Group 12:

Chen Jia	u863194
Jose Orozco Beccera	u512585
Marloes Evers	u264541
Sathya Jagannatha	u580435

The Performance and Positions of NBA Players

A Statistical Research

Programming with R: Group Project

Introduction:

"Analytics are part and parcel of virtually everything we do now." Adam Silver, commissioner of the NBA stated in an interview (2017). And this is not without a reason. The massive data collection within the NBA, tracking every movement of every player on the court, has been changing the game since 2009, turning "past mediocre teams into contenders." (Abbas, 2019). This has led to the fact that nearly all NBA teams nowadays have data analysts employed to analyse and advice on their tactics.

The goal of this paper is to find out whether the combination of statistical features can be used to predict the position of the players and whether these statistical features can predict if a player will score more or less points than average. Hence, two research questions were proposed:

- To what extent can the (statistical) performance of NBA players be used to determine their position?
- To what extent can the statistical data be used to predict whether a player will score more or less points than average?

To answer the research questions, three predictive models were trained. These are respectively KNN, Logistic Regression and Random Forest and were trained on the NBA player statistics from the 2014-2015 season.

Appendix B1 holds an extensive list of all used variables including abbreviations and definitions. Appendix B2 shows the variables of the original dataset that were not used within the analyses.

Pre-processing and EDA

The dataset that was used for this project was provided by the platform Kaggle, and consisted of the combination of two scraped datasets, one including the NBA stats for Season 2014 – 2015, the other one including personal details of the players, such as height and weight. The dimensions of the original dataset were 490 x 34. The dataset however contained several missing values regarding the variables “weight” and “height”. Because these variables were considered important for the final analyses, players with no information on weight and height, were omitted from the dataset.

After this, irrelevant considered features, such as birthplace and college were removed. For an extensive list of all removed variables, see Appendix B2. The variables “Minutes played” and “Games played” were initially kept within the dataset but were eventually not used in the final analyses.

The data was also checked for outliers. Potential outliers within the variables “Points”, “3 Points made” and “Weight”, grouped by Position were identified and can be found in Appendix A1, A2 and A3. This outlier analysis led to the exclusion of four players, making the final dimension of the dataset after pre-processing into 419 x 21.

The data was further explored using several techniques, including checking the mutual correlation. As could be expected the variable “Points” is strongly correlated with “Field Goals Made”, “Field Goals Attempted” and “Free Throws Made” (respectively 0.991, 0.989 and 0.910), as well as these variables among themselves. The distribution of the variables “Height” and “Weight” per position was also checked. Both variables were considered normally distributed. (See Appendix A4 and A5)

Modelling

For each of the models, the data was split into a train and a test set, with sizes of respectively 70 % and 30 %. The number of observations for these sets can be found in Appendix B3.

As stated before, three models were implemented to answer the research questions. The KNN model and Random Forest were used with regard to the prediction of the positions. The prediction of above or below average point score was done using Logistic Regression and Random Forest.

The KNN model was run on all variables, except for the ones that were excluded earlier (Appendix B2). Both possibilities of 5- and 10-fold cross validation were investigated. The best results were found after using 5-fold cross validation and after scaling and centring the model. In Appendix B4 it can be seen that the best accuracy was achieved with n = 11, leading to an accuracy of 0.702. Appendix B5 shows the confusion matrix of the model with a test accuracy of 0.675. The sensitivity scores per position vary from 0.545 (Small Forward) to 0.833 (Point Guard) and the specificity scores vary from 0.875 (Power Forward) to 0.980 (Centre). Overall, this indicates that the predictive power the model achieved was quite good.

The Random Forest was also used to predict the positions of the players and was trained on the same variables as the KNN model. The default setting of 500 trees was used. Appendix B6 shows that there was an OOB (Out of Bag) error estimate of 24.57%. This means that 75.46% of the OBB samples were correctly classified by the Random Forest. The confusion matrix in B7 shows that the test accuracy achieved was 0.762. With the Random Forest, the sensitivity scores per position vary from 0.680 (Small Forward) to 0.846 (Power Forward) and the specificity scores vary from 0.890 (Power Forward) to 0.990 (Centre). After looking at these results it might be concluded that the Random Forest scores slightly better than the KNN model in predicting the player’s positions. The ranges of both the sensitivity and the specificity scores are smaller and the scores are higher. Also, the test accuracy was higher (KNN: 0.675 vs RF: 0.762). The fact that the Random Forest outperforms the KNN was not considered surprising. Where KNN only focusses on the distance between the features, Random Forest considers causal relationships among these features.

To predict whether a player would score less or more points per average in a season, models for Logistic Regression and Random Forest were trained. As can be seen in Appendix A6, the distribution of “Points” is positively skewed. Despite there is a possibility this could slightly affect the final outcomes, the decision was made to continue using the mean of the variable for this analysis. The values of “Points” was converted into a binary classification, using the mean value of 509.75. Players were classified 0 if below the average and 1 if above.

The Logistic Regression model was trained on all the variables available, except for “Position”, as this variable might strongly be related to “Points”. 5-fold cross validation was used to train the model. Appendix B8 displays the initial outcomes of the Logistic Regression model. A train accuracy of 0.976 was achieved. This seems extremely high. When looking into the confusion matrix (Appendix B9), a test accuracy of 0.968 can be found, even as a sensitivity of 1.000 and specificity of 0.927. As a score of 1 is well-nigh impossible, using this model might unfortunately not be appropriate.

Appendix B10 and B11 show the outcomes and confusion matrix of the Random Forest, based on the same binary classification for “Points” and the same other variables as used in the Logistic Regression Model. The OOB error estimate is 1.7 %, indicating a correct classified percentage of 98.3 %. This is again extremely high, even as the test accuracy (0.96), the sensitivity (0.957) and the specificity (0.964). Unfortunately, also in this confusion matrix of the random forest a well-nigh impossible value of 1 can be found, the Mcnemar's Test P-Value.

Conclusion

In this report three models were trained and tested to answer the following research questions: “To what extent can the (statistical) performance of NBA players be used to determine their position?” and “To what extent can the statistical data be used to predict whether a player will score more or less points than average?”

To answer the first question, both KNN and Random Forest had good predictive power when it came to predicting the positions of the players, based on their statistical performance. The test accuracies of both methods can be considered quite good. (KNN: 0.675, RF: 0.762). Nevertheless, the sensitivity and specificity scores per position were slightly better in the Random Forest classification. As Random Forest, in contrast to KNN, considers the causal relationship between the features, this was not surprising.

The models trained to answer RQ 2 performed extremely well, with test accuracies of LR: 0.968 and RF: 0.960. Unfortunately, within the sensitivity and the Mcnemar’s Test, values of 1 could be found, were a value of 1 is usually highly unlikely. However, since it was stated that distribution of “Points” was strongly positively skewed and this might affect the outcome, this might not be the only problem that occurred in the analysis. Within the scope and time of this research, it was unfortunately not possible to discover and solve these problems. Further research and possibly a larger dataset it might be needed to find a final answer to the question.

References

Sources consulted and referred to

Abbas, N. (2019, August 31). NBA Data Analytics: Changing the Game. Retrieved from https://towardsdatascience.com/nba-data-analytics-changing-the-game-a9ad59d1f116

Goldstein, O. (2017). NBA Players Stats 2014-2015 [Points, Assists, Height, Weight and other personal details and stats]. Retrieved from https://www.kaggle.com/drgilermo/nba-players-stats-20142015

Koehrsen, W. (2018, June 21). Random Forest Simple Explanation - Will Koehrsen. Retrieved from https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

Wharton School of Business. (2017). The NBA's Adam Silver: How Analytics Is Transforming Basketball. Retrieved from https://knowledge.wharton.upenn.edu/article/nbas-adam-silver-analytics-transforming-basketball/

Wood, R. (2020, March 10). Navigating the Random Forest Algorithm in R. Retrieved from https://towardsdatascience.com/navigating-the-random-forest-algorithm-in-r-5ccbc0ef70e

RStudio and packages used:

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22

Baptiste Auguie (2017). gridExtra: Miscellaneous Functions for "Grid" Graphics. R package version 2.3. URL https://CRAN.R-project.org/package=gridExtra

Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R package version 1.0.3. URL https://CRAN.R-project.org/package=tidyr

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 0.8.5. URL https://CRAN.R-project.org/package=dplyr

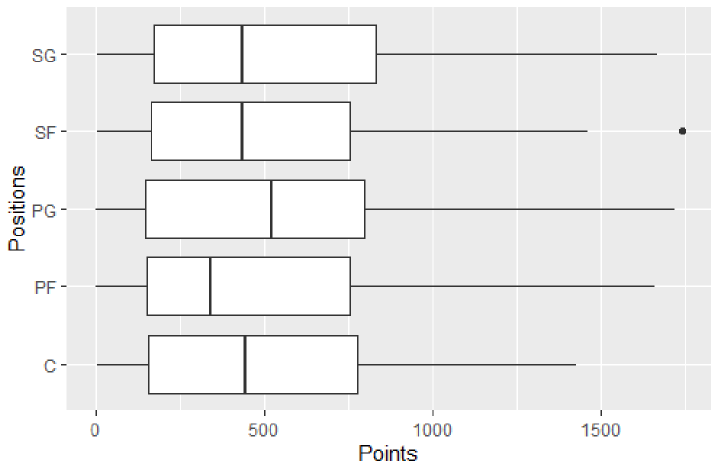
Hadley Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Max Kuhn (2020). caret: Classification and Regression Training. R package version 6.0-86. URL https://CRAN.R-project.org/package=caret

