTRUE                   100.0000
## league_toPremier League     84.5716
## red_cards                   29.6439
## league_fromPremier League   27.4407
## market_value                15.5998
## position_newMidfielder      14.1213
## league_fromLigue 1          13.0145
## league_toSerie A            10.2038
## league_fromLaLiga            9.6344
## league_fromSerie A           9.3314
## age                          7.4111
## goals                        6.5477
## position_newGoalkeeper       5.6762
## league_toLaLiga              4.5639
## yellow_cards                 2.9793
## xA                           2.3664
## league_toLigue 1             2.3279
## xG                           1.1625
## position_newDefender         0.9306
## assists                      0.6913
```

Final LASSO model for dependent variable “fee”

```
lambda <- 10^seq(-3, 3, length = 100)
lasso1 <- train(
  fee ~position_new+market_value+league_from+league_to+loan+games+time+assists+shots+key_passes+
red_cards, data =train, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneGrid = expand.grid(alpha = 1, lambda = lambda)
)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
# Model coefficients
coef(lasso1$finalModel, lasso1$bestTune$lambda)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                1.3693058445
## position_newDefender       .
## position_newGoalkeeper    -0.6247867409
## position_newMidfielder    -0.5059250717
## market_value               0.7195343368
## league_fromLaLiga          .
## league_fromLigue 1         0.2054013525
## league_fromPremier League -1.4288139225
## league_fromSerie A         .
## league_toLaLiga            .
## league_toLigue 1           .
## league_toPremier League    3.3308226741
## league_toSerie A           0.1792222751
## loanTRUE                  -3.7996738896
## games                      .
## time                      -0.0002501147
## assists                    .
## shots                     -0.0178834335
## key_passes                -0.0046891334
## red_cards                  0.8209089754
```

```
# Make predictions
predictions1 <- predict(lasso1, valid)
# Model prediction performance
data.frame(
  RMSE = RMSE(predictions1, valid$fee),
  Rsquare = R2(predictions1, valid$fee)
)
```

```
##      RMSE   Rsquare
## 1 5.76137 0.5003029
```

```
print(lasso1$bestTune$lambda)
```

```
## [1] 0.1
```

Best regression model on Test data

```
predictions2 <- predict(lasso1, test)
summary(predictions2)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  -4.8926  -0.1588   1.5615   5.1031   5.1822 110.7326
```

```
data.frame( R2 = R2(predictions2, test$fee),
            RMSE = RMSE(predictions2, test$fee),
            MAE = MAE(predictions2, test$fee))
```

```
##         R2     RMSE      MAE
## 1 0.498868 9.517463 4.591553
```

#Regression Tree

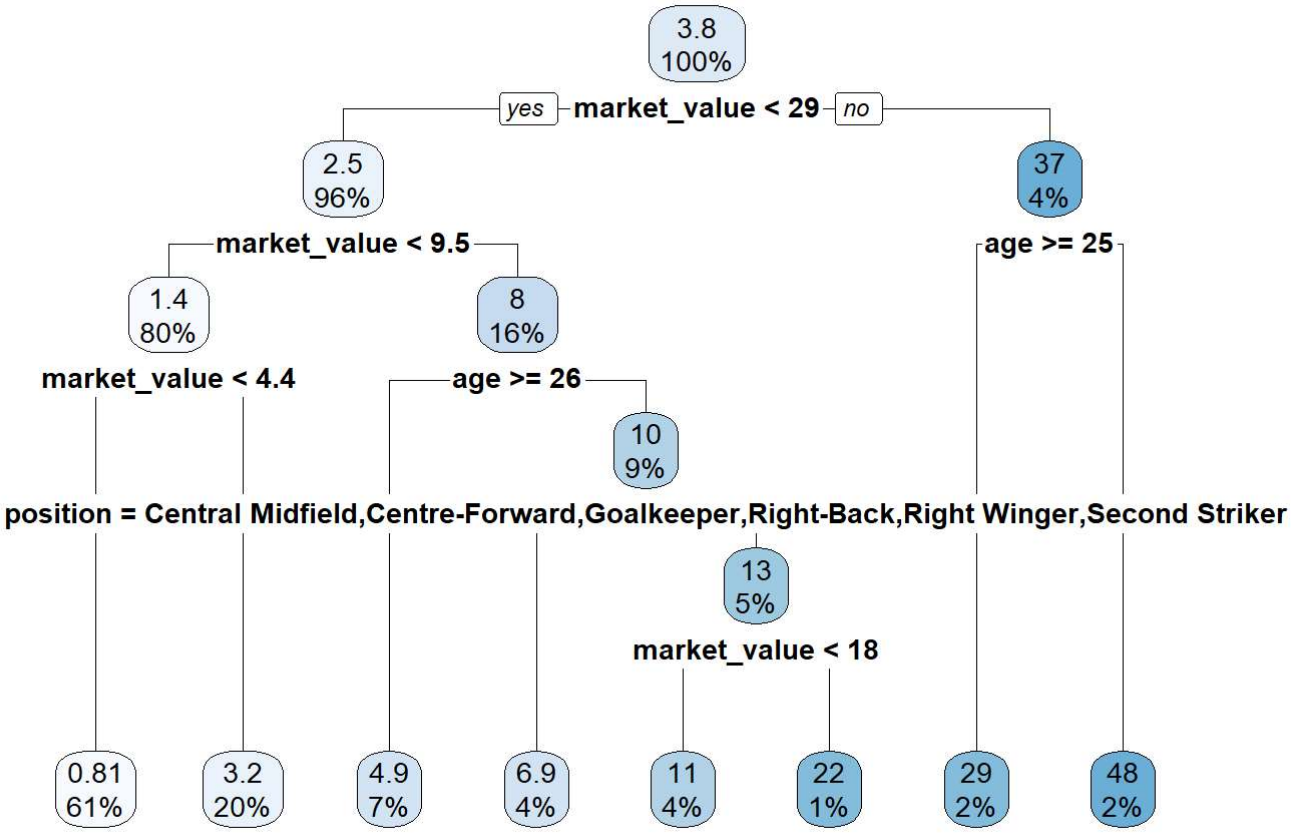
```
regr_tree <- rpart(fee~position+age+market_value+league_from+games+time+goals+assists,data = tra
in,method = "anova", minsplit= 25,xval=0)
summary(regr_tree)
```

```
## Call:
## rpart(formula = fee ~ position + age + market_value + league_from +
##     games + time + goals + assists, data = train, method = "anova",
##     minsplit = 25, xval = 0)
##   n= 1059
##
##           CP nsplit rel error
## 1 0.49202723      0 1.0000000
## 2 0.06912069      1 0.5079728
## 3 0.03932055      2 0.4388521
## 4 0.01344004      3 0.3995315
## 5 0.01150636      4 0.3860915
## 6 0.01002632      6 0.3630788
## 7 0.01000000      7 0.3530524
##
## Variable importance
## market_value          age        goals     position         time        games
##           79            7            6            2            2            2
##  league_from
##            2
##
## Node number 1: 1059 observations,    complexity param=0.4920272
##   mean=3.768752, MSE=85.01278
##   left son=2 (1020 obs) right son=3 (39 obs)
##   Primary splits:
##       market_value < 29     to the left,  improve=0.49202720, (0 missing)
##       goals        < 12.5   to the left,  improve=0.05487311, (0 missing)
##       age          < 26.5   to the right, improve=0.02115502, (0 missing)
##       position     splits as  LLRRRLLLRLLRR, improve=0.01098311, (0 missing)
##       time         < 2017.5 to the left,  improve=0.00910794, (0 missing)
##   Surrogate splits:
##       goals < 22.5   to the left,  agree=0.966, adj=0.077, (0 split)
##
## Node number 2: 1020 observations,    complexity param=0.06912069
##   mean=2.504106, MSE=26.02717
##   left son=4 (848 obs) right son=5 (172 obs)
##   Primary splits:
##       market_value < 9.5    to the left,  improve=0.234401900, (0 missing)
##       age          < 26.5   to the right, improve=0.033337040, (0 missing)
##       position     splits as  LLLLRLRRRLLLR, improve=0.009528188, (0 missing)
##       goals        < 2.5    to the left,  improve=0.008664692, (0 missing)
##       league_from  splits as  RLRRL, improve=0.004042149, (0 missing)
##   Surrogate splits:
##       games    < 36.5   to the left,  agree=0.838, adj=0.041, (0 split)
##       goals    < 7.5    to the left,  agree=0.835, adj=0.023, (0 split)
##       position splits as  LLLLLLLLLLLLR, agree=0.833, adj=0.012, (0 split)
##       time     < 3122.5 to the left,  agree=0.833, adj=0.012, (0 split)
##
## Node number 3: 39 observations,    complexity param=0.03932055
##   mean=36.8441, MSE=491.9059
##   left son=6 (23 obs) right son=7 (16 obs)
##   Primary splits:
```

```
##       age          < 24.5   to the right, improve=0.18452410, (0 missing)
##       league_from  splits as  LRRRL, improve=0.15033190, (0 missing)
##       position     splits as  RLRRR-L-R--RL, improve=0.11424020, (0 missing)
##       market_value < 37.5   to the left,  improve=0.08141370, (0 missing)
##       games        < 13.5   to the left,  improve=0.07619449, (0 missing)
##   Surrogate splits:
##       league_from  splits as  LLRLL, agree=0.718, adj=0.312, (0 split)
##       market_value < 34     to the right, agree=0.692, adj=0.250, (0 split)
##       time         < 1241.5 to the right, agree=0.692, adj=0.250, (0 split)
##       games        < 17.5   to the right, agree=0.667, adj=0.187, (0 split)
##       position     splits as  LLLLL-L-L--RL, agree=0.641, adj=0.125, (0 split)
##
## Node number 4: 848 observations,    complexity param=0.01002632
##   mean=1.391708, MSE=7.741424
##   left son=8 (641 obs) right son=9 (207 obs)
##   Primary splits:
##       market_value < 4.4    to the left,  improve=0.137500700, (0 missing)
##       age          < 23.5   to the right, improve=0.052447380, (0 missing)
##       position     splits as  LLRRLLRLLRLRR, improve=0.008597737, (0 missing)
##       league_from  splits as  RLRRL, improve=0.006333542, (0 missing)
##       time         < 2332.5 to the right, improve=0.003348558, (0 missing)
##
## Node number 5: 172 observations,    complexity param=0.01344004
##   mean=7.988488, MSE=80.00087
##   left son=10 (74 obs) right son=11 (98 obs)
##   Primary splits:
##       age          < 25.5   to the right, improve=0.08793412, (0 missing)
##       position     splits as  RLRLRLRLRL-LR, improve=0.05781631, (0 missing)
##       market_value < 22.5   to the left,  improve=0.03882381, (0 missing)
##       time         < 1583.5 to the right, improve=0.02924128, (0 missing)
##       league_from  splits as  RLRLL, improve=0.02496627, (0 missing)
##   Surrogate splits:
##       games    < 25.5   to the right, agree=0.669, adj=0.230, (0 split)
##       time     < 1370   to the right, agree=0.657, adj=0.203, (0 split)
##       goals    < 3.5    to the right, agree=0.628, adj=0.135, (0 split)
##       position splits as  RRRRRLRLRR-RL, agree=0.610, adj=0.095, (0 split)
##       assists  < 0.5    to the left,  agree=0.581, adj=0.027, (0 split)
##
## Node number 6: 23 observations
##   mean=28.89783, MSE=389.5953
##
## Node number 7: 16 observations
##   mean=48.26687, MSE=417.7292
##
## Node number 8: 641 observations
##   mean=0.8054087, MSE=3.447352
##
## Node number 9: 207 observations
##   mean=3.207251, MSE=16.67788
##
## Node number 10: 74 observations
##   mean=4.936216, MSE=49.68002
```

```
##
## Node number 11: 98 observations,    complexity param=0.01150636
##   mean=10.29327, MSE=90.54941
##   left son=22 (46 obs) right son=23 (52 obs)
##   Primary splits:
##       position     splits as  RLRLRLR-RL-LL, improve=0.11248760, (0 missing)
##       time         < 1777   to the right, improve=0.05942523, (0 missing)
##       market_value < 18.5   to the left,  improve=0.05740730, (0 missing)
##       games        < 21.5   to the right, improve=0.02732189, (0 missing)
##       league_from  splits as  RLRLL, improve=0.02362988, (0 missing)
##   Surrogate splits:
##       league_from  splits as  RLRRR,      agree=0.582, adj=0.109, (0 split)
##       time         < 469    to the left,  agree=0.582, adj=0.109, (0 split)
##       market_value < 17.5   to the right, agree=0.571, adj=0.087, (0 split)
##       assists      < 4.5    to the right, agree=0.561, adj=0.065, (0 split)
##       games        < 23.5   to the right, agree=0.551, adj=0.043, (0 split)
##
## Node number 22: 46 observations
##   mean=6.9, MSE=40.55565
##
## Node number 23: 52 observations,    complexity param=0.01150636
##   mean=13.295, MSE=115.5786
##   left son=46 (40 obs) right son=47 (12 obs)
##   Primary splits:
##       market_value < 17.5   to the left,  improve=0.17863390, (0 missing)
##       time         < 1429.5 to the right, improve=0.08623408, (0 missing)
##       league_from  splits as  LRRRL, improve=0.04869969, (0 missing)
##       games        < 21.5   to the right, improve=0.04680367, (0 missing)
##       position     splits as  L-L-L-L-R----, improve=0.02744474, (0 missing)
##
## Node number 46: 40 observations
##   mean=10.80625, MSE=84.46052
##
## Node number 47: 12 observations
##   mean=21.59083, MSE=129.8383
```

```
rpart.plot(regr_tree)
```




```
rt_pred <- predict(regr_tree,valid)
summary(rt_pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8054  0.8054  0.8054  2.9780  3.2073 48.2669
```

```
data.frame( R2 = R2(rt_pred, valid$fee),
            RMSE = RMSE(rt_pred, valid$fee),
            MAE = MAE(rt_pred, valid$fee))
```

```
##          R2     RMSE     MAE
## 1 0.5927833 5.217431 2.61554
```

#Regression Tree without market value

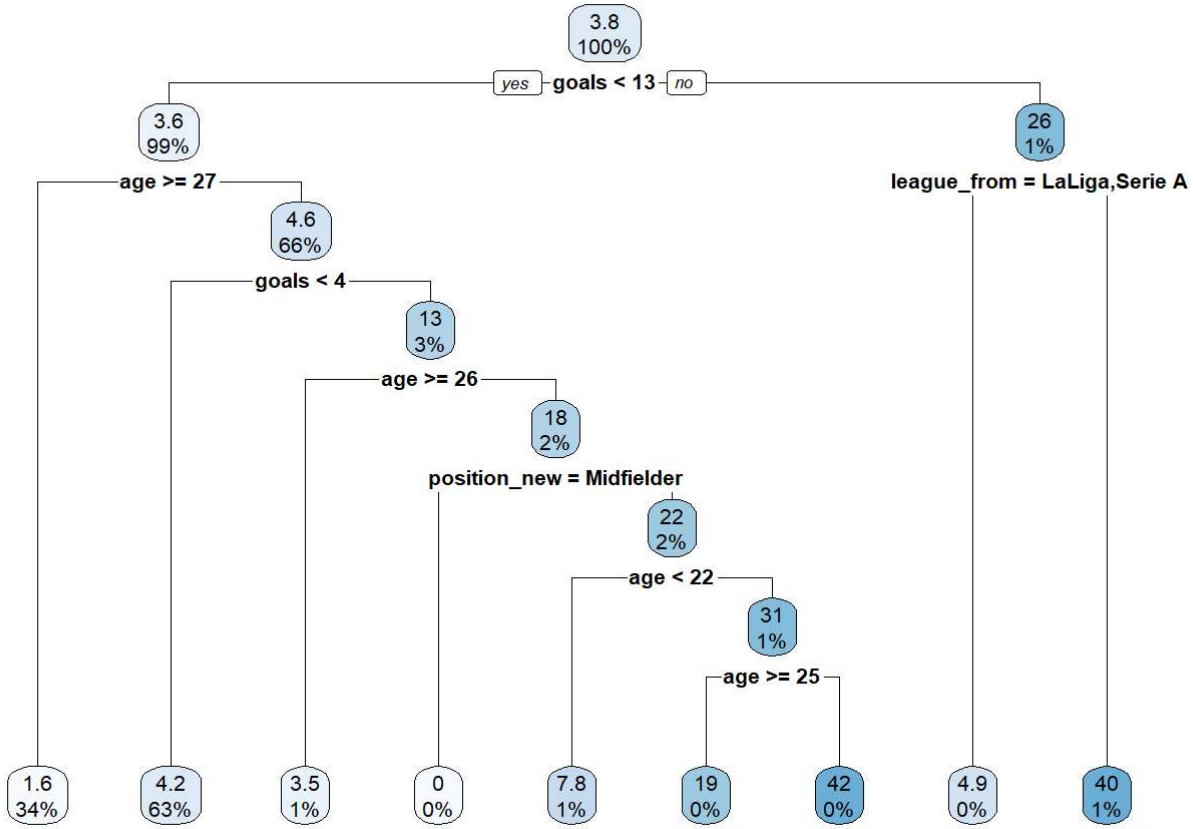
```
regr_tree2 <- rpart(fee~position_new+age+league_from+games+time+goals+assists,data = train,metho
d = "anova", minsplit= 10,xval=0)
summary(regr_tree2)
```

```
## Call:
## rpart(formula = fee ~ position_new + age + league_from + games +
##     time + goals + assists, data = train, method = "anova", minsplit = 10,
##     xval = 0)
##   n= 1059
##
##           CP nsplit rel error
## 1 0.05487311      0 1.0000000
## 2 0.03270298      1 0.9451269
## 3 0.02403335      2 0.9124239
## 4 0.01837474      4 0.8643572
## 5 0.01463874      7 0.8092330
## 6 0.01000000      8 0.7945943
##
## Variable importance
##          age        goals  league_from position_new         time        games
##           31           30           17           10            6            4
##      assists
##            2
##
## Node number 1: 1059 observations,    complexity param=0.05487311
##   mean=3.768752, MSE=85.01278
##   left son=2 (1049 obs) right son=3 (10 obs)
##   Primary splits:
##       goals   < 12.5   to the left,  improve=0.054873110, (0 missing)
##       age     < 26.5   to the right, improve=0.021155020, (0 missing)
##       time    < 2017.5 to the left,  improve=0.009107940, (0 missing)
##       assists < 5.5    to the left,  improve=0.007246809, (0 missing)
##       games   < 21.5   to the left,  improve=0.006880539, (0 missing)
##
## Node number 2: 1049 observations,    complexity param=0.02403335
##   mean=3.557872, MSE=76.15136
##   left son=4 (355 obs) right son=5 (694 obs)
##   Primary splits:
##       age          < 26.5   to the right, improve=0.026198360, (0 missing)
##       goals        < 3.5    to the left,  improve=0.010435950, (0 missing)
##       time         < 949    to the left,  improve=0.005028748, (0 missing)
##       games        < 20.5   to the left,  improve=0.004080949, (0 missing)
##       position_new splits as  RRLR,       improve=0.003213314, (0 missing)
##   Surrogate splits:
##       time         < 1581   to the right, agree=0.675, adj=0.039, (0 split)
##       goals        < 3.5    to the right, agree=0.672, adj=0.031, (0 split)
##       assists      < 3.5    to the right, agree=0.671, adj=0.028, (0 split)
##       games        < 35.5   to the right, agree=0.670, adj=0.025, (0 split)
##       position_new splits as  RRLR,       agree=0.666, adj=0.014, (0 split)
##
## Node number 3: 10 observations,    complexity param=0.03270298
##   mean=25.89, MSE=520.5609
##   left son=6 (4 obs) right son=7 (6 obs)
##   Primary splits:
##       league_from splits as  RL-RL,      improve=0.56558250, (0 missing)
##       age         < 27.5   to the right, improve=0.54928460, (0 missing)
```

```
##       time        < 2769   to the right, improve=0.19882600, (0 missing)
##       goals       < 20.5   to the right, improve=0.12885350, (0 missing)
##       games       < 35     to the right, improve=0.03895186, (0 missing)
##   Surrogate splits:
##       age          < 27.5   to the right, agree=0.8, adj=0.50, (0 split)
##       position_new splits as  R--L,       agree=0.7, adj=0.25, (0 split)
##       goals        < 22.5   to the left,  agree=0.7, adj=0.25, (0 split)
##
## Node number 4: 355 observations
##   mean=1.582989, MSE=20.95044
##
## Node number 5: 694 observations,    complexity param=0.02403335
##   mean=4.568079, MSE=101.3726
##   left son=10 (664 obs) right son=11 (30 obs)
##   Primary splits:
##       goals        < 3.5    to the left,  improve=0.031762540, (0 missing)
##       games        < 13.5   to the left,  improve=0.006517065, (0 missing)
##       time         < 944.5  to the left,  improve=0.005552862, (0 missing)
##       league_from  splits as  LRRRL,      improve=0.004792270, (0 missing)
##       position_new splits as  RRLR,       improve=0.003863660, (0 missing)
##   Surrogate splits:
##       games < 36.5   to the left,  agree=0.96, adj=0.067, (0 split)
##
## Node number 6: 4 observations
##   mean=4.875, MSE=19.29688
##
## Node number 7: 6 observations
##   mean=39.9, MSE=364.0367
##
## Node number 10: 664 observations
##   mean=4.186667, MSE=79.8999
##
## Node number 11: 30 observations,    complexity param=0.01837474
##   mean=13.01, MSE=502.1489
##   left son=22 (10 obs) right son=23 (20 obs)
##   Primary splits:
##       age          < 25.5   to the right, improve=0.09081221, (0 missing)
##       position_new splits as  RR-L,       improve=0.06595658, (0 missing)
##       time         < 2443   to the left,  improve=0.06576635, (0 missing)
##       league_from  splits as  LRLRR,      improve=0.05918121, (0 missing)
##       games        < 28.5   to the left,  improve=0.01876162, (0 missing)
##   Surrogate splits:
##       games       < 36     to the right, agree=0.733, adj=0.2, (0 split)
##       time        < 2725   to the right, agree=0.733, adj=0.2, (0 split)
##       league_from splits as  RRRLR,      agree=0.700, adj=0.1, (0 split)
##       goals       < 7.5    to the right, agree=0.700, adj=0.1, (0 split)
##
## Node number 22: 10 observations
##   mean=3.46, MSE=21.7064
##
## Node number 23: 20 observations,    complexity param=0.01837474
##   mean=17.785, MSE=673.9683
```

```
##   left son=46 (4 obs) right son=47 (16 obs)
##   Primary splits:
##       position_new splits as  RR-L,       improve=0.11732980, (0 missing)
##       time         < 2388.5 to the left,  improve=0.08372379, (0 missing)
##       league_from  splits as  LRLRR,      improve=0.06573523, (0 missing)
##       age          < 21.5   to the left,  improve=0.06403516, (0 missing)
##       goals        < 5.5    to the right, improve=0.05148822, (0 missing)
##
## Node number 46: 4 observations
##   mean=0, MSE=0
##
## Node number 47: 16 observations,    complexity param=0.01837474
##   mean=22.23125, MSE=743.6146
##   left son=94 (6 obs) right son=95 (10 obs)
##   Primary splits:
##       age         < 21.5   to the left,  improve=0.16920590, (0 missing)
##       league_from splits as  LRLRL,      improve=0.11995770, (0 missing)
##       time        < 1623   to the left,  improve=0.09200894, (0 missing)
##       games       < 28.5   to the left,  improve=0.07554316, (0 missing)
##       assists     < 2      to the left,  improve=0.01722717, (0 missing)
##   Surrogate splits:
##       league_from splits as  LRLRR,      agree=0.75, adj=0.333, (0 split)
##       time        < 1171   to the left,  agree=0.75, adj=0.333, (0 split)
##
## Node number 94: 6 observations
##   mean=7.75, MSE=42.72917
##
## Node number 95: 10 observations,    complexity param=0.01463874
##   mean=30.92, MSE=962.8276
##   left son=190 (5 obs) right son=191 (5 obs)
##   Primary splits:
##       age         < 24.5   to the right, improve=0.13687850, (0 missing)
##       league_from splits as  LRLRL,      improve=0.11264740, (0 missing)
##       time        < 1623   to the left,  improve=0.07750533, (0 missing)
##       games       < 23     to the left,  improve=0.06917128, (0 missing)
##       goals       < 5.5    to the right, improve=0.02098335, (0 missing)
##   Surrogate splits:
##       league_from  splits as  LRRLR,      agree=0.8, adj=0.6, (0 split)
##       games        < 23     to the left,  agree=0.7, adj=0.4, (0 split)
##       time         < 2136   to the left,  agree=0.7, adj=0.4, (0 split)
##       assists      < 2      to the right, agree=0.7, adj=0.4, (0 split)
##       position_new splits as  LR--,       agree=0.6, adj=0.2, (0 split)
##
## Node number 190: 5 observations
##   mean=19.44, MSE=417.0344
##
## Node number 191: 5 observations
##   mean=42.4, MSE=1245.04
```

```
rpart.plot(regr_tree2)
```




```
rt_pred2 <- predict(regr_tree2,valid)
summary(rt_pred2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.583   4.187   4.569   4.187  42.400
```

```
data.frame( R2 = R2(rt_pred2, valid$fee),
            RMSE = RMSE(rt_pred2, valid$fee),
            MAE = MAE(rt_pred2, valid$fee))
```

```
##            R2     RMSE      MAE
## 1 0.001440209 10.77716 5.625939
```

#Train-Test split (player_data$goals)

```
set.seed(123)
train_sample <- createDataPartition(player_data$goals, p = 0.8, list = FALSE)
str(train_sample)
```

```
##  int [1:1060, 1] 2 4 5 6 7 9 10 11 13 14 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "Resample1"
```

```
train <- player_data[train_sample, ]
testdata  <- player_data[-train_sample, ]
```

#SLR - Predicting goals model

```
model2 <- lm(goals~position_new+age+market_value+fee+loan++shots+games+time+key_passes+assists+x
G+xA+yellow_cards+red_cards, data =train)
summary(model2)
```

```
##
## Call:
## lm(formula = goals ~ position_new + age + market_value + fee +
##     loan + +shots + games + time + key_passes + assists + xG +
##     xA + yellow_cards + red_cards, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3578 -0.6067 -0.1566  0.7888  7.0058
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.517e-01  2.629e-01   1.338   0.1813
## position_newDefender     6.400e-02  8.313e-02   0.770   0.4416
## position_newGoalkeeper -1.745e-01  1.525e-01  -1.144   0.2528
## position_newMidfielder   4.819e-02  8.467e-02   0.569   0.5693
## age                     -2.154e-02  9.335e-03  -2.307   0.0212 *
## market_value            -5.420e-03  5.433e-03  -0.998   0.3187
## fee                      1.086e-02  5.323e-03   2.041   0.0415 *
## loanTRUE                 1.368e-01  7.738e-02   1.768   0.0774 .
## shots                   -1.395e-02  5.910e-03  -2.360   0.0185 *
## games                    4.208e-02  9.789e-03   4.299 1.88e-05 ***
## time                     9.079e-05  1.238e-04   0.733   0.4635
## key_passes              -4.204e-02  9.504e-03  -4.423 1.07e-05 ***
## assists                  3.572e-02  4.300e-02   0.831   0.4063
## xG                       9.584e-01  3.480e-02  27.542  < 2e-16 ***
## xA                       4.421e-01  8.884e-02   4.976 7.58e-07 ***
## yellow_cards            -1.979e-01  1.712e-02 -11.559  < 2e-16 ***
## red_cards                2.657e-02  1.069e-01   0.249   0.8038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.045 on 1043 degrees of freedom
## Multiple R-squared:  0.8247, Adjusted R-squared:  0.822
## F-statistic: 306.8 on 16 and 1043 DF,  p-value: < 2.2e-16
```

```
predictions2 <- predict(model2, testdata)
summary(predictions2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6164  0.7638  1.2379  1.7476  2.1857 23.8075
```

```
data.frame( R2 = R2(predictions2, testdata$goals),
            RMSE = RMSE(predictions2, testdata$goals),
            MAE = MAE(predictions2, testdata$goals))
```

```
##          R2     RMSE       MAE
## 1 0.8484075 1.109512 0.8206235
```

Best SLR for predicting goals

```
model2 <- lm(goals~position_new+age+fee+loan++shots+games+time+key_passes+xG+xA+yellow_cards, da
ta =train)
summary(model2)
```

```
##
## Call:
## lm(formula = goals ~ position_new + age + fee + loan + +shots +
##     games + time + key_passes + xG + xA + yellow_cards, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2048 -0.6419 -0.1605  0.8071  7.1665
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.337e-01  2.609e-01   1.279   0.2012
## position_newDefender     6.600e-02  8.280e-02   0.797   0.4256
## position_newGoalkeeper -1.716e-01  1.522e-01  -1.128   0.2598
## position_newMidfielder   4.880e-02  8.418e-02   0.580   0.5623
## age                     -2.204e-02  9.298e-03  -2.370   0.0180 *
## fee                      6.826e-03  3.668e-03   1.861   0.0630 .
## loanTRUE                 1.258e-01  7.622e-02   1.650   0.0992 .
## shots                   -1.469e-02  5.784e-03  -2.541   0.0112 *
## games                    4.730e-02  8.352e-03   5.663 1.92e-08 ***
## time                     3.849e-05  1.145e-04   0.336   0.7367
## key_passes              -4.276e-02  9.437e-03  -4.531 6.55e-06 ***
## xG                       9.562e-01  3.440e-02  27.796  < 2e-16 ***
## xA                       4.809e-01  7.732e-02   6.219 7.21e-10 ***
## yellow_cards            -1.974e-01  1.671e-02 -11.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.044 on 1046 degrees of freedom
## Multiple R-squared:  0.8244, Adjusted R-squared:  0.8223
## F-statistic: 377.9 on 13 and 1046 DF,  p-value: < 2.2e-16
```

```
predictions2 <- predict(model2, testdata)
summary(predictions2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6249  0.7889  1.2374  1.7495  2.1594 23.8242
```

```
data.frame( R2 = R2(predictions2, testdata$goals),
            RMSE = RMSE(predictions2, testdata$goals),
            MAE = MAE(predictions2, testdata$goals))
```

```
##          R2     RMSE       MAE
## 1 0.8500186 1.103653 0.8189285
```

#Train-Test split

```
set.seed(123)
train_sample <- createDataPartition(player_data$assists, p = 0.8, list = FALSE)
str(train_sample)
```

```
##  int [1:1060, 1] 2 5 6 7 8 9 11 12 15 19 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "Resample1"
```

```
train <- player_data[train_sample, ]
testdata  <- player_data[-train_sample, ]
```

#SLR - Predicting assists model

```
model4 <- lm(assists~position_new+age+market_value+fee+loan++shots+games+time+key_passes+goals+x
G+xA+yellow_cards+red_cards, data =train)
summary(model4)
```

```
##
## Call:
## lm(formula = assists ~ position_new + age + market_value + fee +
##     loan + +shots + games + time + key_passes + goals + xG +
##     xA + yellow_cards + red_cards, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4188 -0.3865 -0.0580  0.5474  4.2865
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -6.016e-01  1.953e-01  -3.081  0.00212 **
## position_newDefender    1.547e-01  6.033e-02   2.564  0.01047 *
## position_newGoalkeeper -1.404e-01  1.111e-01  -1.264  0.20662
## position_newMidfielder  9.215e-02  6.115e-02   1.507  0.13217
## age                     4.386e-03  6.891e-03   0.636  0.52465
## market_value           -9.289e-03  3.933e-03  -2.362  0.01836 *
## fee                     4.586e-04  3.866e-03   0.119  0.90560
## loanTRUE                6.627e-02  5.571e-02   1.190  0.23446
## shots                  -6.202e-03  4.364e-03  -1.421  0.15559
## games                   1.159e-01  6.319e-03  18.343  < 2e-16 ***
## time                   -8.834e-04  8.641e-05 -10.224  < 2e-16 ***
## key_passes             -1.986e-02  6.870e-03  -2.891  0.00391 **
## goals                  -4.607e-03  2.178e-02  -0.212  0.83253
## xG                     -2.418e-02  3.253e-02  -0.743  0.45737
## xA                      1.040e+00  5.669e-02  18.343  < 2e-16 ***
## yellow_cards           -5.692e-02  1.342e-02  -4.241 2.42e-05 ***
## red_cards              -6.600e-01  7.492e-02  -8.809  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7546 on 1043 degrees of freedom
## Multiple R-squared:  0.6873, Adjusted R-squared:  0.6825
## F-statistic: 143.3 on 16 and 1043 DF,  p-value: < 2.2e-16
```

```
predictions4 <- predict(model4, testdata)
summary(predictions4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.9459  0.9128  2.0253  1.7058  2.3206  6.9085
```

```
data.frame( R2 = R2(predictions4, testdata$assists),
            RMSE = RMSE(predictions4, testdata$assists),
            MAE = MAE(predictions4, testdata$assists))
```

```
##          R2     RMSE       MAE
## 1 0.7870352 0.651781 0.4935086
```