NBA Scoring Analysis

Peter Mulhall

12/31/2020

1.0 Introduction

This project is an attempt to use machine learning to predict the scores of NBA games with the goal of using historical statistics to train models in R to predict the number of points a team might score.

1.1 Dataset Description

This dataset represents all NBA games played from 2003 to 2020, and was supplied by Kaggle. It is by no means comprehensive in the statistics which have been captured, but provides all necessary and relevant data points for our purposes of analysis for this project.

Below is the head and summary of the dataset for familiarization:

```
str(games)
```

```
## 'data.frame':    23421 obs. of  23 variables:
##  $ GAME_DATE_EST   : chr  "19/12/2020" "19/12/2020" "19/12/2020" "18/12/2020" ...
##  $ GAME_ID         : int  12000047 12000048 12000049 12000039 12000040 12000042 12000041 12000043 120
##  $ GAME_STATUS_TEXT: chr  "Final" "Final" "Final" "Final" ...
##  $ HOME_TEAM_ID    : Factor w/ 30 levels "ATL","BKN","BOS",..: 22 30 15 12 28 20 3 19 21 8 ...
##  $ VISITOR_TEAM_ID : Factor w/ 30 levels "ATL","BKN","BOS",..: 4 9 1 23 16 6 2 17 5 25 ...
##  $ SEASON          : Factor w/ 18 levels "2003","2004",..: 18 18 18 18 18 18 18 18 18 18 ...
##  $ TEAM_ID_home    : int  1610612753 1610612764 1610612763 1610612754 1610612761 1610612752 161061273
##  $ PTS_home        : int  120 99 116 107 105 119 89 127 103 129 ...
##  $ FG_PCT_home     : num  0.433 0.427 0.4 0.371 0.38 0.513 0.348 0.512 0.411 0.474 ...
##  $ FT_PCT_home     : num  0.792 0.625 0.744 0.692 0.737 0.788 0.81 0.614 0.737 0.778 ...
##  $ FG3_PCT_home    : num  0.425 0.295 0.396 0.262 0.356 0.517 0.178 0.364 0.395 0.409 ...
##  $ AST_home        : int  23 24 21 19 27 27 18 25 21 30 ...
##  $ REB_home        : int  50 45 43 45 37 41 48 51 51 53 ...
##  $ TEAM_ID_away    : int  1610612766 1610612765 1610612737 1610612755 1610612748 1610612739 161061275
##  $ PTS_away        : int  117 96 117 113 117 83 113 113 105 96 ...
##  $ FG_PCT_away     : num  0.444 0.402 0.422 0.533 0.534 0.395 0.432 0.422 0.424 0.371 ...
##  $ FT_PCT_away     : num  0.864 0.647 0.837 0.629 0.741 0.611 0.778 0.9 0.588 0.87 ...
##  $ FG3_PCT_away    : num  0.439 0.326 0.297 0.355 0.514 0.387 0.457 0.25 0.268 0.303 ...
##  $ AST_away        : int  21 18 24 23 30 20 26 21 27 17 ...
##  $ REB_away        : int  52 51 47 48 51 35 53 44 56 42 ...
##  $ HOME_TEAM_WINS  : Factor w/ 2 levels "LOSS","WIN": 2 2 1 1 1 2 1 2 1 2 ...
##  $ PTS_total       : int  237 195 233 220 222 202 202 240 208 225 ...
##  $ PTS_diff        : int  3 3 -1 -6 -12 36 -24 14 -2 33 ...
```

```
summary(games)
```

```
##  GAME_DATE_EST         GAME_ID         GAME_STATUS_TEXT    HOME_TEAM_ID
##  Length:23421       Min.   :10300001   Length:23421       LAL    :  858
##  Class :character   1st Qu.:20600663   Class :character   MIA    :  844
##  Mode  :character   Median :21100368   Mode  :character   SAS    :  836
##                     Mean   :21653578                      BOS    :  834
##                     3rd Qu.:21600314                      CLE    :  818
##                     Max.   :51900111                      DET    :  800
##                                                           (Other):18431
##  VISITOR_TEAM_ID     SEASON       TEAM_ID_home          PTS_home
##  SAS    :  830   2005   : 1432   Min.   :1.611e+09   Min.   : 36.0
##  MIA    :  828   2013   : 1427   1st Qu.:1.611e+09   1st Qu.: 93.0
##  BOS    :  820   2008   : 1425   Median :1.611e+09   Median :102.0
##  DEN    :  804   2009   : 1424   Mean   :1.611e+09   Mean   :102.3
##  DAL    :  802   2010   : 1422   3rd Qu.:1.611e+09   3rd Qu.:111.0
##  GSW    :  800   2012   : 1420   Max.   :1.611e+09   Max.   :168.0
##  (Other):18537   (Other):14871
##   FG_PCT_home      FT_PCT_home      FG3_PCT_home       AST_home
##  Min.   :0.2500   Min.   :0.1430   Min.   :0.0000   Min.   : 6.00
##  1st Qu.:0.4200   1st Qu.:0.6960   1st Qu.:0.2830   1st Qu.:19.00
##  Median :0.4590   Median :0.7650   Median :0.3550   Median :22.00
##  Mean   :0.4599   Mean   :0.7582   Mean   :0.3555   Mean   :22.54
##  3rd Qu.:0.5000   3rd Qu.:0.8260   3rd Qu.:0.4290   3rd Qu.:26.00
##  Max.   :0.6840   Max.   :1.0000   Max.   :1.0000   Max.   :47.00
##
##     REB_home     TEAM_ID_away          PTS_away       FG_PCT_away
##  Min.   :15.0   Min.   :1.611e+09   Min.   : 33.00   Min.   :0.2440
##  1st Qu.:39.0   1st Qu.:1.611e+09   1st Qu.: 90.00   1st Qu.:0.4110
##  Median :43.0   Median :1.611e+09   Median : 99.00   Median :0.4470
##  Mean   :43.2   Mean   :1.611e+09   Mean   : 99.34   Mean   :0.4483
##  3rd Qu.:48.0   3rd Qu.:1.611e+09   3rd Qu.:108.00   3rd Qu.:0.4860
##  Max.   :72.0   Max.   :1.611e+09   Max.   :168.00   Max.   :0.6740
##
##   FT_PCT_away      FG3_PCT_away       AST_away        REB_away
##  Min.   :0.1430   Min.   :0.0000   Min.   : 4.00   Min.   :19.00
##  1st Qu.:0.6920   1st Qu.:0.2760   1st Qu.:18.00   1st Qu.:37.00
##  Median :0.7620   Median :0.3490   Median :21.00   Median :42.00
##  Mean   :0.7564   Mean   :0.3488   Mean   :21.14   Mean   :41.88
##  3rd Qu.:0.8280   3rd Qu.:0.4210   3rd Qu.:24.00   3rd Qu.:46.00
##  Max.   :1.0000   Max.   :1.0000   Max.   :46.00   Max.   :81.00
##
##  HOME_TEAM_WINS   PTS_total        PTS_diff
##  LOSS: 9520     Min.   : 69.0   Min.   :-58.000
##  WIN :13901     1st Qu.:186.0   1st Qu.: -6.000
##                 Median :201.0   Median :  4.000
##                 Mean   :201.6   Mean   :  2.947
##                 3rd Qu.:216.0   3rd Qu.: 11.000
##                 Max.   :329.0   Max.   : 61.000
##
```

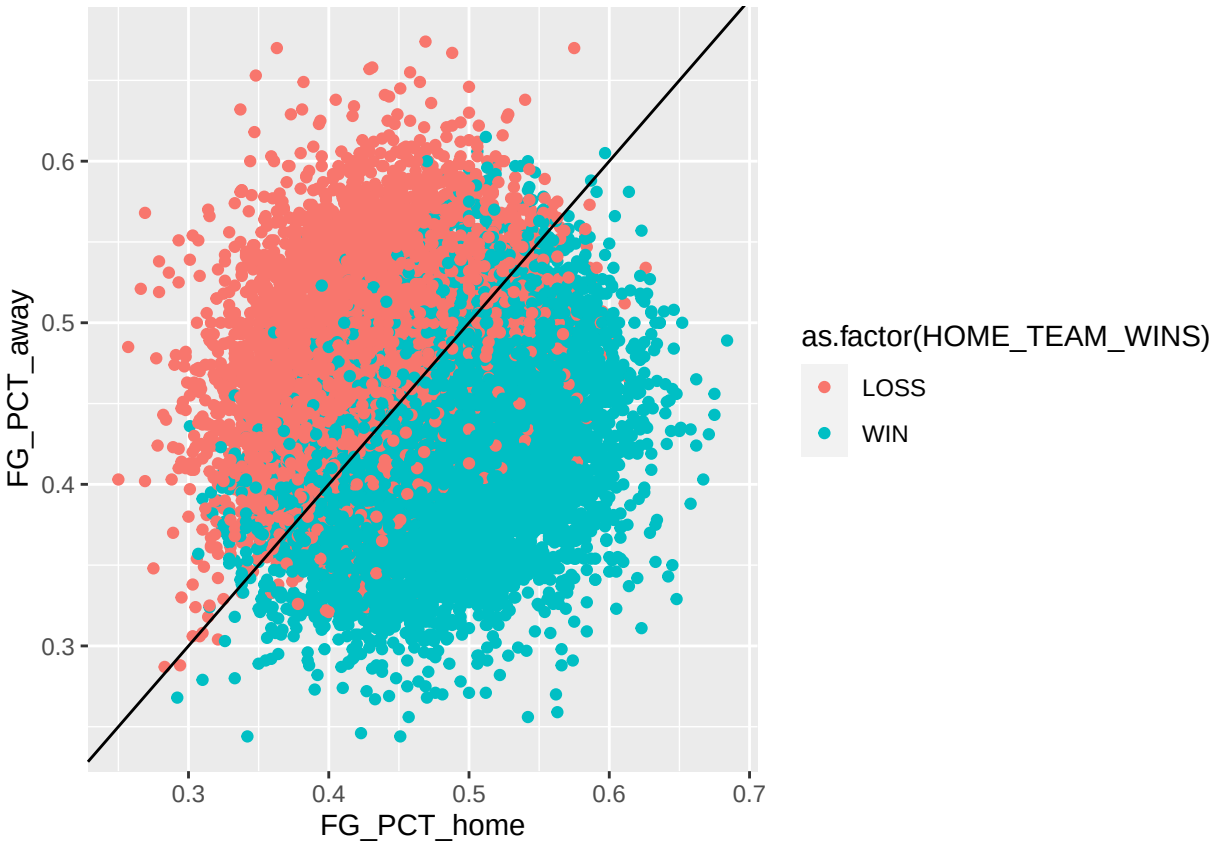
2.0 Methods and Analysis

2.1 Data Cleaning

Original data set had 99 games missing scoring information that needed to be removed before analyzing. Also joined in Team name abbreviations as original dataset only included id numbers. Mutated dataset to include point differentials and factor for win/loss information.

2.2 Data Exploration and Visualization

```
ggplot(games, aes(x=FG_PCT_home, y=FG_PCT_away, color=as.factor(HOME_TEAM_WINS))) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1)
```



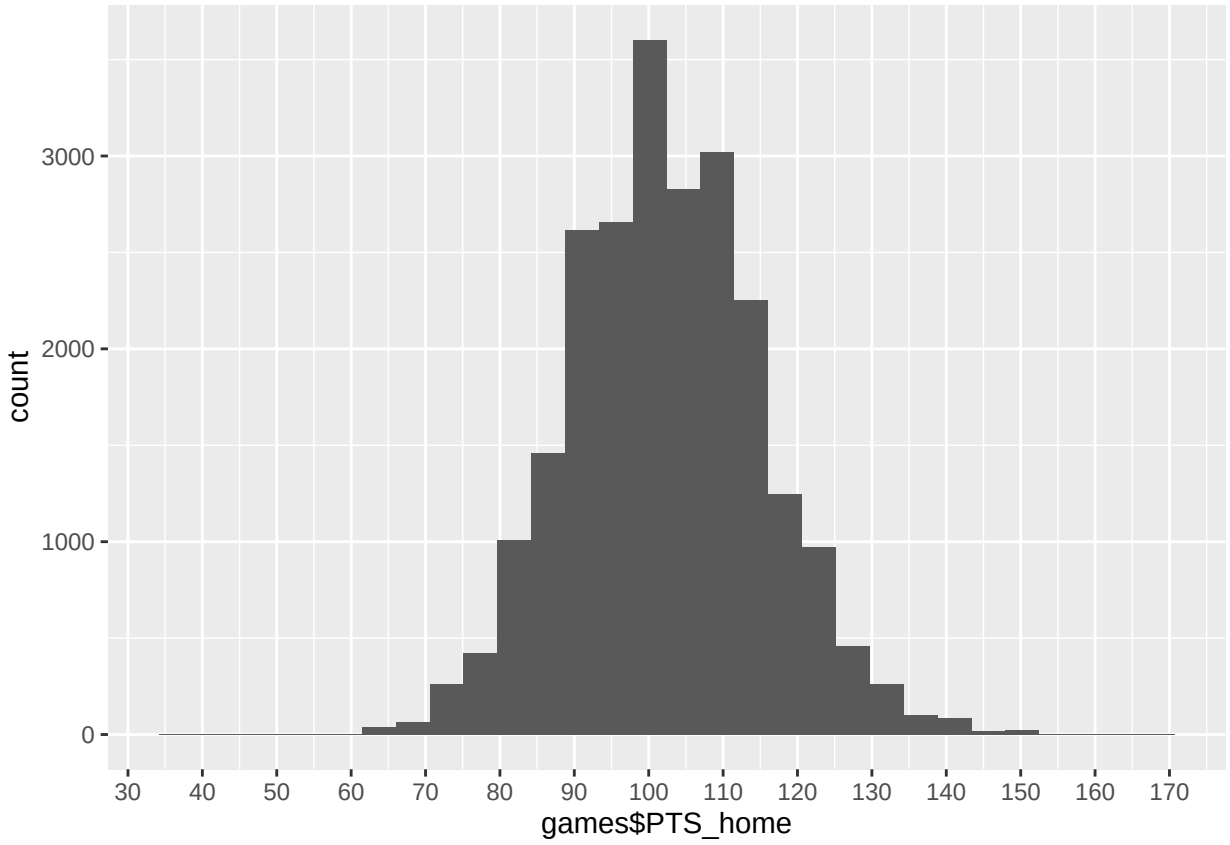
As Home teams win approximately 60% of games, we can see from the above graph that they are more likely to win even when shooting more poorly than the opponents.

```
points <- as.data.frame(games$PTS_home, games$PTS_away)
```

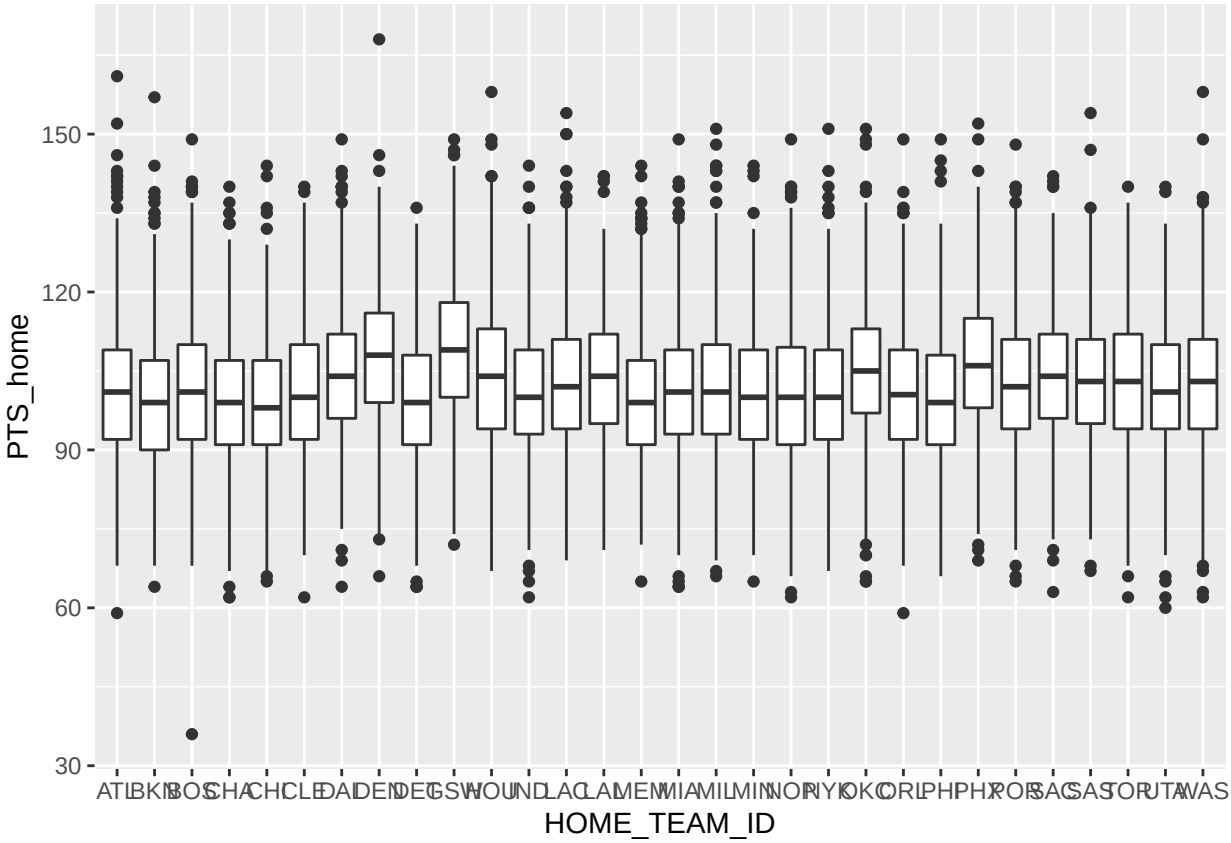
```
## Warning in as.data.frame.integer(games$PTS_home, games$PTS_away): 'row.names' is
## not a character vector of length 23421 -- omitting it. Will be an error!
```

```
ggplot(points, aes(x=games$PTS_home)) + geom_histogram() + scale_x_continuous(breaks = seq(0, 400, by =
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

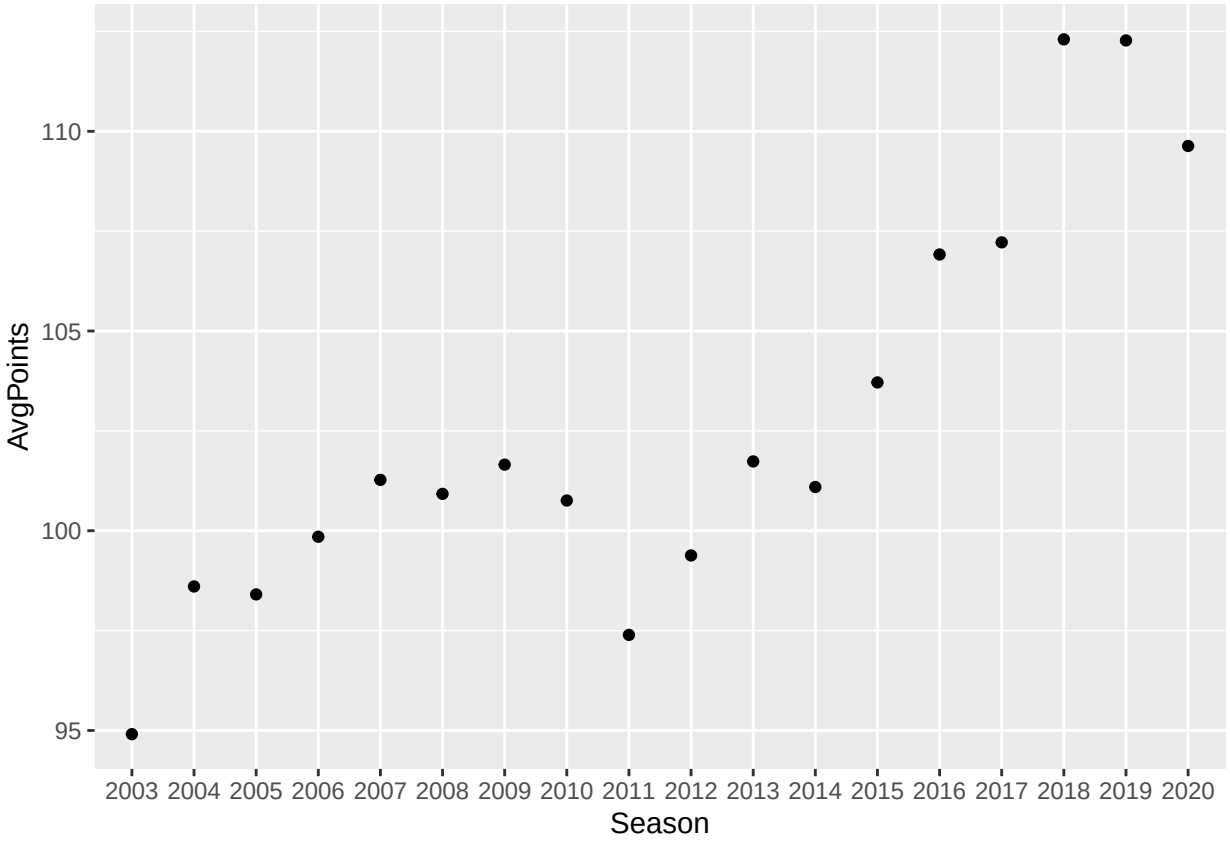



```
ggplot(games, aes(y=PTS_home, x=HOME_TEAM_ID)) +
  geom_boxplot()
```



From this boxplot we can see that points scored by the home team do not vary wildly between teams. While there are clear outliers, the majority of teams average scores which seem to be within approximately 20 points of each other.

```
ggplot(Ptsbyyear, aes(x=Season, y=AvgPoints)) +
  geom_point()
```



Looking at scoring by season, the average has fluctuated, but shows a clear indication that high scoring games are becoming more frequent over time.

2.3 Insights Gained

From the data exploration and visualization phase we have determined that there is a significant difference in home and away team performance, so we should seek to isolate these variables for the time being. Additionally, we may see better performance if we only evaluate using data from the same season being predicted, and eliminate or devalue outliers from the models.

2.4 Modelling Approach

Based on the insights gained, we will take a step by step approach attempting to build a substantive model one layer at a time. Starting with the average, we will use that as a baseline to compare regression algorithms. Next, we will attempt to add to the regression by tuning it.

3.0 Results

In this section we will look at the results of different models and discusses the model performance using RMSE.

3.1 Modelling Results and Performance

3.2 Baseline Model

First we will look at the performance of a simple mean model:

```
avg_points <- mean(games$PTS_home)
avg_points
```

```
## [1] 102.2834
```

```
rmse_results <- tibble()
# baseline - the average points for all home teams in games played
baseline_rmse <- RMSE(validation$PTS_home,avg_points)
## Test results based on simple prediction
baseline_rmse
```

```
## [1] 12.97371
```

```
rmse_results <- tibble(method = "Mean Only", RMSE = baseline_rmse)
rmse_results
```

```
## # A tibble: 1 x 2
##   method     RMSE
##   <chr>     <dbl>
## 1 Mean Only  13.0
```

3.2 Regression Model

Next we will look at a regression model, starting by considering all of the data points we have available.

```
HomePointsModel = lm(PTS_home ~ FG_PCT_home + FT_PCT_home + FG3_PCT_home + AST_home + REB_home + FG_PCT

predictedscores <- predict(HomePointsModel, newdata = validation, type = "response")
HomePointsModel_rmse <- RMSE(validation$PTS_home,predictedscores)

## Test results based on regression algorithm
rmse_results<- add_row(rmse_results, method = "Regression", RMSE = HomePointsModel_rmse)
rmse_results
```

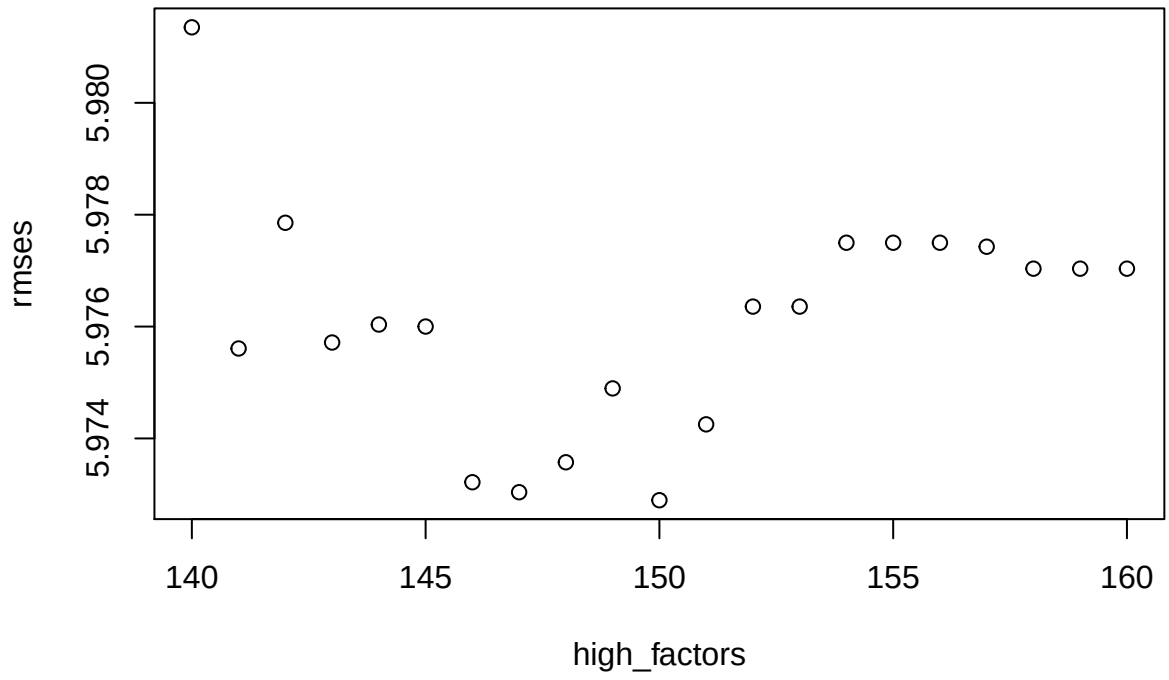
```
## # A tibble: 2 x 2
##   method      RMSE
##   <chr>      <dbl>
## 1 Mean Only  13.0
## 2 Regression  5.95
```

Using the logistic regression model we have already reduced the RMSE by half, a fantastic improvement, but can we fine tune it to make it more advanced?

3.3 Advanced Model

First we will try to improve the model by tuning to ignore outliers from negatively impacting predictions.

```
low_factors <- seq(50, 70, 1) #test for lambda value
high_factors <- seq(140, 160, 1) #test for lambda value

rmses <- sapply(high_factors, function(l){

adv_games<-games[!(games$PTS_home>l),]

HomePointsModel = lm(PTS_home ~ FG_PCT_home + FT_PCT_home + FG3_PCT_home + AST_home + REB_home + FG_PCT

advpredictedscores <- predict(HomePointsModel, newdata = validation, type = "response")
test <- data.frame(Points = as.integer(advpredictedscores))

#HomePointsModel_rmse <- RMSE(validation$PTS_home,test$Points)

  return(RMSE(test$Points, validation$PTS_home))
})
plot(high_factors, rmses)
```




```
## Test results based on regression algorithm
rmse_results<- add_row(rmse_results, method = "Remove Outliers", RMSE = min(rmses))
rmse_results
```

```
## # A tibble: 3 x 2
##   method           RMSE
##   <chr>           <dbl>
## 1 Mean Only       13.0
## 2 Regression       5.95
## 3 Remove Outliers  5.97
```

Using a tuning algorithm to remove severe outliers (either high or low) does not actually improve the RMSE score at all.

4.0 Conclusion

As you can see from the modeling results and performance, there is much that could be improved to increase its accuracy.

The report was limited from the data available and from the skills of its author. More advanced metrics such as offensive or defensive rating would likely add weight to the correlation, as would number of shot attempts for each category rather than only shot percentage. Being able to isolate which players were available to play in each specific game would also be helpful as some players have an outsized impact compared to minutes played.

All of these limitations could be addressed in future work that builds on this modelling approach.