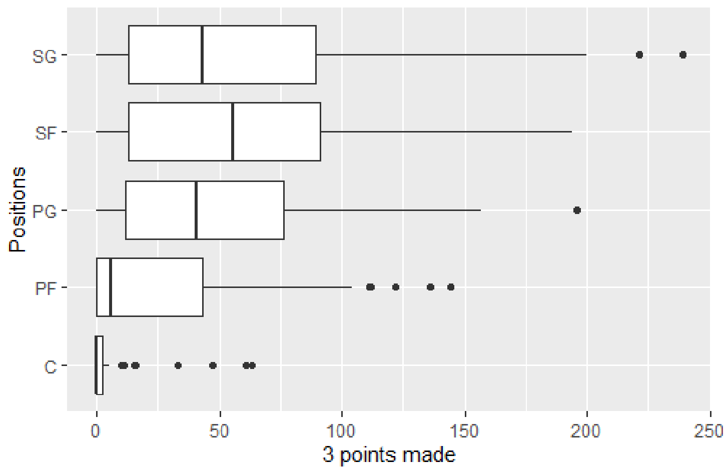
R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

APPENDIX A:

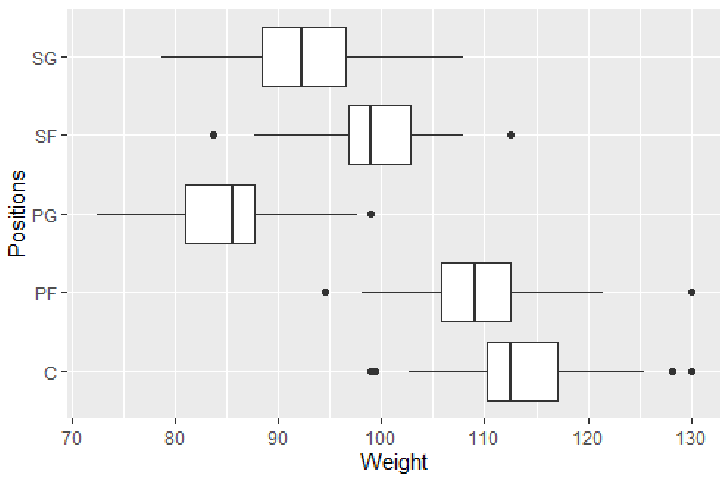
A1: Outliers Points per position



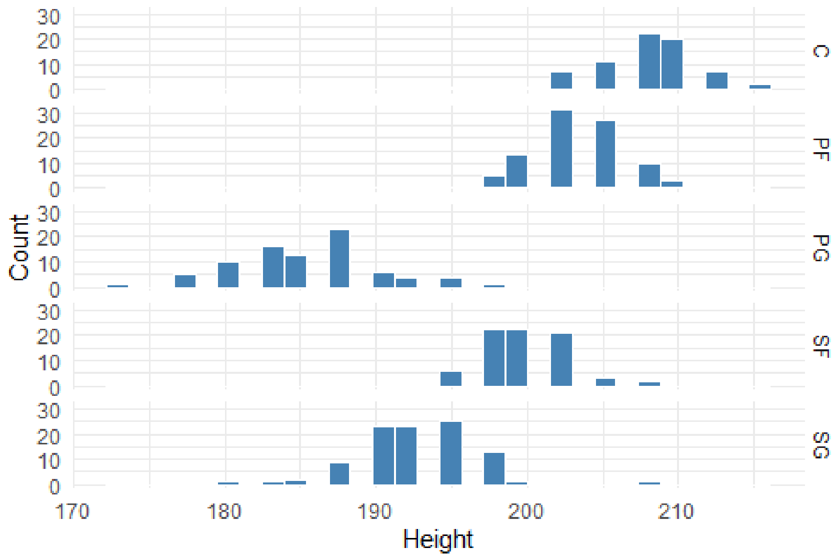
A2: Outliers 3 Points made per position



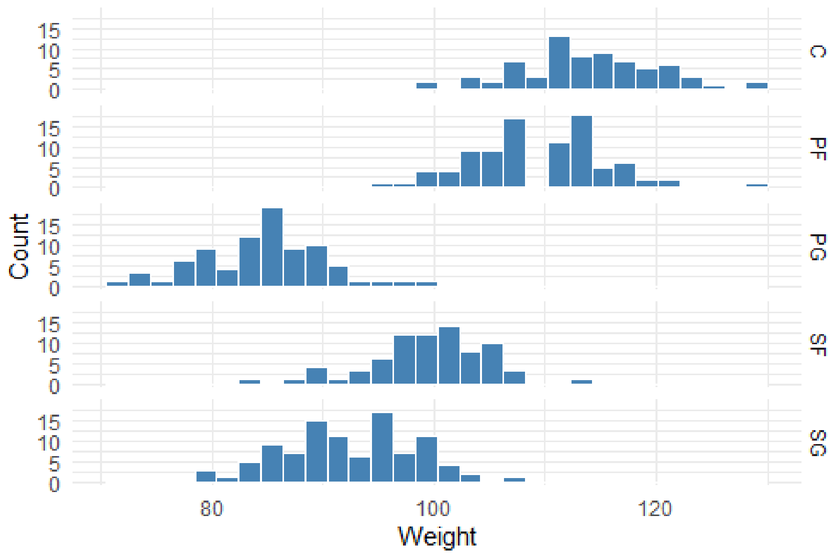
A3: Outliers Weight per position



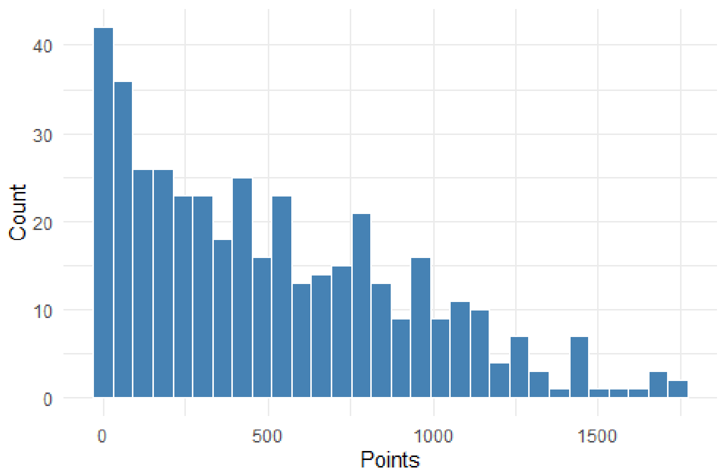
A4: Distribution of Height per position



A5: Distribution of Weight per position



A6: Distribution of Points



APPENDIX B

B1: Used variables, their abbreviation and definition

ABBREVIATION	VARIABLE	DEFINITION
AGE	*Age*	Age
AST	*Assists*	The time of successful assist. Assist means help teammate to take the score.
BLK	*Blocks*	The time of successfully block balls when opponents attempting to shoot.
DREB	*Defensive Rebounds*	The time of successful rebounds in his field.
FGA	*Field Goals Attempted*	The times of field shoot attempted
FGM	*Field Goals Made*	The times of successful field shoot
FTA	*Free Throws Attempted*	The times of free throws attempted
FTM	*Free Throws Made*	The times of successful free throws. Free throws are opportunity offered by judge when opponent team violate the rules.
HEIGHT	*Height*	Height in centimeters
OREB	*Offensive Rebounds*	The time of successful rebounds in opponent's field. In basketball, a rebound, sometimes colloquially referred to as a board is a statistic awarded to a player who retrieves the ball after a missed field goal or free throw.
PF	*Personal Foul*	A personal foul is a breach of the rules that concerns illegal personal contact with an opponent.
POS	*Position*	The positions of player in a team. There are 5 positions: PG – Point Guard SG – Shooting Guard SF – Small Forward PF – Power Forward C – Centre
PTS	*Points*	How many points the player got in 2014 - 2015
STL	*Steals*	The time of successfully steal balls from opponents
TOV	*Turnovers*	A turnover occurs when a team loses possession of the ball to the opposing team before a player takes a shot at their team's basket.
WEIGHT	*Weight*	Weight in kilograms
X3PA	*3 Point Field Goals Attempted*	The times of 3 points field shoot attempted
X3PM	*3 Point Field Goals Made*	The times of successful 3 points field shoot

B2: Unused variables, their abbreviation and definition

ABBREVIATION	VARIABLE	DEFINITION
AST.TOV	*Assist Turnovers*	The rate of assist out of turnover, AST/TOV
BIRTH_PLACE	*Place of Birth*	Place of Birth
BIRTHDATE	*Data of Birth*	Data of Birth
BMI	*Body Mass Index*	Measure of ratio between the length and weight
COLLEGE	*College*	College
EFF	*Efficiency*	Efficiency is expressed there by a stat referred to as 'efficiency' and abbreviated EFF. It is derived by a simple formula: (PTS + REB + AST + STL + BLK − Missed FG − Missed FT - TO) / GP.
EXPERIENCE	*Years of Experience*	Years of Experience
FG	*Field Goal Percentage*	Percentage of successful shoots out of all shoots FGM/FGA
FT	*Free Throw Percentage*	Percentage of successful free throws out of all free throws. FTM/FTA
GAMES.PLAYED	*Games Played*	How many games the player played in 2014 - 2015
MIN	*Minutes Played*	How many mins the player played in 2014 - 2015
REB	*Rebounds*	Total times of rebounds, OREB + DREB
STL.TOV	*Stealth Turnovers*	The rate of stealth out of turnover, STL/TOV
TEAM	*Team*	Team
X3P	*3 Point Field Goals Percentage*	Percentage of successful 3 points shoots out of all 3 points shoots X3PM/X3PA

B3: Number of observations in the training and test set per model

KNN

POSITION	TRAINING SET (70 %)	TEST SET (30 %)	TOTAL
PG – POINT GUARD	59	24	83
SG – SHOOTING GUARD	70	29	99
SF – SMALL FORWARD	54	22	76
PF – POWER FORWARD	63	27	90
C – CENTRE	50	21	71
TOTAL	*296*	*123*	*419*

Logistic Regression

POINTS AVERAGE (509.8)	TRAINING SET	TESTING SET	TOTAL
0 (< AVERAGE)	165	70	135
1 (> AVERAGE)	129	55	194
TOTAL	*294*	*125*	*419*

Random Forest

POSITION	TRAINING SET (70 %)	TEST SET (30 %)	TOTAL
PG – POINT GUARD	56	27	83
SG – SHOOTING GUARD	73	26	99
SF – SMALL FORWARD	51	25	76
PF – POWER FORWARD	64	26	90
C – CENTRE	49	22	71
TOTAL	*293*	*126*	*419*

POINTS AVERAGE (509.8)	TRAINING SET	TESTING SET	TOTAL
0 (< AVERAGE)	165	70	135
1 (> AVERAGE)	129	55	194
TOTAL	*294*	*125*	*419*

B4: KNN outcomes

```
296 samples
 17 predictor
  5 classes: 'C', 'PF', 'PG', 'SF', 'SG'

Pre-processing: centered (17), scaled (17)
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 236, 237, 238, 237, 236
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.6822190  0.6011685
   7  0.6613053  0.5740574
   9  0.6681395  0.5821305
  11  0.7022170  0.6250285
  13  0.6825054  0.6001042
  15  0.6689480  0.5828738
  17  0.6557900  0.5661612
  19  0.6455601  0.5529752
  21  0.6626300  0.5743866
  23  0.6355620  0.5402278
  25  0.6217709  0.5217937
  27  0.6286635  0.5305963
  29  0.6218839  0.5214317
  31  0.6084376  0.5040016
  33  0.5981473  0.4905466
  35  0.6150438  0.5118428
  37  0.5645354  0.4470429
  39  0.5610871  0.4420413
  41  0.5610871  0.4421589
  43  0.5575823  0.4372863

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 11.
```

B5: KNN Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction  C PF PG SF SG
        C  12  2  0  0  0
        PF  7 21  0  4  1
        PG  0  0 20  0  6
        SF  2  4  0 12  4
        SG  0  0  4  6 18

Overall Statistics

               Accuracy : 0.6748
                 95% CI : (0.5845, 0.7565)
    No Information Rate : 0.2358
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5908

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: C Class: PF Class: PG Class: SF Class: SG
Sensitivity           0.57143    0.7778    0.8333   0.54545    0.6207
Specificity           0.98039    0.8750    0.9394   0.90099    0.8936
Pos Pred Value        0.85714    0.6364    0.7692   0.54545    0.6429
Neg Pred Value        0.91743    0.9333    0.9588   0.90099    0.8842
Prevalence            0.17073    0.2195    0.1951   0.17886    0.2358
Detection Rate        0.09756    0.1707    0.1626   0.09756    0.1463
Detection Prevalence  0.11382    0.2683    0.2114   0.17886    0.2276
Balanced Accuracy     0.77591    0.8264    0.8864   0.72322    0.7572
```

B6: Random Forest outcomes

```
 randomForest(formula = