FIFA 19 Player Ratings Report

Rhys van den Handel

01/01/2021

Executive Summary

Every year EA sports releases the latest version FIFA. FIFA is a soccer video game available on most popular platforms that allows users to control players against other users or the computer. Released in September 2018, FIFA 19 contains over 18000 players from 205 clubs from around the world. When defining a players capability many aspects are taken into consideration. These aspects are broken down into attributes such as shooting power, goalkeeper diving and dribbling which are given a score out of 100. The attributes are then combined based on the position of the player to give an overall score.

The attribute scores are determined by EA sports using thier own method. The purose of this project was to build a system that can recommend an overall score to players based on factors that are known or are easy to determine. This was achieved by investigating the data that exists in the FIFA 19 player database.

After analysing the data, choosing variables that can be easily found or simply assigned, a complex regularisation model was built utilising as much information as possible to make informed decissions. The final result was evaluated using the RMSE method in conjunction with mean absolute error.

The final model used 10 variables split into 4 groups. Where possible the analysis was run through the Caret glm method and where not possible a penalty term optimised regularisation was done. While the individual models worked, the combined model made much better predictions and resulted in an RMSE of ~ 4.0 and a mean absolute error of ~ 2.8. The model had difficulty predicting outliers and for this reason the model should only be used to inform on ratings where the complex FIFA data is unavailable.

1. Introduction

The HarvardX data science certificate takes part over 9 Courses. This is the final Capstone project for individual learners. The project is a recommendation system chosen by the learner on any dataset. For this project FIFA 19 player data was chosen. The purpose is to take information on players and predict the overall ability out of 100 of the player.

EA Sports releases a new FIFA video game every year. This system looks at the FIFA 19 player dataset only. The dataset contains over 18000 players from more than 200 clubs around the world. There are 89 variables that FIFA uses to describe a player and thier ability. However, many of these attributes are complex based on FIFA's rating system. Some of the data FIFA uses to make decisions on a players ability is limited. Therefore, deriving an overall rating for a player may be difficult and cum bersome.

This project looks at variables that can be found easily or easily derived (value out of 5 vs value out of 100) and uses this to predict the overall rating of the player.

1.1 FIFA Players Dataset

The FIFA players dataset is a single repository of player data. It contains 89 variables which are used to describe each player. Players are identified by a unique ID. The variables can be broken into simple groups of information used for describing the player

1. Physical

- Age
- Height
- Weight
- Body.Type

2. General

- Nationality
- Club
- Jersey.Number

3. Simple Attributes

- Special
- Preferred.Foot
- International.Reputation
- Weak.Foot
- Skill.Moves
- Work.Rate
- Position

4. Monetary Values

- Value
- Wage
- Release.Clause

5. Complex Attributes

- All skill ratings
- Adjustment for position
- Picturs and Logos

All of these variables feed into the overall and potetntial rating of each player.

1.2 Limitations

As discussed earlier the complex attributes will not be used to make any jugements on the overall rating of each player. This is due to FIFA using the complex skill ratings out of 100 to directly inform the overall rating. These are complex and are created by FIFA based on expert opinion and known complex statistics. This project will only look at attributes from the first 4 groups mentioned in the section above.

This project has been run on a middle of the range laptop. For the purpose of simplicity all numeric grouped values will be evaluated using the same method of glm. The model was built on each individual group and combined at the end to save on processing. This resulted in the final combined model not being as effective as a single combined and trained model.

1.3 Evaluation Method

"The RMSE is a quadratic scoring rule which measures the average magnitude of the error. The equation for the RMSE is given in both of the references. Expressing the formula in words, the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable." - http://www.eumetrain.org

As described by eumetrain.org RMSE penalizes high errors. This makes it an ideal evaluation method for recommendation systems. A lower RMSE means that there is a lower likelihood of a very poor recommendation being made which could result in a player been grosely over or under rated.

The accuracy of each model will by evaluated using Mean Absolute Error the following formula: |mean(prediction - actual)|. The RMSE indicated the quality of the model however, the absolute error will show the true value of the errors vs the overall score.

2. Methodology

The methodology used to evaluate the project is shown and discussed bellow:

2.1 Data Import

The data has been made available on Kaggle.com by user Karan Gadiya. The dataset can be found at the following link: *"https://www.kaggle.com/karangadiya/fifa19"* The data is downloaded into the project in a csv format and run to create the PlayerData set.

The dataset has approximately 60 rows containing NA's these are removed. The data is then loaded into a working and validation set based on a split of 10%. A seed of 1 is used to ensure consistency with each run and to allow any users to test the model on the same data.

2.2 Data Analysis

The data analysis allows for a better understanding of the data components and their interactions. The process of running the data analysis was primarily to determine the usability of each attribute listed in section 1.1. The key analysis variables were the impact on overall rating and the count of each group. The following was investigated:

1. FIFA dataset

- view the data
- check for nulls
- Distribution of Player Ratings

2. Physical

- Age
  - Distribution of ages
  - Number of players and mean rating by age
- Height and Weight
  - Convert to numeric values
  - Distributions of height and weight
  - Height and weight vs overall rating
- Body.Type
  - plot gainst overall

3. General

- Nationality
  - Summary of counts
  - Countries with most and least players
  - Countries with best and worst Players
- Club
  - Summary of counts
  - Clubs with most and least players
  - Clubs with best and worst Players
- Jersey.Number
  - Summary of counts
  - Jersey numbers with most and least players
  - Jersey numbers Clubs with best and worst Players
  - Plot of number vs average rating

4. Simple Attributes

- Test for quantifyability
- Plot against overall

5. Monetary Columns

- Fix leading characters to make numeric variable
- Plot each monetary attribute against overall rating

The purpose of the data analysis would be to allow for informed data cleaning decisions that would have a positive outcome on the final results. The data cleaning was used as part of the predictive model as discussed in section 2.3 bellow. The analysis showed which attributes were suitable for the model and which were to be excluded from the model.

2.3 Predictive Model

Due to the limitations discussed in section 1.2 *Limitations* above the predictive model was built in sections based on the classification of data. The model was built using simple glm methods and regularisation for non-numeric or non-linear models.

2.3.1 Splitting Data The data was cleaned according to the analysis made. It was then split into a training and testing set. The partition is 10% this allows for a large proportion of training data. This was due to the small numbers of players per jersey number and clubs. Ensuring a good fit can be made.

2.3.2 Prediction Models As introduced the the model was built from a simplified groups and combined into more complex model:

- Mean Model:
  - Using the simple mean of the dataset
  - The mean model serves only to ensure the models offer an improvemnt to the overall prediction.
- Caret GLM Model:
  - Using caret train with a GLM method on linear numeric columns per group
- Regularization model:
  - Using a penalty term regularise non numeric and non-linear attributes
  - Optimized for the best penalty term
- Combined Model:
  - Combine the models
  - Retune the regularisation models for optimised output

2.4 Validation

Once an acceptable RMSE has been identified the final iteration would be the validation, run through the final combined model.

3. Results and Discussion

This sections presents the results of the methodology.

3.1 Data Import and Preparation

```
##########################################################
# LOAD DATA
##########################################################

#Download Data from internet
#temp <- tempfile()

#url <- "https://www.kaggle.com/karangadiya/fifa19/download/archive.zip"
#download.file(url, temp)
#unzip(temp, "archive")
#data<-read.csv("/archive/data.csv", header = TRUE)

#unlink(temp)


#Read the data in from project
data <- read.csv(".\\data.csv", header = TRUE)


#Replace data with readcsv for using project dataset
PlayerData <- data.frame(data)

#view the dataset
head(PlayerData)
```

```
##   ï..     ID              Name Age
## 1   0 158023          L. Messi  31
## 2   1  20801 Cristiano Ronaldo  33
## 3   2 190871         Neymar Jr  26
## 4   3 193080            De Gea  27
## 5   4 192985      K. De Bruyne  27
## 6   5 183277         E. Hazard  27
##                                            Photo Nationality
## 1 https://cdn.sofifa.org/players/4/19/158023.png   Argentina
## 2  https://cdn.sofifa.org/players/4/19/20801.png    Portugal
## 3 https://cdn.sofifa.org/players/4/19/190871.png      Brazil
## 4 https://cdn.sofifa.org/players/4/19/193080.png       Spain
## 5 https://cdn.sofifa.org/players/4/19/192985.png     Belgium
## 6 https://cdn.sofifa.org/players/4/19/183277.png     Belgium
##                                  Flag Overall Potential                Club
## 1 https://cdn.sofifa.org/flags/52.png      94        94        FC Barcelona
## 2 https://cdn.sofifa.org/flags/38.png      94        94            Juventus
## 3 https://cdn.sofifa.org/flags/54.png      92        93 Paris Saint-Germain
## 4 https://cdn.sofifa.org/flags/45.png      91        93   Manchester United
## 5  https://cdn.sofifa.org/flags/7.png      91        92     Manchester City
## 6  https://cdn.sofifa.org/flags/7.png      91        91             Chelsea
##                                     Club.Logo      Value    Wage Special
## 1 https://cdn.sofifa.org/teams/2/light/241.png â‚¬110.5M â‚¬565K    2202
## 2  https://cdn.sofifa.org/teams/2/light/45.png    â‚¬77M â‚¬405K    2228
```

```
## 3  https://cdn.sofifa.org/teams/2/light/73.png â‚¬118.5M â‚¬290K    2143
## 4  https://cdn.sofifa.org/teams/2/light/11.png    â‚¬72M â‚¬260K    1471
## 5  https://cdn.sofifa.org/teams/2/light/10.png   â‚¬102M â‚¬355K    2281
## 6   https://cdn.sofifa.org/teams/2/light/5.png    â‚¬93M â‚¬340K    2142
##   Preferred.Foot International.Reputation Weak.Foot Skill.Moves      Work.Rate
## 1           Left                        5         4           4 Medium/ Medium
## 2          Right                        5         4           5      High/ Low
## 3          Right                        5         5           5   High/ Medium
## 4          Right                        4         3           1 Medium/ Medium
## 5          Right                        4         5           4     High/ High
## 6          Right                        4         4           4   High/ Medium
##    Body.Type Real.Face Position Jersey.Number       Joined Loaned.From
## 1      Messi       Yes       RF            10  Jul 1, 2004
## 2 C. Ronaldo       Yes       ST             7 Jul 10, 2018
## 3     Neymar       Yes       LW            10  Aug 3, 2017
## 4       Lean       Yes       GK             1  Jul 1, 2011
## 5     Normal       Yes      RCM             7 Aug 30, 2015
## 6     Normal       Yes       LF            10  Jul 1, 2012
##   Contract.Valid.Until Height Weight   LS   ST   RS   LW   LF   CF   RF   RW
## 1                 2021    5'7 159lbs 88+2 88+2 88+2 92+2 93+2 93+2 93+2 92+2
## 2                 2022    6'2 183lbs 91+3 91+3 91+3 89+3 90+3 90+3 90+3 89+3
## 3                 2022    5'9 150lbs 84+3 84+3 84+3 89+3 89+3 89+3 89+3 89+3
## 4                 2020    6'4 168lbs
## 5                 2023   5'11 154lbs 82+3 82+3 82+3 87+3 87+3 87+3 87+3 87+3
## 6                 2020    5'8 163lbs 83+3 83+3 83+3 89+3 88+3 88+3 88+3 89+3
##    LAM  CAM  RAM   LM  LCM   CM  RCM   RM  LWB  LDM  CDM  RDM  RWB   LB  LCB
## 1 93+2 93+2 93+2 91+2 84+2 84+2 84+2 91+2 64+2 61+2 61+2 61+2 64+2 59+2 47+2
## 2 88+3 88+3 88+3 88+3 81+3 81+3 81+3 88+3 65+3 61+3 61+3 61+3 65+3 61+3 53+3
## 3 89+3 89+3 89+3 88+3 81+3 81+3 81+3 88+3 65+3 60+3 60+3 60+3 65+3 60+3 47+3
## 4
## 5 88+3 88+3 88+3 88+3 87+3 87+3 87+3 88+3 77+3 77+3 77+3 77+3 77+3 73+3 66+3
## 6 89+3 89+3 89+3 89+3 82+3 82+3 82+3 89+3 66+3 63+3 63+3 63+3 66+3 60+3 49+3
##     CB  RCB   RB Crossing Finishing HeadingAccuracy ShortPassing Volleys
## 1 47+2 47+2 59+2       84        95              70           90      86
## 2 53+3 53+3 61+3       84        94              89           81      87
## 3 47+3 47+3 60+3       79        87              62           84      84
## 4                      17        13              21           50      13
## 5 66+3 66+3 73+3       93        82              55           92      82
## 6 49+3 49+3 60+3       81        84              61           89      80
##   Dribbling Curve FKAccuracy LongPassing BallControl Acceleration SprintSpeed
## 1        97    93         94          87          96           91          86
## 2        88    81         76          77          94           89          91
## 3        96    88         87          78          95           94          90
## 4        18    21         19          51          42           57          58
## 5        86    85         83          91          91           78          76
## 6        95    83         79          83          94           94          88
##   Agility Reactions Balance ShotPower Jumping Stamina Strength LongShots
## 1      91        95      95        85      68      72       59        94
## 2      87        96      70        95      95      88       79        93
## 3      96        94      84        80      61      81       49        82
## 4      60        90      43        31      67      43       64        12
## 5      79        91      77        91      63      90       75        91
## 6      95        90      94        82      56      83       66        80
##   Aggression Interceptions Positioning Vision Penalties Composure Marking
```

```
## 1          48             22           94     94         75          96       33
## 2          63             29           95     82         85          95       28
## 3          56             36           89     87         81          94       27
## 4          38             30           12     68         40          68       15
## 5          76             61           87     94         79          88       68
## 6          54             41           87     89         86          91       34
##   StandingTackle SlidingTackle GKDiving GKHandling GKKicking GKPositioning
## 1             28            26        6         11        15            14
## 2             31            23        7         11        15            14
## 3             24            33        9          9        15            15
## 4             21            13       90         85        87            88
## 5             58            51       15         13         5            10
## 6             27            22       11         12         6             8
##   GKReflexes Release.Clause
## 1          8      â‚¬226.5M
## 2         11      â‚¬127.1M
## 3         11      â‚¬228.1M
## 4         94      â‚¬138.6M
## 5         13      â‚¬196.4M
## 6          8      â‚¬172.1M
```

```r
any(is.na(PlayerData)) #True: therefore there are NA's
```

```
## [1] TRUE
```

```r
nrow(PlayerData) # 18207
```

```
## [1] 18207
```

```r
#Remove the NA's
PlayerData <- PlayerData %>% drop_na()
nrow(PlayerData) # 18147 Therefore, only 60 rows dropped
```

```
## [1] 18147
```

Due to the data containing nulls, the nulls were removed. This elimitnated 60 rows (0.3%) of the data

```r
##########################################################
# LOAD INTO WORKING AND VALIDATION SET
##########################################################

# Validation set will be 10% of the dataset
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
test_index <- createDataPartition(y = PlayerData$Overall, times = 1, p = 0.1, list = FALSE)
players <- PlayerData[-test_index,]
validation <- PlayerData[test_index,]
```

The data was then split into a working and validation dataset.

3.2 Data Analysis

Data exploration was an integral portion of the project this was to ensure the variables chosen for the model would be correct and useful.

```r
#Checking the players data set
names(players)
```

3.2.1 FIFA Dataset

```
##  [1] "ï.."                      "ID"
##  [3] "Name"                     "Age"
##  [5] "Photo"                    "Nationality"
##  [7] "Flag"                     "Overall"
##  [9] "Potential"                "Club"
## [11] "Club.Logo"                "Value"
## [13] "Wage"                     "Special"
## [15] "Preferred.Foot"           "International.Reputation"
## [17] "Weak.Foot"                "Skill.Moves"
## [19] "Work.Rate"                "Body.Type"
## [21] "Real.Face"                "Position"
## [23] "Jersey.Number"            "Joined"
## [25] "Loaned.From"              "Contract.Valid.Until"
## [27] "Height"                   "Weight"
## [29] "LS"                       "ST"
## [31] "RS"                       "LW"
## [33] "LF"                       "CF"
## [35] "RF"                       "RW"
## [37] "LAM"                      "CAM"
## [39] "RAM"                      "LM"
## [41] "LCM"                      "CM"
## [43] "RCM"                      "RM"
## [45] "LWB"                      "LDM"
## [47] "CDM"                      "RDM"
## [49] "RWB"                      "LB"
## [51] "LCB"                      "CB"
## [53] "RCB"                      "RB"
## [55] "Crossing"                 "Finishing"
## [57] "HeadingAccuracy"          "ShortPassing"
## [59] "Volleys"                  "Dribbling"
## [61] "Curve"                    "FKAccuracy"
## [63] "LongPassing"              "BallControl"
## [65] "Acceleration"             "SprintSpeed"
## [67] "Agility"                  "Reactions"
## [69] "Balance"                  "ShotPower"
## [71] "Jumping"                  "Stamina"
## [73] "Strength"                 "LongShots"
## [75] "Aggression"               "Interceptions"
## [77] "Positioning"              "Vision"
## [79] "Penalties"                "Composure"
## [81] "Marking"                  "StandingTackle"
## [83] "SlidingTackle"            "GKDiving"
## [85] "GKHandling"               "GKKicking"
## [87] "GKPositioning"            "GKReflexes"
```

```
## [89] "Release.Clause"
```

```r
head(players)
```

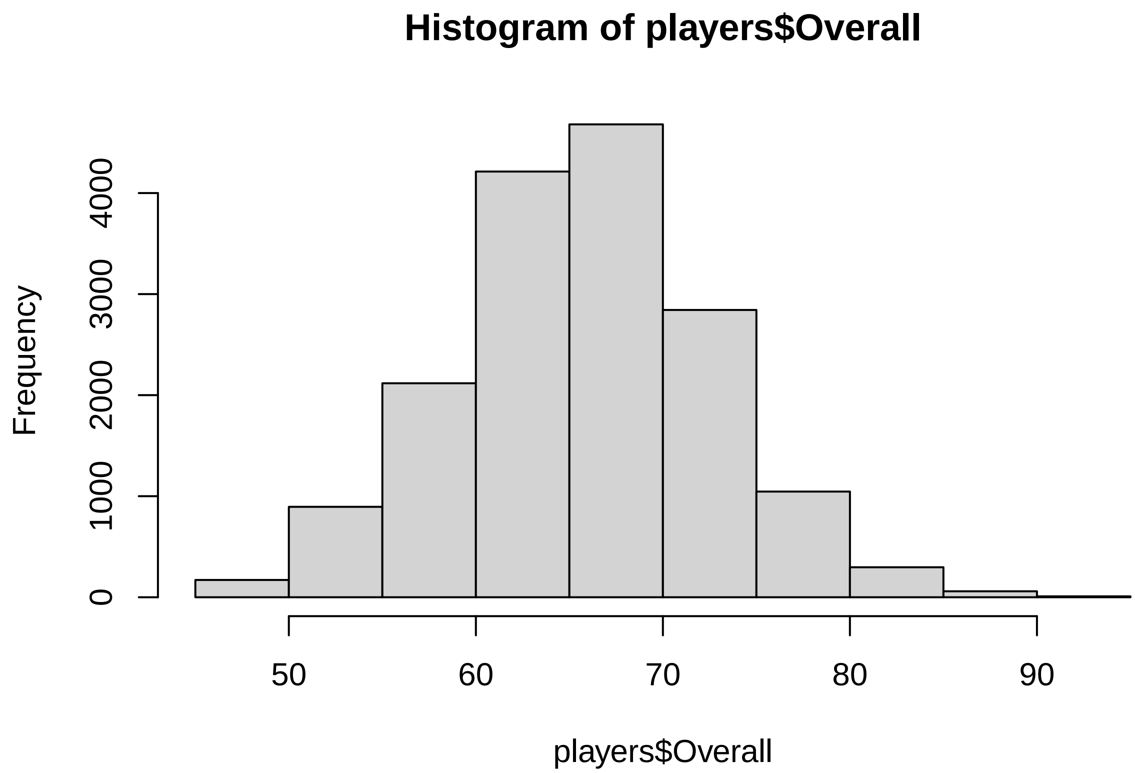
```
##   ï..     ID              Name Age
## 1   0 158023          L. Messi  31
## 2   1  20801 Cristiano Ronaldo  33
## 3   2 190871         Neymar Jr  26
## 4   3 193080            De Gea  27
## 5   4 192985      K. De Bruyne  27
## 6   5 183277         E. Hazard  27
##                                            Photo Nationality
## 1 https://cdn.sofifa.org/players/4/19/158023.png   Argentina
## 2  https://cdn.sofifa.org/players/4/19/20801.png    Portugal
## 3 https://cdn.sofifa.org/players/4/19/190871.png      Brazil
## 4 https://cdn.sofifa.org/players/4/19/193080.png       Spain
## 5 https://cdn.sofifa.org/players/4/19/192985.png     Belgium
## 6 https://cdn.sofifa.org/players/4/19/183277.png     Belgium
##                                  Flag Overall Potential                Club
## 1 https://cdn.sofifa.org/flags/52.png      94        94        FC Barcelona
## 2 https://cdn.sofifa.org/flags/38.png      94        94            Juventus
## 3 https://cdn.sofifa.org/flags/54.png      92        93 Paris Saint-Germain
## 4 https://cdn.sofifa.org/flags/45.png      91        93   Manchester United
## 5  https://cdn.sofifa.org/flags/7.png      91        92     Manchester City
## 6  https://cdn.sofifa.org/flags/7.png      91        91             Chelsea
##                                      Club.Logo     Value    Wage Special
## 1 https://cdn.sofifa.org/teams/2/light/241.png â‚¬110.5M â‚¬565K    2202
## 2  https://cdn.sofifa.org/teams/2/light/45.png    â‚¬77M â‚¬405K    2228
## 3  https://cdn.sofifa.org/teams/2/light/73.png â‚¬118.5M â‚¬290K    2143
## 4  https://cdn.sofifa.org/teams/2/light/11.png    â‚¬72M â‚¬260K    1471
## 5  https://cdn.sofifa.org/teams/2/light/10.png   â‚¬102M â‚¬355K    2281
## 6   https://cdn.sofifa.org/teams/2/light/5.png    â‚¬93M â‚¬340K    2142
##   Preferred.Foot International.Reputation Weak.Foot Skill.Moves      Work.Rate
## 1           Left                        5         4           4 Medium/ Medium
## 2          Right                        5         4           5      High/ Low
## 3          Right                        5         5           5   High/ Medium
## 4          Right                        4         3           1 Medium/ Medium
## 5          Right                        4         5           4     High/ High
## 6          Right                        4         4           4   High/ Medium
##    Body.Type Real.Face Position Jersey.Number       Joined Loaned.From
## 1      Messi       Yes       RF            10  Jul 1, 2004
## 2 C. Ronaldo       Yes       ST             7 Jul 10, 2018
## 3     Neymar       Yes       LW            10  Aug 3, 2017
## 4       Lean       Yes       GK             1  Jul 1, 2011
## 5     Normal       Yes      RCM             7 Aug 30, 2015
## 6     Normal       Yes       LF            10  Jul 1, 2012
##   Contract.Valid.Until Height Weight   LS   ST   RS   LW   LF   CF   RF   RW
## 1                 2021    5'7 159lbs 88+2 88+2 88+2 92+2 93+2 93+2 93+2 92+2
## 2                 2022    6'2 183lbs 91+3 91+3 91+3 89+3 90+3 90+3 90+3 89+3
## 3                 2022    5'9 150lbs 84+3 84+3 84+3 89+3 89+3 89+3 89+3 89+3
## 4                 2020    6'4 168lbs
## 5                 2023   5'11 154lbs 82+3 82+3 82+3 87+3 87+3 87+3 87+3 87+3
## 6                 2020    5'8 163lbs 83+3 83+3 83+3 89+3 88+3 88+3 88+3 89+3
##    LAM  CAM  RAM   LM  LCM   CM  RCM   RM  LWB  LDM  CDM  RDM  RWB   LB  LCB
## 1 93+2 93+2 93+2 91+2 84+2 84+2 84+2 91+2 64+2 61+2 61+2 61+2 64+2 59+2 47+2
```

```
## 2 88+3 88+3 88+3 88+3 81+3 81+3 81+3 88+3 65+3 61+3 61+3 61+3 65+3 61+3 53+3
## 3 89+3 89+3 89+3 88+3 81+3 81+3 81+3 88+3 65+3 60+3 60+3 60+3 65+3 60+3 47+3
## 4
## 5 88+3 88+3 88+3 88+3 87+3 87+3 87+3 88+3 77+3 77+3 77+3 77+3 77+3 73+3 66+3
## 6 89+3 89+3 89+3 89+3 82+3 82+3 82+3 89+3 66+3 63+3 63+3 63+3 66+3 60+3 49+3
##     CB  RCB   RB Crossing Finishing HeadingAccuracy ShortPassing Volleys
## 1 47+2 47+2 59+2       84        95              70           90      86
## 2 53+3 53+3 61+3       84        94              89           81      87
## 3 47+3 47+3 60+3       79        87              62           84      84
## 4                      17        13              21           50      13
## 5 66+3 66+3 73+3       93        82              55           92      82
## 6 49+3 49+3 60+3       81        84              61           89      80
##   Dribbling Curve FKAccuracy LongPassing BallControl Acceleration SprintSpeed
## 1        97    93         94          87          96           91          86
## 2        88    81         76          77          94           89          91
## 3        96    88         87          78          95           94          90
## 4        18    21         19          51          42           57          58
## 5        86    85         83          91          91           78          76
## 6        95    83         79          83          94           94          88
##   Agility Reactions Balance ShotPower Jumping Stamina Strength LongShots
## 1      91        95      95        85      68      72       59        94
## 2      87        96      70        95      95      88       79        93
## 3      96        94      84        80      61      81       49        82
## 4      60        90      43        31      67      43       64        12
## 5      79        91      77        91      63      90       75        91
## 6      95        90      94        82      56      83       66        80
##   Aggression Interceptions Positioning Vision Penalties Composure Marking
## 1         48            22          94     94        75        96      33
## 2         63            29          95     82        85        95      28
## 3         56            36          89     87        81        94      27
## 4         38            30          12     68        40        68      15
## 5         76            61          87     94        79        88      68
## 6         54            41          87     89        86        91      34
##   StandingTackle SlidingTackle GKDiving GKHandling GKKicking GKPositioning
## 1             28            26        6         11        15            14
## 2             31            23        7         11        15            14
## 3             24            33        9          9        15            15
## 4             21            13       90         85        87            88
## 5             58            51       15         13         5            10
## 6             27            22       11         12         6             8
##   GKReflexes Release.Clause
## 1          8      â‚¬226.5M
## 2         11      â‚¬127.1M
## 3         11      â‚¬228.1M
## 4         94      â‚¬138.6M
## 5         13      â‚¬196.4M
## 6          8      â‚¬172.1M
```

```r
any(is.na(players))
```

```
## [1] FALSE
```

The datset contains many attributes however, most are complex. These columns will be removed and not selected for analysis. There are no nulls in the dataset.

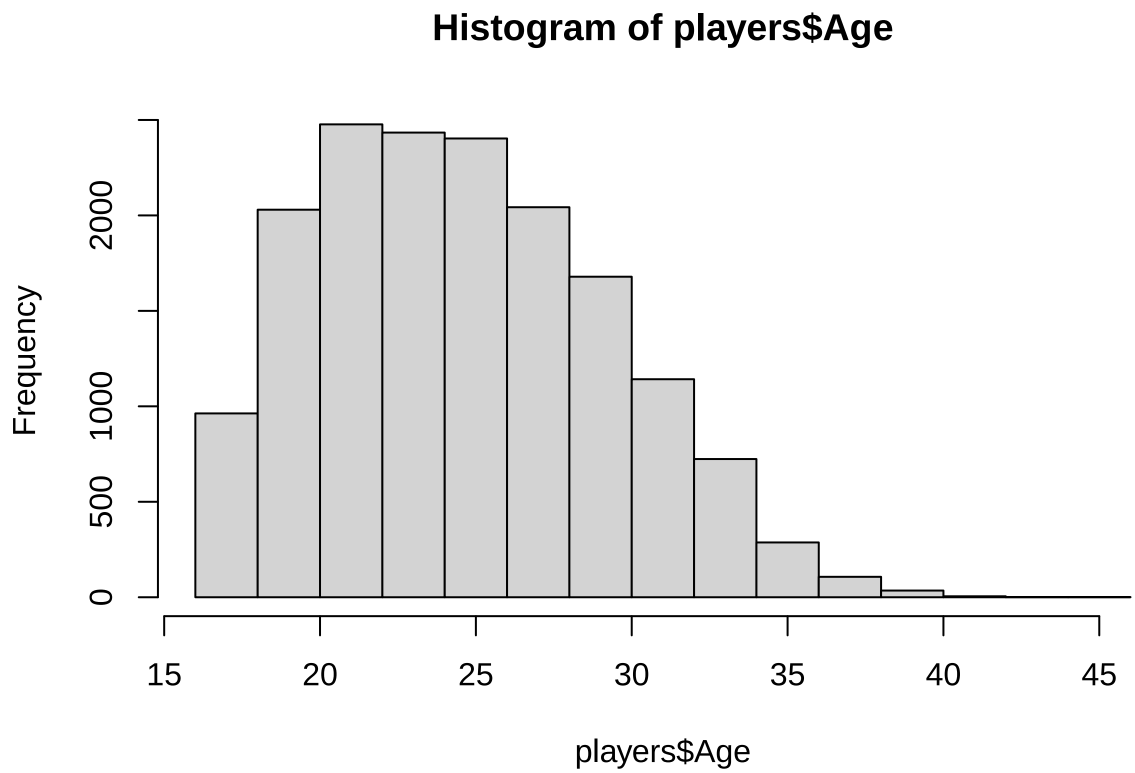


The distribution of player ratings is normal. This is ideal for the purpose of this project.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.00   62.00   66.00   66.25   71.00   94.00
```

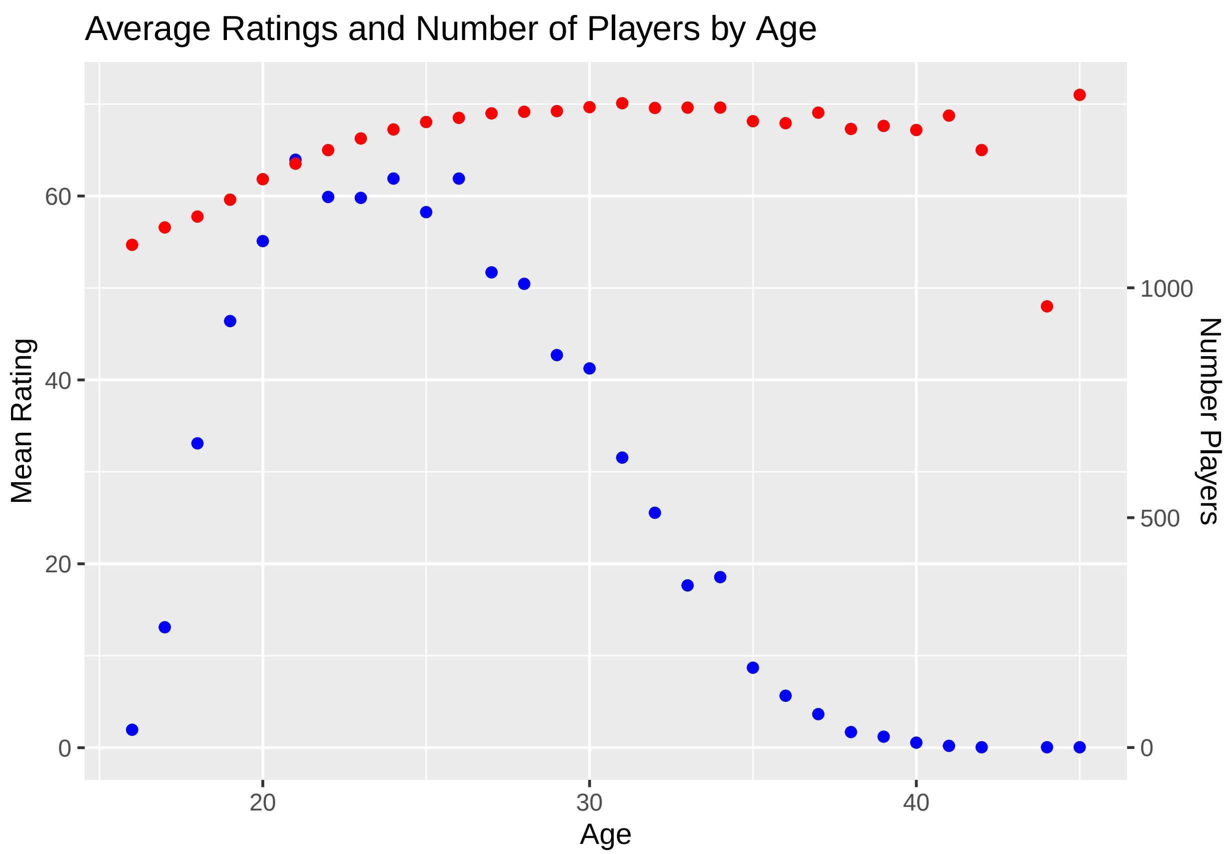
The summary statistics indicate the mean and median of 66 as well as a small interquartile range.

3.2.2 Physical The physical attributes were broken down as follows:



Although not perfectly normal there is a decent shape to the distribution of ages.

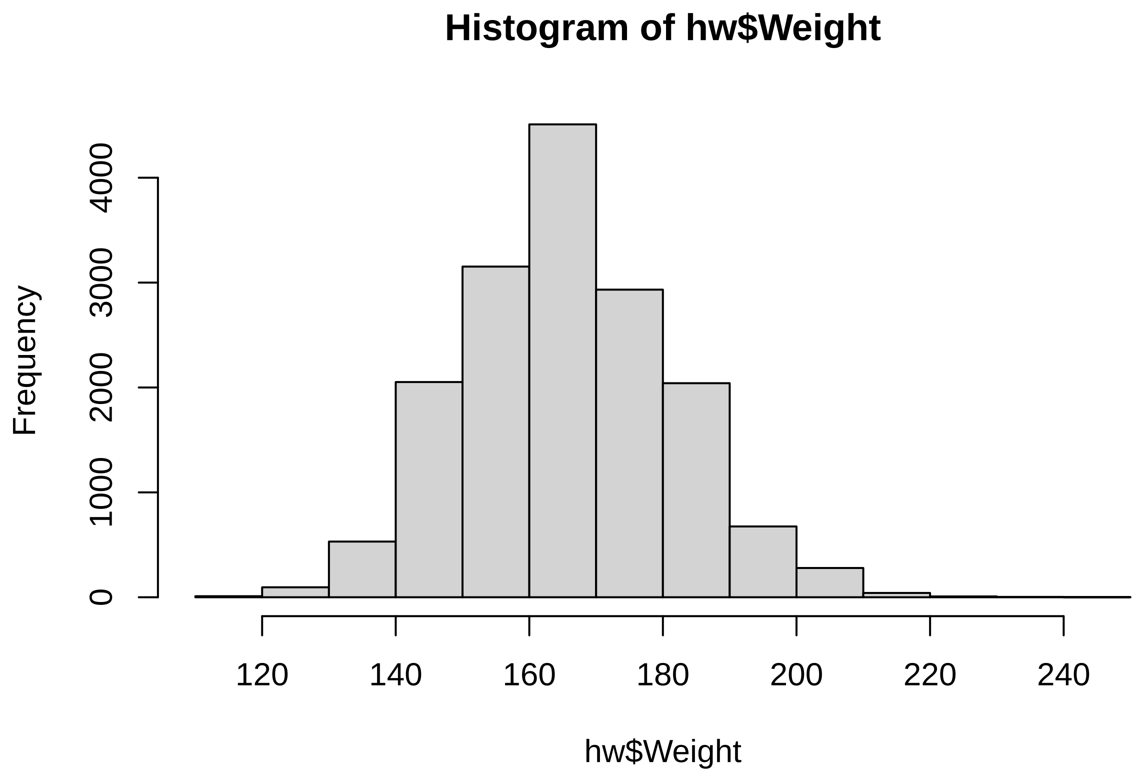
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



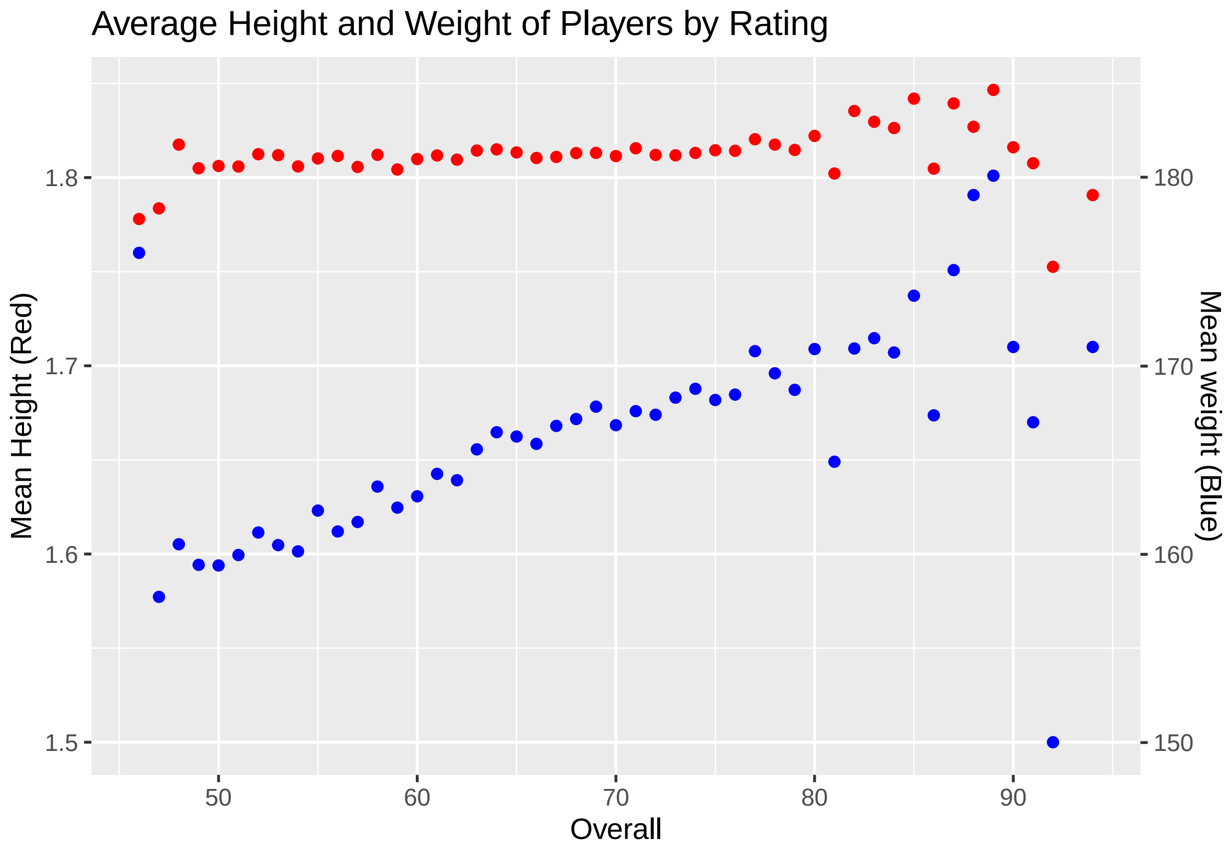
There is a clear relationship between age and mean rating. However, this is slightly affected by low volumes of older players resulting in slight variance.

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(hwcols)` instead of `hwcols` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

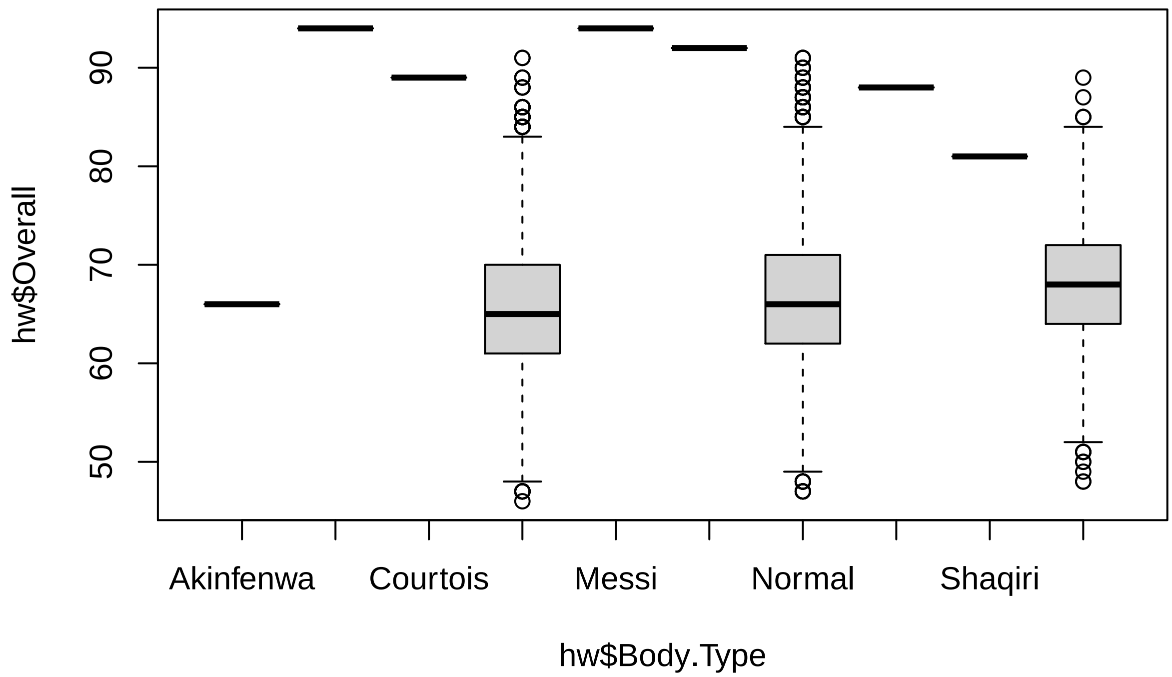
Histogram of hw$Height
Frequency
0
1000
2000
3000
4000
1.5
1.6
1.7
1.8
1.9
2.0
2.1
hw$Height




```
## `summarise()` ungrouping output (override with `.groups` argument)
```



Both height and weight are nicely distributed. However, only weight has any correlation to overall rating. Therefore height will be excluded from the model. In order to utilise weight effectively weight must be converted to a numeric column for the model.



As shown there is no correlation or identifiable impact to body type. therefore body type will be excluded from the model.

```
nation <- players %>% group_by(Nationality) %>%
  summarize(n=n(),rating=mean(Overall))
```

3.2.3 General

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
summary(nation$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     3.0    12.0   100.2    74.0  1499.0
```

```
#Nations with most players
nation %>% arrange(desc(n)) %>%
  top_n(10,n)
```

```
## # A tibble: 10 x 3
##    Nationality     n rating
##    <chr>       <int>  <dbl>
##  1 England      1499   63.4
##  2 Germany      1070   66.0
##  3 Spain         962   69.6
##  4 Argentina     854   68.6
##  5 France        829   67.7
```

```
##   6 Brazil        752   71.4
##   7 Italy         626   68.1
##   8 Colombia      537   65.5
##   9 Japan         430   62.7
##  10 Netherlands   417   67.6
```

```r
#Nations with least players
nation %>% arrange((n)) %>%
  top_n(10,-n)
```

```
## # A tibble: 24 x 3
##    Nationality     n rating
##    <chr>       <int>  <dbl>
##  1 Andorra         1     62
##  2 Belize          1     60
##  3 Botswana        1     56
##  4 Ethiopia        1     64
##  5 Fiji            1     71
##  6 Grenada         1     63
##  7 Guam            1     67
##  8 Indonesia       1     56
##  9 Jordan          1     63
## 10 Kuwait          1     70
## # ... with 14 more rows
```

```r
#Nations with best players
nation %>% filter(n>12) %>%
  arrange(desc(rating)) %>%
  top_n(10,rating)
```

```
## # A tibble: 10 x 3
##    Nationality     n rating
##    <chr>       <int>  <dbl>
##  1 Israel         14   72.1
##  2 Cape Verde     17   71.5
##  3 Brazil        752   71.4
##  4 Portugal      284   71.3
##  5 Algeria        55   70.6
##  6 Peru           32   70.4
##  7 Egypt          28   70.4
##  8 Uruguay       138   70.2
##  9 Gabon          14   70.1
## 10 Morocco        80   70.0
```

```r
#Nations with worst players
nation %>% filter(n>12) %>%
  arrange((rating)) %>%
  top_n(10,-rating)
```

```
## # A tibble: 10 x 3
##    Nationality             n rating
##    <chr>               <int>  <dbl>
##  1 China PR              350   59.9
##  2 India                  18   60
##  3 Saudi Arabia          304   60.7
##  4 Republic of Ireland   331   60.8
```

```
##   5 Australia              214   62.7
##   6 Japan                  430   62.7
##   7 Canada                  58   62.8
##   8 New Zealand             42   62.8
##   9 Northern Ireland        69   63.1
##  10 Poland                 312   63.1
```

Nationality has an impact but due to the inconsistent number of players of each nation it would be difficult to use. Therefore nationality will be excluded from the model.

```
clubs <- players %>% group_by(Club) %>%
  summarize(n=n(),rating=mean(Overall))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
summary(clubs$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   23.00   25.00   25.05   27.00  205.00
```

```
#Clubs with most players
clubs %>% arrange(desc(n)) %>%
  top_n(10,n)
```

```
## # A tibble: 16 x 3
##    Club                            n rating
##    <chr>                       <int>  <dbl>
##  1 ""                            205   67.8
##  2 "Borussia Dortmund"            32   75.3
##  3 "FC Barcelona"                 32   77.7
##  4 "Levante UD"                   32   72.2
##  5 "Manchester City"              32   76.4
##  6 "Arsenal"                      31   75.3
##  7 "AS Monaco"                    31   73.1
##  8 "Brighton & Hove Albion"       31   71
##  9 "Burnley"                      31   71.1
## 10 "Chelsea"                      31   77.3
## 11 "Crystal Palace"               31   71.1
## 12 "Fortuna DÃ¼sseldorf"          31   68.2
## 13 "Leicester City"               31   74.1
## 14 "Southampton"                  31   71.7
## 15 "Valencia CF"                  31   74.5
## 16 "Wolverhampton Wanderers"      31   68.6
```

```
#Clubs with least players
clubs %>% arrange((n)) %>%
  top_n(10,-n)
```

```
## # A tibble: 16 x 3
##    Club                        n rating
##    <chr>                   <int>  <dbl>
##  1 Ã–stersunds FK             15   63.7
##  2 Cruzeiro                  15   71.8
##  3 Limerick FC               15   55
##  4 AtlÃ©tico Paranaense       17   69
##  5 CearÃ¡ Sporting Club       17   68.2
##  6 Derry City                17   55.9
```

```
##  7 Sligo Rovers            17   56.5
##  8 Sport Club do Recife    17   69.5
##  9 ÅšlÄ...sk WrocÅ‚aw        18   62.6
## 10 Botafogo                18   71.3
## 11 FK Haugesund            18   62.7
## 12 GrÃªmio                 18   73.2
## 13 Grenoble Foot 38        18   64.7
## 14 KasimpaÅŸa SK            18   67.6
## 15 ParanÃ¡                 18   69
## 16 VitÃ³ria                18   70.2
```

```
#Clubs with best players
clubs %>% arrange(desc(rating)) %>%
  top_n(10,rating)
```

```
## # A tibble: 10 x 3
##    Club                     n rating
##    <chr>                <int>  <dbl>
##  1 Juventus                22   81.9
##  2 Napoli                  24   80.1
##  3 Inter                   22   79.7
##  4 Real Madrid             30   79.6
##  5 Milan                   27   78.1
##  6 Roma                    24   78
##  7 FC Barcelona            32   77.7
##  8 Paris Saint-Germain     27   77.7
##  9 Manchester United       27   77.6
## 10 Chelsea                 31   77.3
```

```
#Clubs with worst players
clubs %>% arrange((rating)) %>%
  top_n(10,-rating)
```

```
## # A tibble: 10 x 3
##    Club                  n rating
##    <chr>             <int>  <dbl>
##  1 Bray Wanderers       22   53.8
##  2 Bohemian FC          22   55
##  3 Limerick FC          15   55
##  4 Derry City           17   55.9
##  5 Sligo Rovers         17   56.5
##  6 Crewe Alexandra      25   56.6
##  7 Waterford FC         23   57
##  8 Cambridge United     25   57.1
##  9 Morecambe            26   57.4
## 10 Cork City            21   57.4
```

Clubs are definitely a good option for training there is a clear difference between clubs and the number of players is fairly consistent

```
Jersey <- players %>% group_by(Jersey.Number) %>%
  summarize(n=n(),rating=mean(Overall))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
summary(Jersey$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     1.0     8.0    33.5   166.6   364.5   552.0
```

```r
#Clubs with most players
Jersey %>% arrange(desc(n)) %>%
  top_n(10,n)
```

```
## # A tibble: 11 x 3
##    Jersey.Number     n rating
##            <int> <int>  <dbl>
##  1             7   552   68.8
##  2             8   546   68.9
##  3            10   542   70.3
##  4            11   529   68.2
##  5             6   522   68.3
##  6             5   519   68.6
##  7             4   517   67.8
##  8             9   504   69.3
##  9             1   503   68.3
## 10            18   501   66.5
## 11            20   501   66.7
```

```r
#Clubs with least players
Jersey %>% arrange((n)) %>%
  top_n(10,-n)
```

```
## # A tibble: 11 x 3
##    Jersey.Number     n rating
##            <int> <int>  <dbl>
##  1            79     1   71
##  2            64     2   62.5
##  3            74     2   67.5
##  4            63     3   67.7
##  5            68     3   68.7
##  6            76     3   73
##  7            59     4   57.5
##  8            65     4   58.5
##  9            81     4   65.2
## 10            84     4   63.8
## 11            86     4   64
```

```r
#Clubs with best players
Jersey %>% arrange(desc(rating)) %>%
  top_n(10,rating)
```

```
## # A tibble: 10 x 3
##    Jersey.Number     n rating
##            <int> <int>  <dbl>
##  1            76     3   73
##  2            92     7   71.3
##  3            79     1   71
##  4            10   542   70.3
##  5             9   504   69.3
##  6            87    10   68.9
##  7             8   546   68.9
##  8             7   552   68.8
```

```
## 9              68     3   68.7
## 10             69     6   68.7
```

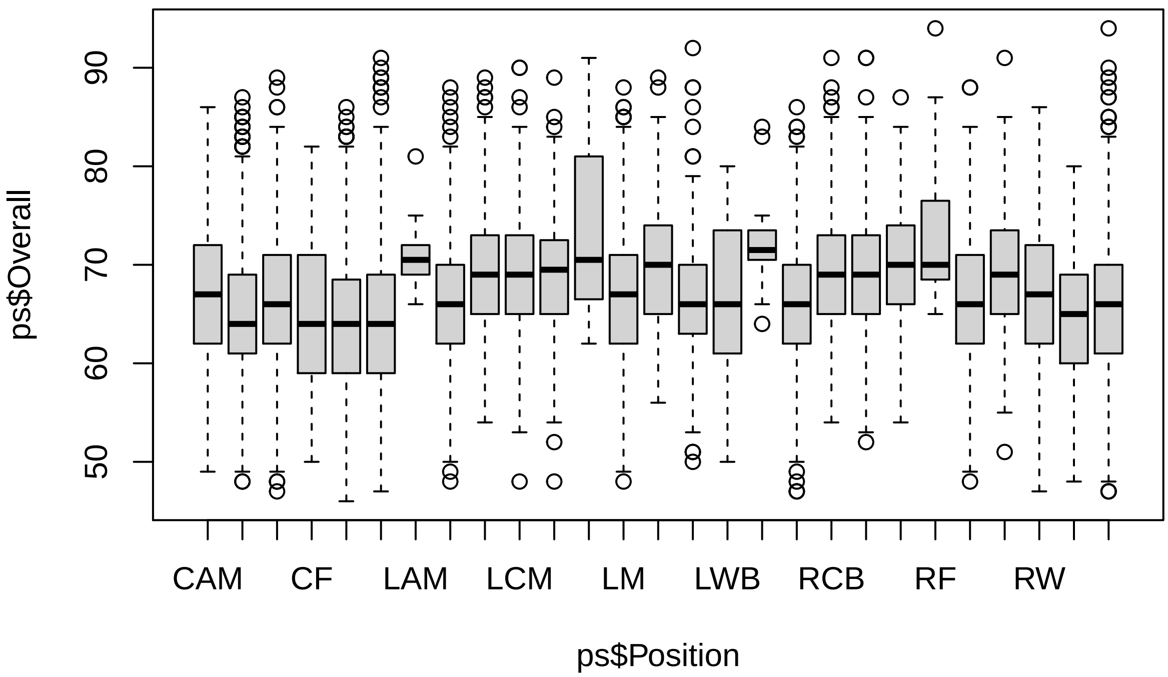
```r
#Clubs with worst players
Jersey %>% arrange((rating)) %>%
  top_n(10,-rating)
```

```
## # A tibble: 10 x 3
##    Jersey.Number     n rating
##            <int> <int>  <dbl>
##  1            59     4   57.5
##  2            51     7   58
##  3            65     4   58.5
##  4            49    17   59.6
##  5            82     5   59.8
##  6            61     6   59.8
##  7            46    28   60.3
##  8            36   136   60.6
##  9            35   162   61.0
## 10            54    11   61
```

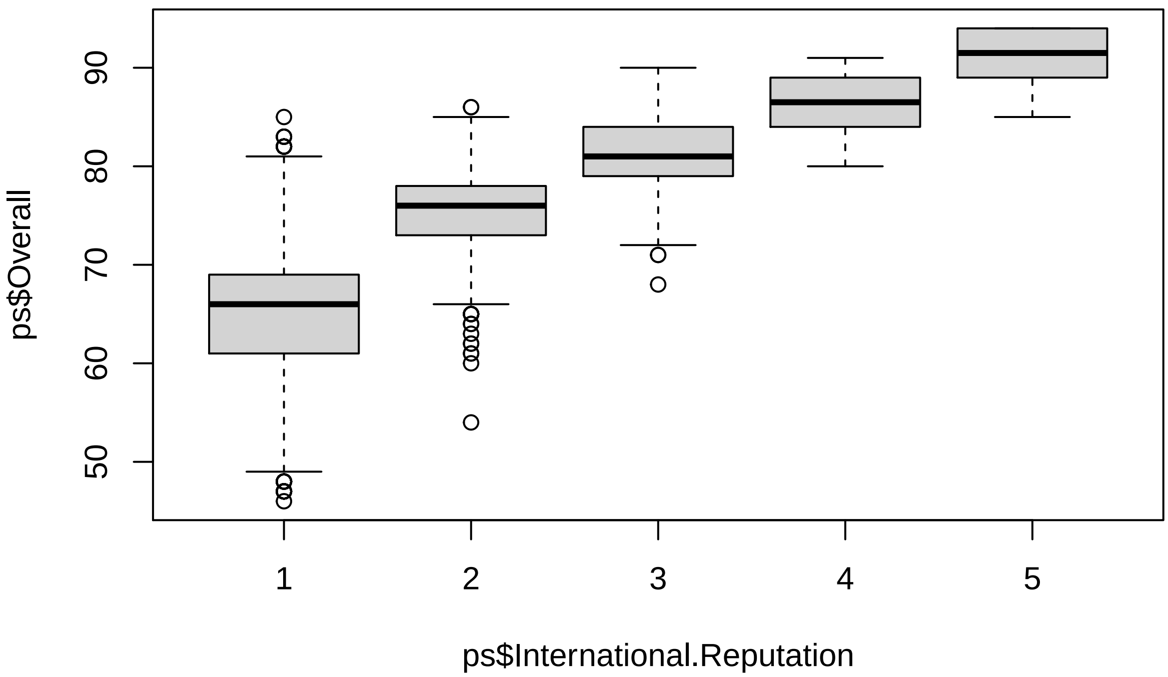
Jersey Numbers are a good option for training. However, due to non linearity regularization should be used and not the GLM method.

3.2.4 Simple As each of the simple attributes are so minimalistic boxplots will be used to quanitify them. Special and Work rate are difficult to quantify and therefore will be exluded from the model.

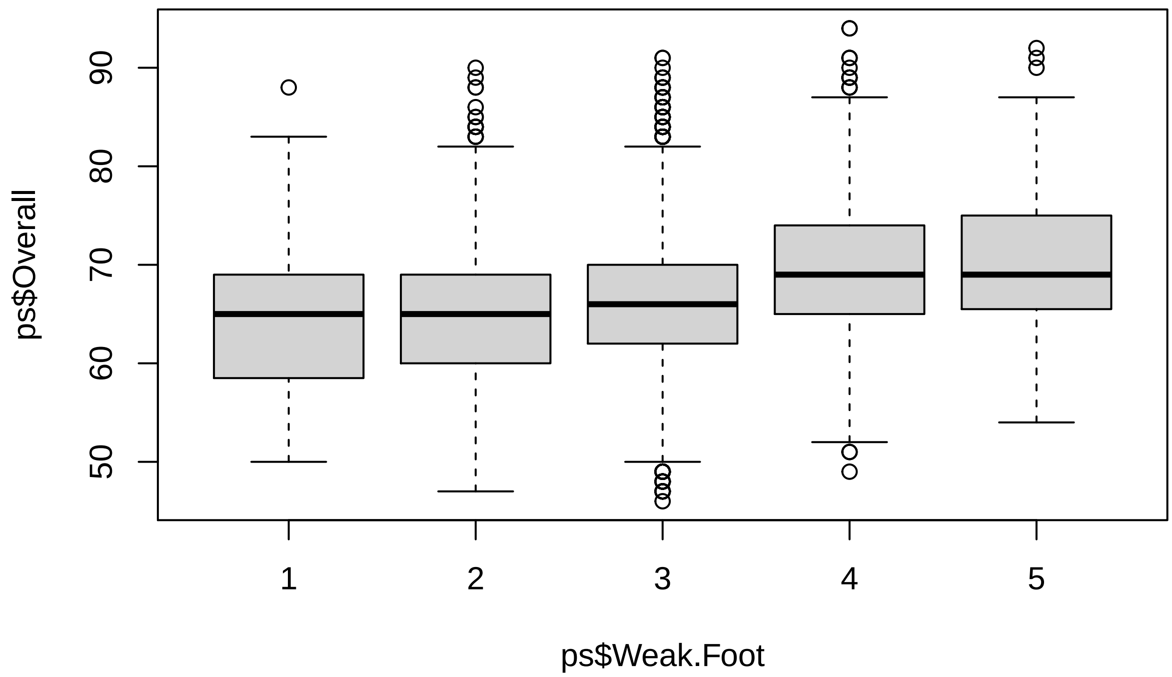
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(pscols)` instead of `pscols` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```



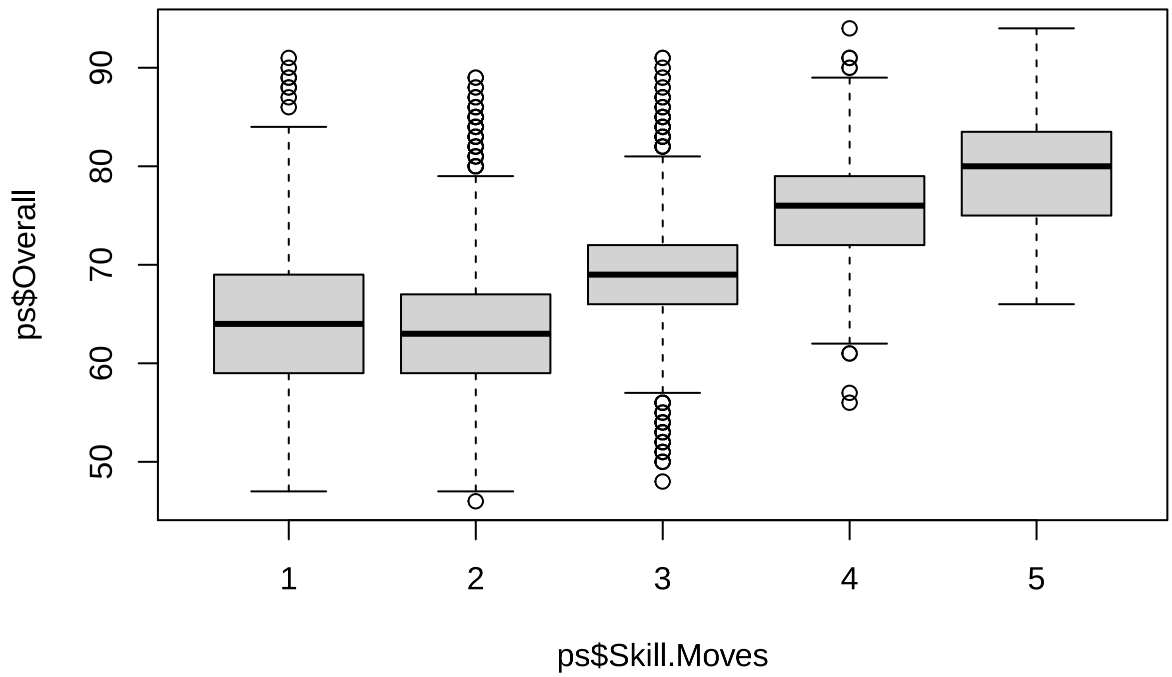
Position does not have a set effect and has high variability in overall making it not very useful. Therefore position was excluded from the model



International reputation has a good correlation to overall and was used.



Although weak foot does not have a strong correlation, a correlation exists. Therefore, weak foot will be used.

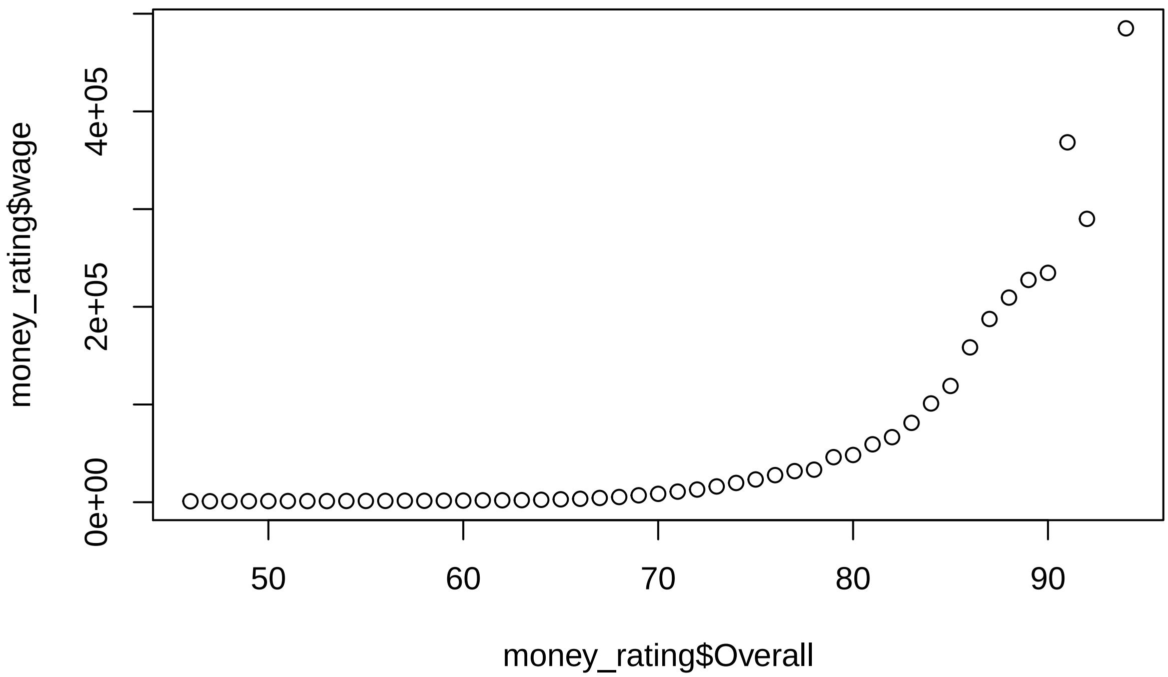


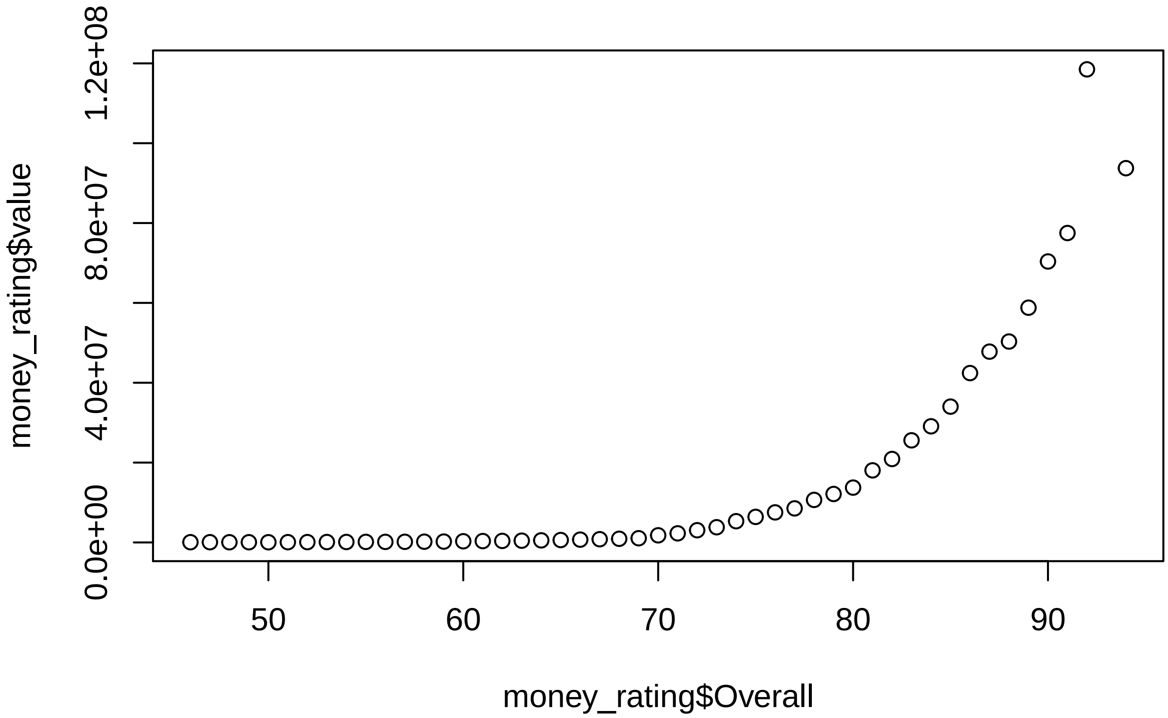
Skill moves have a decent correlation to overall and was used.

3.2.5 Monetary Money is a big driver in football therefore strong correlation to overall rating is expected.

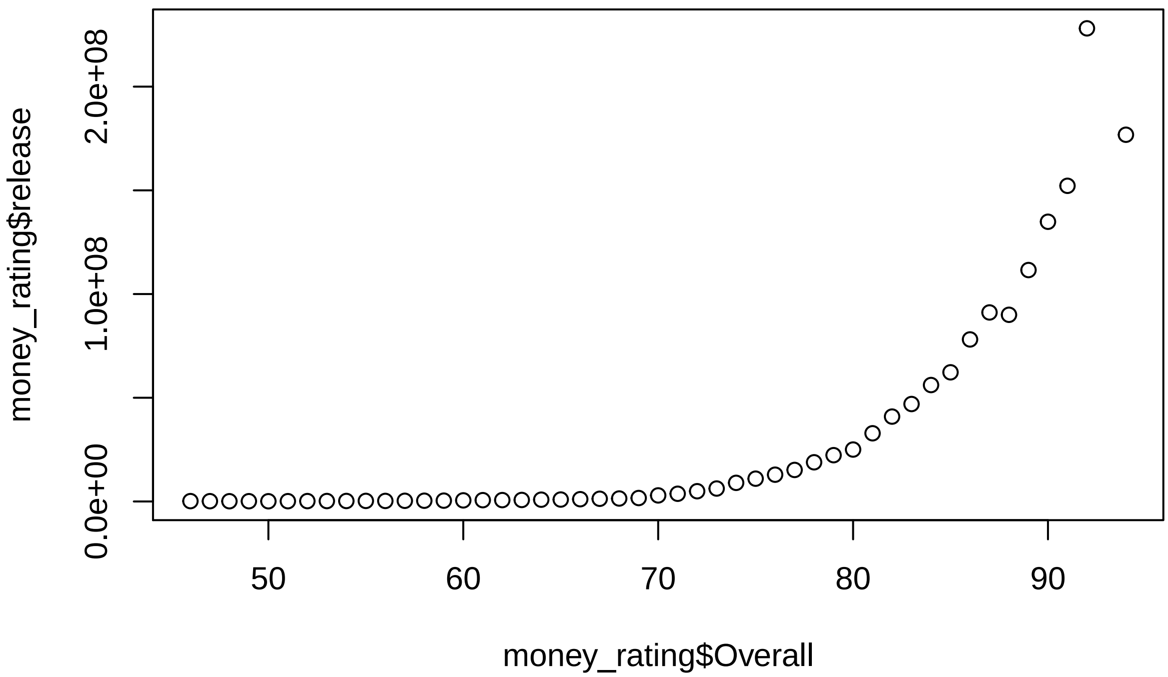
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(moncols)` instead of `moncols` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```


money_rating$wage
0e+00
2e+05
4e+05
50
60
70
80
90
money_rating$Overall



1.2e+08
8.0e+07
4.0e+07
0.0e+00
money_rating$value
50
60
70
80
90
money_rating$Overall



As expected all Monetary values have a large impact. specifically at higher overall ratings where there is a stronger correlation.. The monetary columns have leading characters that need to be removed so that they can be turned into numeric columns. This also includes converting from thousands and millions into base 1.

3.3 Predictive Model

Based on the analysis performed above there was sufficent evidence to sugget that a predictive model can be built on the available data. The analysis revealed that some of the columns needed changing to the formatting. This was performed before the model was built. The model was then built and trained as follows:

```
############################################################
# DATA PREPARATION
############################################################


#--- Columns to be used ---

header <- c("Name","Age","Overall","Weight","Value","Wage","Release.Clause","International.Reputation",

players <- players %>% select(header)
```

3.3.1 Data Split

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(header)` instead of `header` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
validation <- validation %>% select(header)
#--- players data set ---

#Fix monetary formatting


temp <- players %>% mutate(value_unit=(str_sub(Value,-1,-1)),value_euro=as.numeric(str_sub(Value,4,-2))
players <- temp %>% mutate(Value=ifelse(value_unit=="K",value_euro*1000,value_euro*1000000),Wage=ifelse
  select(header)

temp <- validation %>% mutate(value_unit=(str_sub(Value,-1,-1)),value_euro=as.numeric(str_sub(Value,4,-
validation <- temp %>% mutate(Value=ifelse(value_unit=="K",value_euro*1000,value_euro*1000000),Wage=ife
  select(header)

#Fix weight format

temp <- players %>% mutate(Weight=as.numeric(str_sub(Weight,1,-4)))
players <- temp %>% select(header)

temp <- validation %>% mutate(Weight=as.numeric(str_sub(Weight,1,-4)))
validation <- temp %>% select(header)

#Ensure N/As are set as 0
players[is.na(players)] <- 0
validation[is.na(validation)] <- 0

#--- View Data set ---

#players
head(players)
```

```
##                 Name Age Overall Weight     Value   Wage Release.Clause
## 1          L. Messi  31      94    159 110500000 565000      226500000
## 2 Cristiano Ronaldo  33      94    183  77000000 405000      127100000
## 3         Neymar Jr  26      92    150 118500000 290000      228100000
## 4            De Gea  27      91    168  72000000 260000      138600000
## 5      K. De Bruyne  27      91    154 102000000 355000      196400000
## 6         E. Hazard  27      91    163  93000000 340000      172100000
##   International.Reputation Weak.Foot Skill.Moves                Club
## 1                        5         4           4        FC Barcelona
## 2                        5         4           5            Juventus
## 3                        5         5           5 Paris Saint-Germain
## 4                        4         3           1   Manchester United
## 5                        4         5           4     Manchester City
## 6                        4         4           4             Chelsea
##   Jersey.Number
## 1            10
## 2             7
## 3            10
## 4             1
## 5             7
## 6            10
```

```r
nrow(players) #Should be 16331
```

```
## [1] 16331
```

```r
any(is.na(players))
```

```
## [1] FALSE
```

```r
# validation
head(validation)
```

```
##             Name Age Overall Weight    Value   Wage Release.Clause
## 1      D. GodÃn  32      90    172 44000000 125000       90200000
## 2   A. Griezmann  27      89    161 78000000 145000      165800000
## 3 J. RodrÃguez  26      88    172 69500000 315000              0
## 4     C. Eriksen  26      88    168 73500000 205000      141500000
## 5       Coutinho  26      88    150 69500000 340000      147700000
## 6    C. Immobile  28      87    187 52000000 115000       88400000
##   International.Reputation Weak.Foot Skill.Moves               Club
## 1                        3         3           2    AtlÃ©tico Madrid
## 2                        4         3           4    AtlÃ©tico Madrid
## 3                        4         3           4 FC Bayern MÃ¼nchen
## 4                        3         5           4  Tottenham Hotspur
## 5                        3         4           5       FC Barcelona
## 6                        3         4           3              Lazio
##   Jersey.Number
## 1            10
## 2             7
## 3            10
## 4            10
## 5             7
## 6            17
```

```r
nrow(validation) #Should be 1816
```

```
## [1] 1816
```

```r
any(is.na(validation))
```

```
## [1] FALSE
```

```r
#--- Clear Memory ---
rm(ages,clubs,hw,hw_rating,money,money_rating,nation,PlayerData,ps,readcsv,temp,Jersey)
```

```
## Warning in rm(ages, clubs, hw, hw_rating, money, money_rating, nation,
## PlayerData, : object 'readcsv' not found
```

```r
##########################################################
# LOAD INTO TEST AND TRAIN DATA
##########################################################

# Test set will be 10% of the dataset
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
test_index <- createDataPartition(y = players$Overall, times = 1, p = 0.1, list = FALSE)
train_set <- players[-test_index,]
test_set <- players[test_index,]
```

The monetary and weight columns were succesfully converted to numeric columns. Any NA's due to failed conversion were set to zero. The model was then split into a training set and 10% test set.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.00   62.00   66.00   66.24   71.00   94.00
```

The besic statistics were checked and align well to the statistics from the analysis section showing the split was a good split of the data and the training will be representitive.

3.3.2 Mean Model The purpose of the mean model is to set a benchmark for the rest of the predictive model

```r
#Running an RMSE on mean
RMSE_mu <- sqrt(mean((test_set$Overall-mu)^2))
RMSE_mu
```

```
## [1] 6.970715
```

```r
#Absolute Error
AbsError_mu <- mean(abs(mu-test_set$Overall))
AbsError_mu
```

```
## [1] 5.507522
```

This will now be the RMSE and absolute error to aim to improve on for the rest of the model components and the combined final model

3.3.3 Caret GLM Model The Caret GLM model used the caret train to fit a glm model to the data. This then informs a prediction based on testing data which is evaluated against the expecetd result.

```r
#--- AGE AND WEIGHT PREDICTION------------------------------#

#run a GLM
fit_aw <- train(Overall~Age+Weight,data=train_set, method = "glm")
```

```r
train_pred_aw <-predict(fit_aw,train_set)
pred_aw <-predict(fit_aw,test_set)

RMSE_aw <- sqrt(mean((test_set$Overall-pred_aw)^2))
RMSE_aw
```

```
## [1] 6.158537
```

```r
#Absolute Error
AbsError_aw <- mean(abs(pred_aw-test_set$Overall))
AbsError_aw
```

```
## [1] 4.766212
```

The physical attributes tested with the glm are age and weight. The final RMSE is slightly better than the mean. the absolute error is nearly 1 better than the mean indicating that this method works well on the middle range ratings but there are many outliers which are responsible for the high RMSE.

```r
#--- SKILL, WEAK FOOT AND REPUTATION PREDICTION ----------#

#run a GLM
fit_swr <- train(Overall~Skill.Moves+Weak.Foot+International.Reputation,data=train_set, method = "glm")

train_pred_swr <-predict(fit_swr,train_set)
pred_swr <-predict(fit_swr,test_set)

RMSE_swr <- sqrt(mean((test_set$Overall-pred_swr)^2))
RMSE_swr
```

```
## [1] 5.609622
```

```r
#Absolute Error
AbsError_swr <- mean(abs(pred_swr-test_set$Overall))
AbsError_swr
```

```
## [1] 4.428945
```

The simple attributes tested through glm are Skill moves, Weak foort and International Reputation. the strong correlation from reputation and skill moves allows this model to predict better than the physical attributes.

```r
#--- MONETARY PREDICTION -----------------------------#

#run a GLM
fit_mon <- train(Overall~Value + Wage + Release.Clause,data=train_set, method = "glm")

train_pred_mon <-predict(fit_mon,train_set)
pred_mon <-predict(fit_mon,test_set)

#Running an RMSE to view the error
RMSE_mon_glm <- sqrt(mean((test_set$Overall-pred_mon)^2))
RMSE_mon_glm
```

```
## [1] 5.560253
```

```r
#Absolute Error
AbsError_mon <- mean(abs(pred_mon-test_set$Overall))
AbsError_mon
```

```
## [1] 4.231967
```

As expected the monetary glm gave the most accurate result and lowest RMSE of the three Caret predition models. The monetary glm used Value, Wage and release clause to determine the predicted values.

3.3.4 Regularisation Regularisation is used to predict the general items club and jersey number.

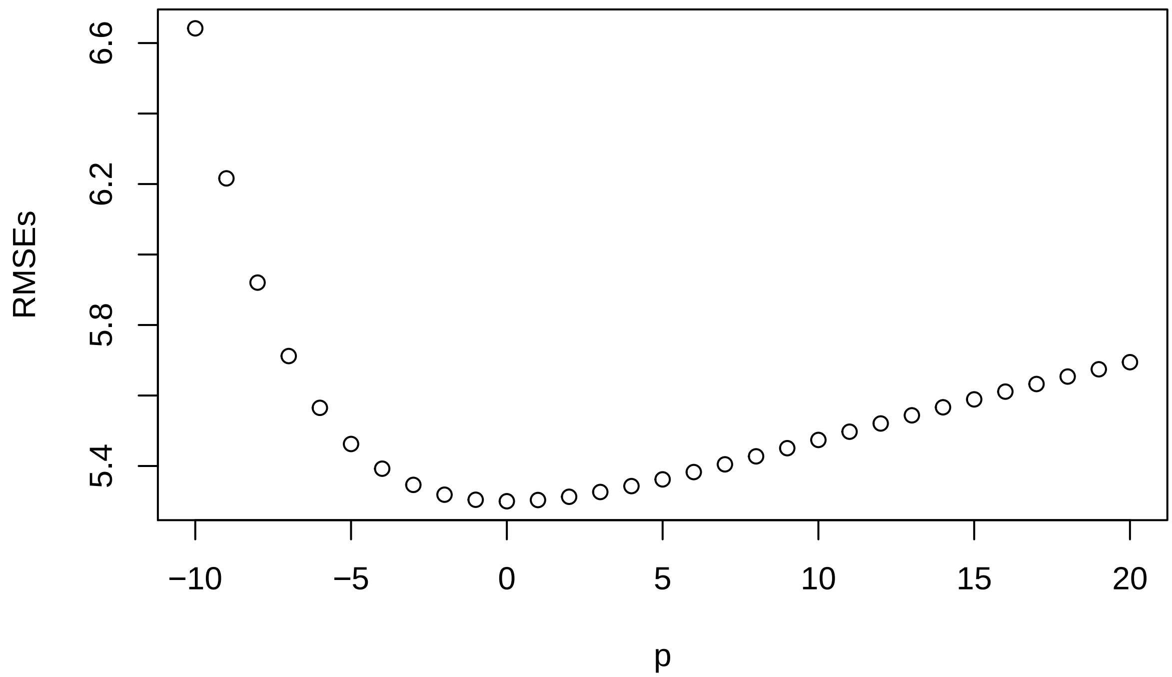
```
#--- Club ----------------------------------------------#

#Create the regularized for sum and mean
club <- train_set %>% group_by(Club) %>%
  summarize(n=n(),rsum=sum(Overall-mu))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

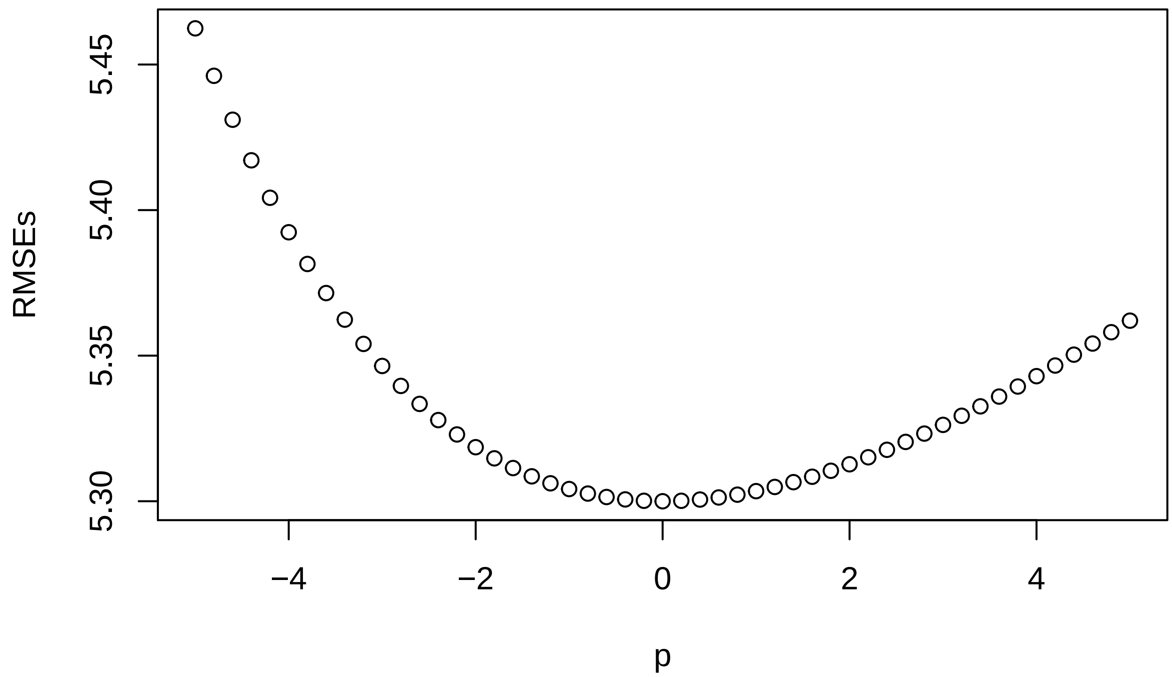
```
t_club <- test_set %>% left_join(club,by='Club')
train_club <- train_set %>% left_join(club,by='Club')

#Sample size regularization accounting for sample size n
t_club <- t_club %>% mutate(b_club=rsum/n)

RMSE_club_nopen <- sqrt(mean((test_set$Overall-(mu+t_club$b_club))^2))
RMSE_club_nopen
```

```
## [1] 5.657675
```

```
#regularization optimized with penalty term p
p <- seq(-10,20)

#sapply the terms
RMSEs <- sapply(p,function(p){
  club <- club %>% mutate(b_club=rsum/(n+p))
  train_club <- train_set %>% left_join(club,by='Club')
  sqrt(mean((train_set$Overall-(mu+train_club$b_club))^2))
})

#plot outputs
plot(p,RMSEs)
```




```
#Improve accuracy
p <- seq(-5,5,0.2)

#sapply the terms
RMSEs <- sapply(p,function(p){
  club <- club %>% mutate(b_club=rsum/(n+p))
  train_club <- train_set %>% left_join(club,by='Club')
  sqrt(mean((train_set$Overall-(mu+train_club$b_club))^2))
})

#plot outputs
plot(p,RMSEs)
```




```
p_club <- p[which.min(RMSEs)]

#final optimized output
t_club <- t_club %>% mutate(b_club=rsum/(n+p_club))

RMSE_club <- sqrt(mean((test_set$Overall-(mu+t_club$b_club))^2))
RMSE_club
```

```
## [1] 5.657675
```

```
#Absolute Error
AbsError_club <- mean(abs((mu+t_club$b_club)-test_set$Overall))
AbsError_club
```

```
## [1] 4.471578
```

The regularisation of clubs has an effect. Interestingly when optimised for a penalty term the best penalty is 0 indicating that the mean club score results in the best prediction.

```
#--- Jersey Number ----------------------------------------------#

#Create the regularized for sum and mean
Jersey <- train_set %>% group_by(Jersey.Number) %>%
  summarize(n=n(),rsum=sum(Overall-mu))
```

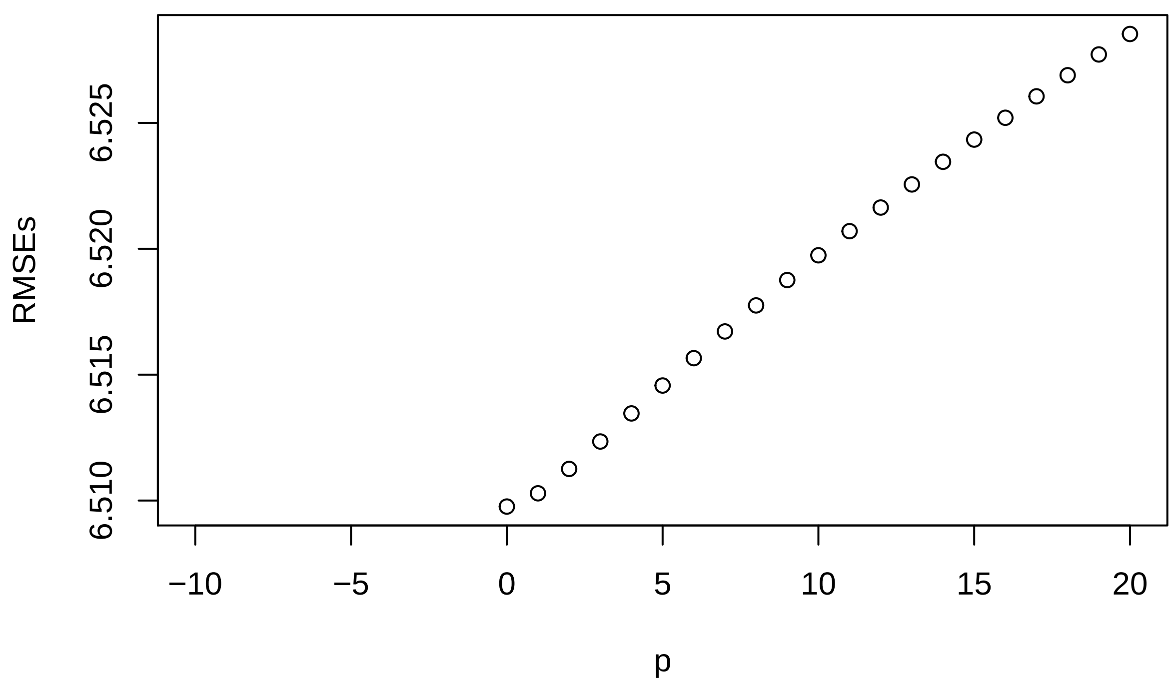
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
t_jersey <- test_set %>% left_join(Jersey,by='Jersey.Number')
train_jn <- train_set %>% left_join(Jersey,by='Jersey.Number')
```

```r
#Sample size regularization accounting for sample size n
t_jersey <- t_jersey %>% mutate(b_jersey=rsum/n)

RMSE_jn_nopen <- sqrt(mean((test_set$Overall-(mu+t_jersey$b_jersey))^2))
RMSE_jn_nopen
```

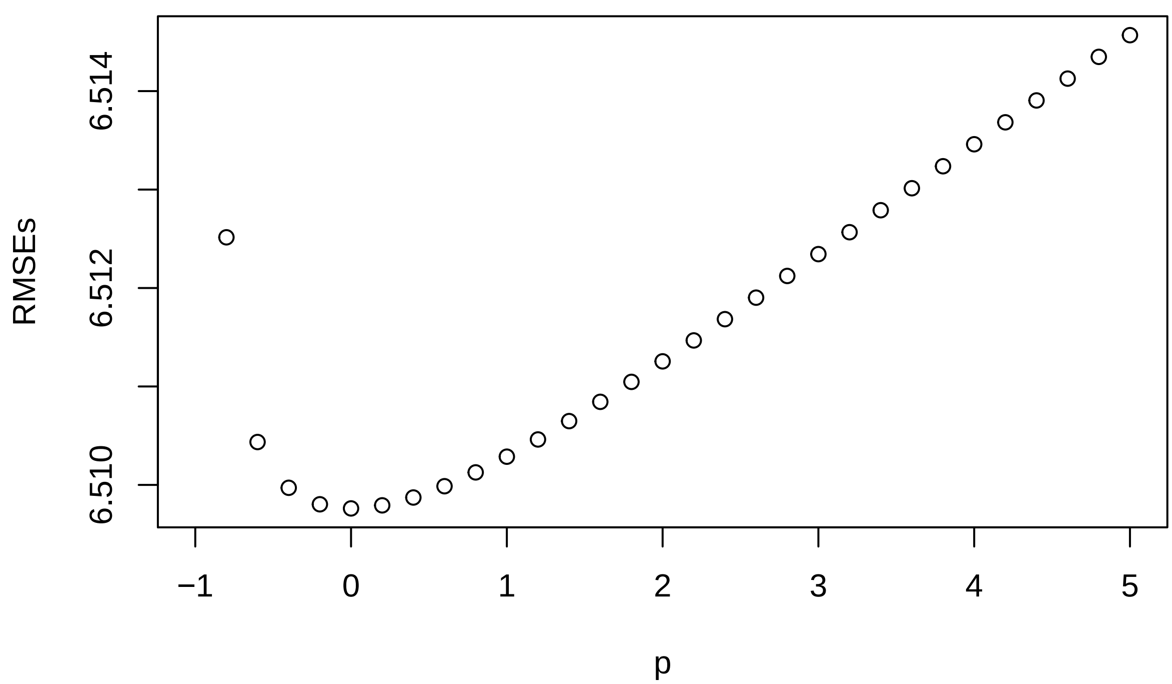
```
## [1] 6.576452
```

```r
#regularization optimized with penalty term p
p <- seq(-10,20)

#sapply the terms
RMSEs <- sapply(p,function(p){
  Jersey <- Jersey %>% mutate(b_jersey=rsum/(n+p))
  train_jn <- train_set %>% left_join(Jersey,by='Jersey.Number')
  sqrt(mean((train_set$Overall-(mu+train_jn$b_jersey))^2))
})

#plot outputs
plot(p,RMSEs)
```




```r
#Improve accuracy
p <- seq(-1,5,0.2)

#sapply the terms
RMSEs <- sapply(p,function(p){
```

```
  Jersey <- Jersey %>% mutate(b_jersey=rsum/(n+p))
  train_jn <- train_set %>% left_join(Jersey,by='Jersey.Number')
  sqrt(mean((train_set$Overall-(mu+train_jn$b_jersey))^2))
})

#plot outputs
plot(p,RMSEs)
```




```
p_jersey <- p[which.min(RMSEs)]

#final optimized output
t_jersey <- t_jersey %>% mutate(b_jersey=rsum/(n+p_jersey))

RMSE_jn <- sqrt(mean((test_set$Overall-(mu+t_jersey$b_jersey))^2))
RMSE_jn
```

```
## [1] 6.576452
```

```
#Absolute Error
AbsError_jn <- mean(abs((mu+t_jersey$b_jersey)-test_set$Overall))
AbsError_jn
```

```
## [1] 5.177001
```

Again the optimised penalty term is 0. This is likely to change in the combined model. However, the RMSE of jersey is only marginally better than the mean. This model is not the best one.

```r
#Create b_sum to hold all previous prediction values for tuning
b_sum = train_pred_mon

#Combine monetary and age weight values (Note these already contain mu)
train_b_sum = (train_pred_mon + train_pred_aw + train_pred_swr)/3
b_sum = (pred_mon + pred_aw + pred_swr)/3

#Using a more powerful computer to run all
fit_glm <- train(Overall~Age+Weight+Skill.Moves+Weak.Foot+International.Reputation+Value + Wage + Releas

train_b_sum <-predict(fit_glm,train_set)
b_sum <-predict(fit_glm,test_set)

#Results of the glm combined
RMSE_glm <- sqrt(mean((test_set$Overall-b_sum)^2))
RMSE_glm
```

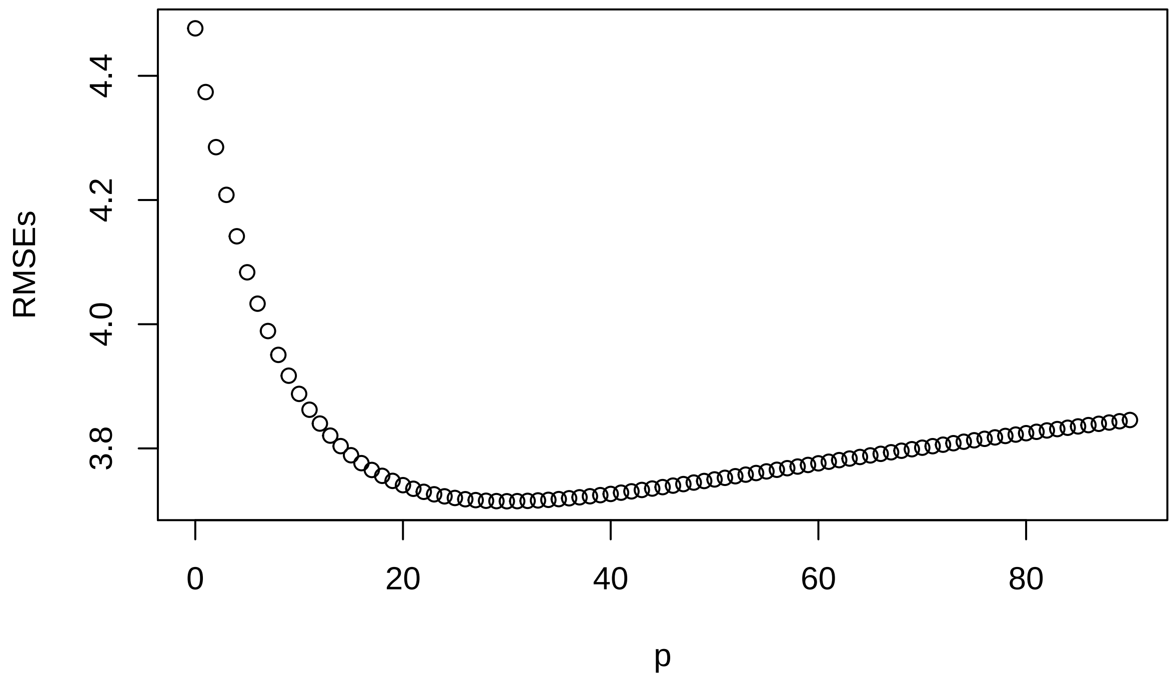
3.3.5 Combined

```
## [1] 4.36562
```

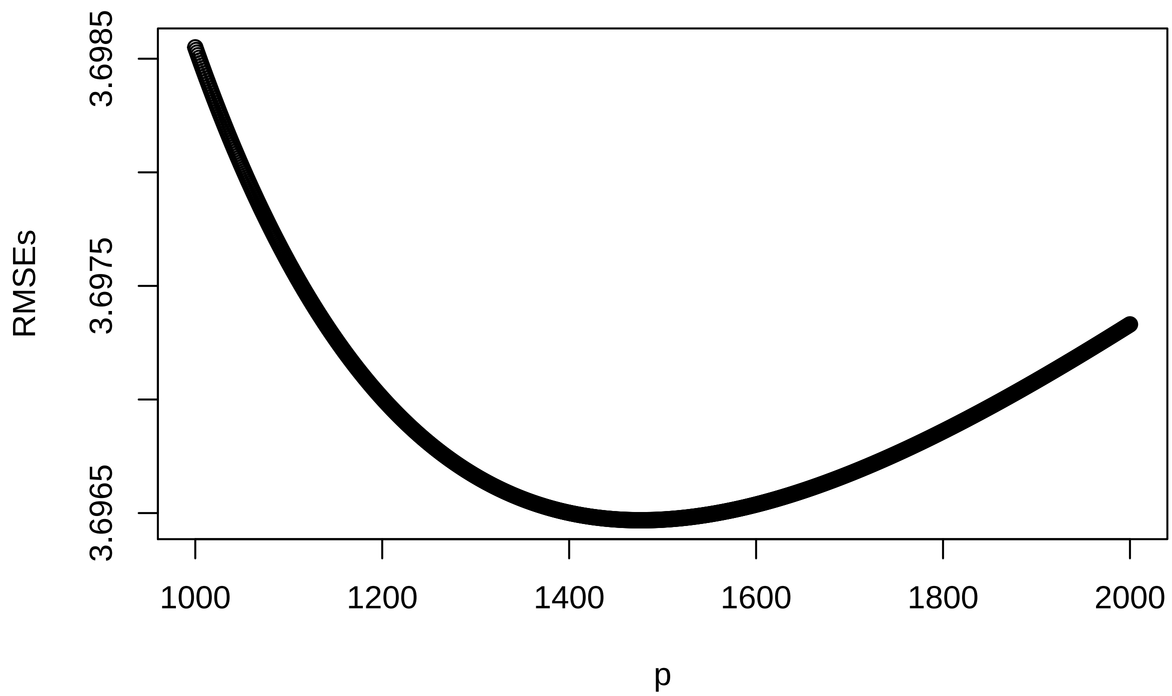
```r
#Absolute Error
AbsError_glm <- mean(abs(b_sum-test_set$Overall))
AbsError_glm
```

```
## [1] 3.244178
```

```r
#COMBINE CLUB INTO MODEL WITH TUNING

#regularization optimized with penalty term p
p <- seq(0,90)

#sapply the terms
RMSEs <- sapply(p,function(p){
  club <- club %>% mutate(b_club=rsum/(n+p))
  train_club <- train_set %>% left_join(club,by='Club')
  sqrt(mean((train_set$Overall-(train_b_sum+train_club$b_club))^2))
})

#plot outputs
plot(p,RMSEs)
```




```
p_club <- p[which.min(RMSEs)]

#final optimized output
t_club <- t_club %>% mutate(b_club=rsum/(n+p_club))
train_club <- train_club %>% mutate(b_club=rsum/(n+p_club))

b_sum=b_sum+t_club$b_club
train_b_sum=train_b_sum+train_club$b_club

#COMBINE Jersey Number INTO MODEL WITH TUNING

#regularization optimized with penalty term p
p <- seq(1000,2000)

#sapply the terms
RMSEs <- sapply(p,function(p){
  Jersey <- Jersey %>% mutate(b_jersey=rsum/(n+p))
  train_jn <- train_set %>% left_join(Jersey,by='Jersey.Number')
  sqrt(mean((train_set$Overall-(train_b_sum+train_jn$b_jersey))^2))
})

#As the sequence gets larger the accuracy improvement is less effective therefore will not be used.

#plot outputs
plot(p,RMSEs)
```




```r
p_jersey <- p[which.min(RMSEs)]

#final optimized output
t_jersey <- t_jersey %>% mutate(b_jersey=rsum/(n+p_jersey))
train_jn <- train_jn %>% mutate(b_jersey=rsum/(n+p_jersey))

b_sum=b_sum+t_jersey$b_jersey
train_b_sum=train_b_sum+train_jn$b_jersey

RMSE_combined <- sqrt(mean((test_set$Overall-b_sum)^2))
RMSE_combined
```

```
## [1] 4.111078
```

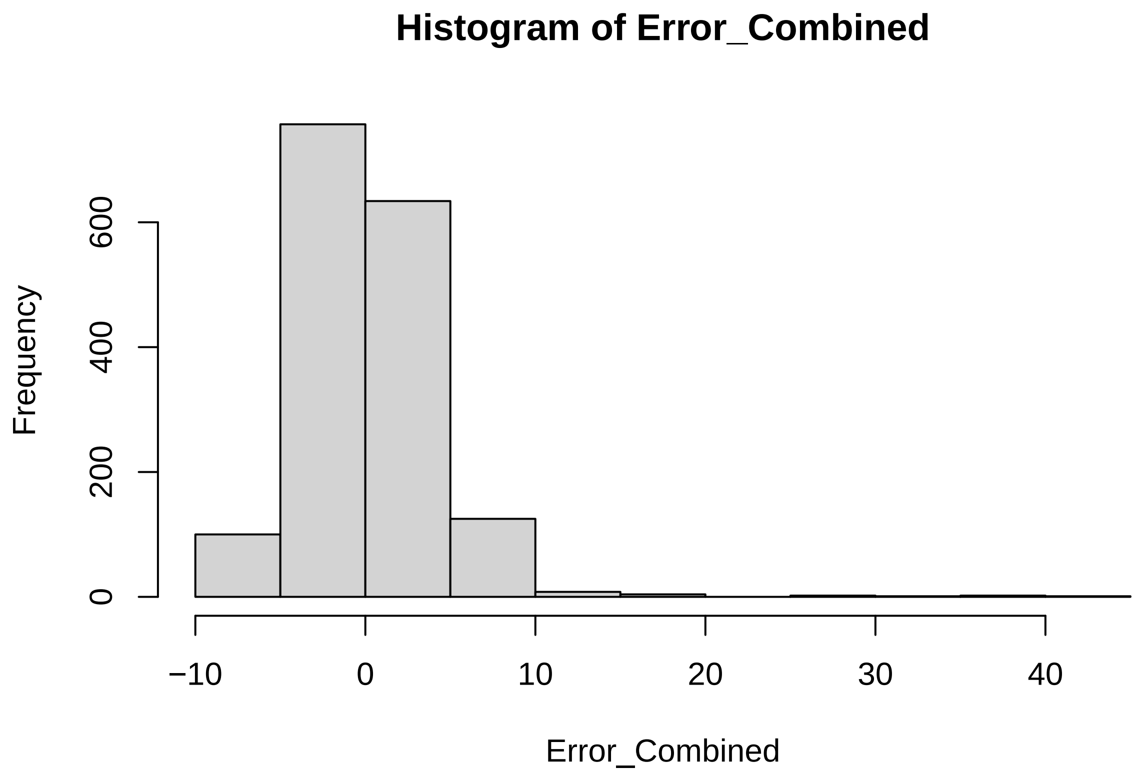
```r
#Combined Error
AbsError_combined <- mean(abs(b_sum-test_set$Overall))
AbsError_combined
```

```
## [1] 2.814999
```

```r
#Accuracy of prediction
acc <- round(b_sum,0) == test_set$Overall
mean(acc)*100
```

```
## [1] 13.15789
```

```r
#Distribution of error
Error_Combined <-b_sum-test_set$Overall
hist(Error_Combined)
```



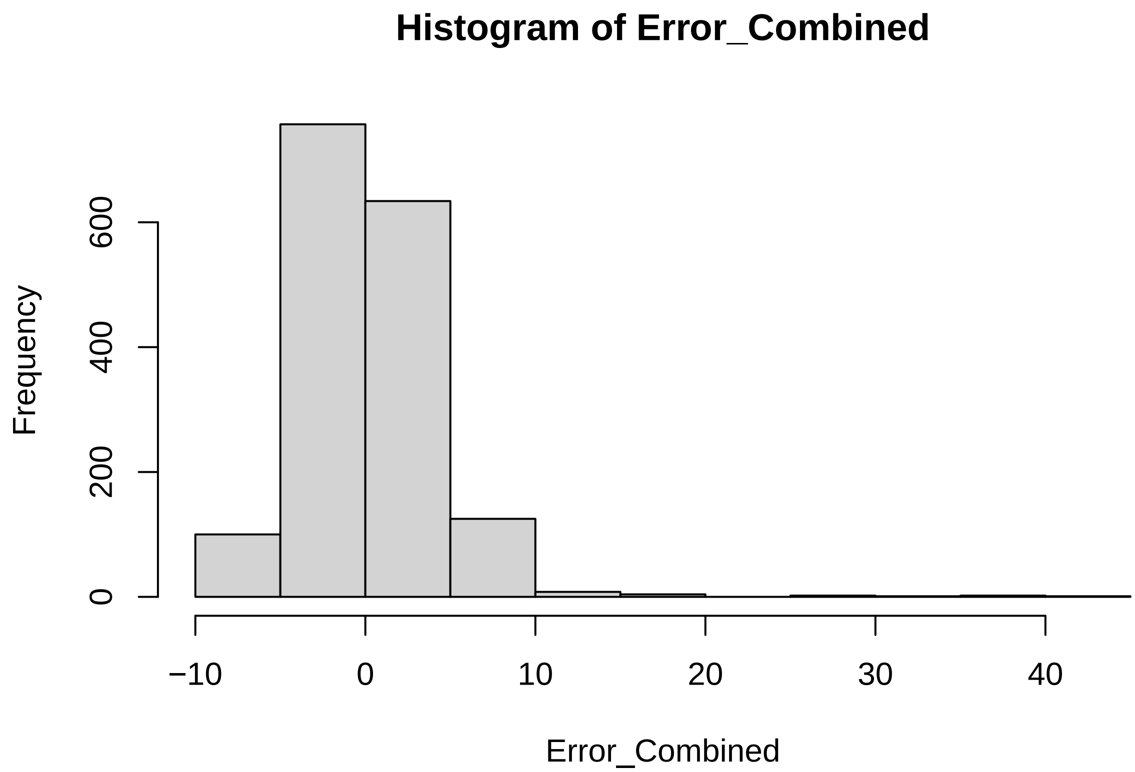
As predicted the tuning optimisation for both clubs and jerseys was required in the final model. The combined RMSE and absolute error show a substantial and adequate improvement from the mean model results. This is ideal and shows that the models are best when applied together.

```
#Accuracy of prediction
acc <- round(b_sum,0) == test_set$Overall
mean(acc)*100
```

3.3.6 Results

```
## [1] 13.15789
```

When testing the accuracy of the final model. this is done by testing to see what portion of results lie within +-0.5 of the actual overall rating. Therefore at a 99% test the accuracy of the model is ~13%. This can be expecetd as the mean absolute error is 2.8 but RMSE of 4 meaning there are more outliers as shown by the histogram below:



The final results from each model and the combined model are:

Method	RMSE	Error
Mean Prediction	6.970715	5.507522
Physical Prediction	6.158537	4.766212
Club Prediction	5.657675	4.471578
Jersey Number Prediction	6.576452	5.177001
Simple Attributes Prediction	5.609622	4.428945
Monetary Prediction	5.560253	4.231967
Combined GLM Results	4.365620	3.244178
Combined Results	4.111078	2.814999

3.3 Validation

```
val_mon <-predict(fit_mon,validation)
val_aw <-predict(fit_aw,validation)
val_swr <-predict(fit_swr,validation)


#Club regularization
club <- train_set %>% group_by(Club) %>%
  summarize(n=n(),rsum=sum(Overall-mu))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
val_club <- validation %>% left_join(club,by='Club')
val_club <- val_club %>% mutate(b_club=rsum/(n+p_club))

#Jersey Number regularization
Jersey <- train_set %>% group_by(Jersey.Number) %>%
  summarize(n=n(),rsum=sum(Overall-mu))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
val_jn <- validation %>% left_join(Jersey,by='Jersey.Number')
val_jn <- val_club %>% mutate(b_jersey=rsum/(n+p_jersey))

#Combine results
val_b_sum <-  (val_mon + val_aw + val_swr)/3

#With a more powerful computer
val_glm <-predict(fit_glm,validation)
val_b_sum <- val_glm

val_b_sum <- val_b_sum+val_club$b_club+val_jn$b_jersey

#Final result
RMSE_Final <- sqrt(mean((validation$Overall-val_b_sum)^2))
RMSE_Final
```

```
## [1] 4.03834
```

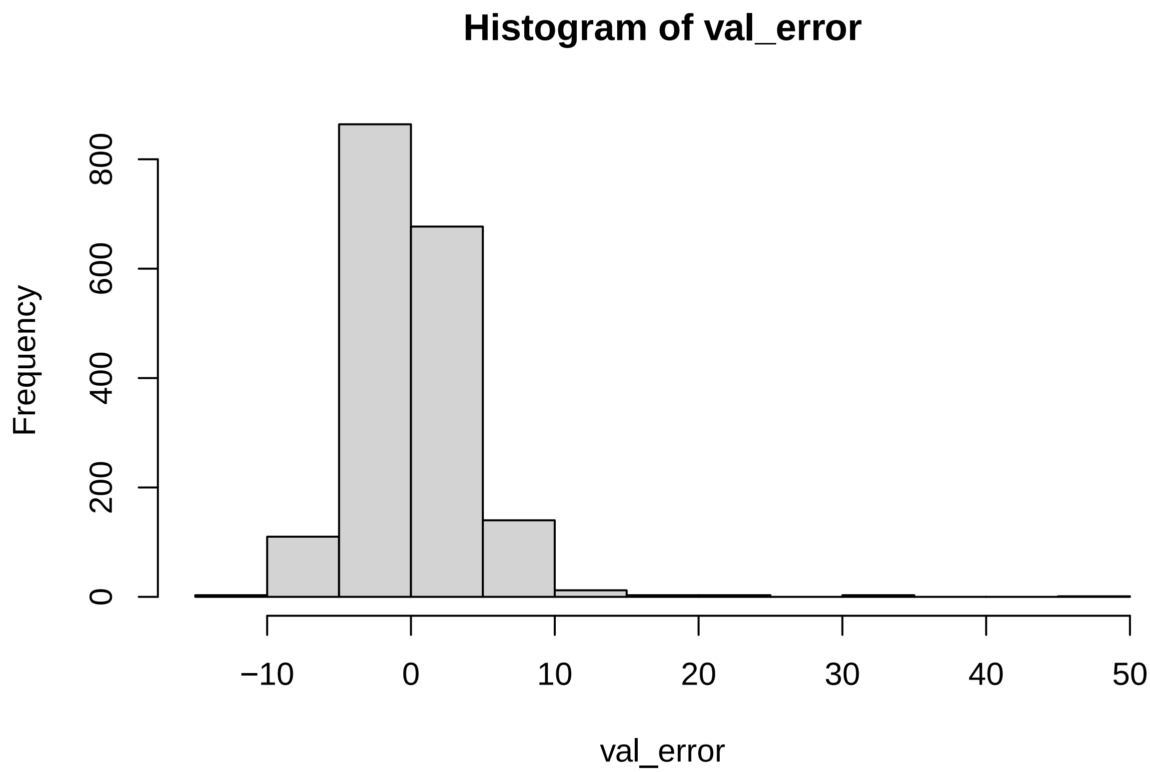
```
#Final Error
AbsError_Final <- mean(abs(val_b_sum-validation$Overall))
AbsError_Final
```

```
## [1] 2.877896
```

```
#Accuracy of prediction
acc <- round(val_b_sum,0) == validation$Overall
mean(acc)*100
```

```
## [1] 11.28855
```

```
#Distribution of error
val_error <- val_b_sum-validation$Overall
hist(val_error)
```



The final validation was performed on the validation set using the individual models and combined to create the final result. As expected the final result is not as good as the trained model. However the final result is still satisfactory. The accuracy is drops to ~11% on a 99% confidence rating. However, this can once again be attributed by the large number of outliers.

Method	RMSE	Error
Mean Prediction	6.970715	5.507522
Physical Prediction	6.158537	4.766212
Club Prediction	5.657675	4.471578
Jersey Number Prediction	6.576452	5.177001
Simple Attributes Prediction	5.609622	4.428945
Monetary Prediction	5.560253	4.231967
Combined GLM Results	4.365620	3.244178
Combined Results	4.111078	2.814999
Validation Results	4.038340	2.877896

Therefore the final results can be seen in the table above.

Conclusion

The project goal was to build a model that would predict the overall FIFA ratings of soccer players using only available data and simple to estimate data. The dataset of more than 18000 players and 89 variables was a very good dataset. There were only 60 missing values (0.3%) and only 5 variables required engineering to get into a correct format. Of the 89 variables 17 variables were classified into 4 groups for the model. Upon analysis of the data 10 were chosen for the model. The grouped models had success however, it was the final combined model that was tuned through regularisation which was substantially better than the others. The final results are as follows:

Method	RMSE	Error
Mean Prediction	6.970715	5.507522
Physical Prediction	6.158537	4.766212
Club Prediction	5.657675	4.471578
Jersey Number Prediction	6.576452	5.177001
Simple Attributes Prediction	5.609622	4.428945
Monetary Prediction	5.560253	4.231967
Combined GLM Results	4.365620	3.244178
Combined Results	4.111078	2.814999
Validation Results	4.038340	2.877896

There was still a large error involved with this kind of model. This is generally due to the inability of the model to predict outliers as can be shown by the high RMSE values. The distribution of errors and the lower mean absolute error show that for values closer to the means the model predicts well. Ultimately this model will give a good idea of the players overall ability but will only serve to inform on the rating. More complex classifiers and building the combined model as a single piece may lead to better results this can be shown when combining all the glm models together. The actual rating will require the complex variables that FIFA uses to give the overall score.

Note: Without a powerful computer where the full glm suite can be run the validation RMSE is 4.373 with a mean absolute error of 3.470 and accuracy of 9.4%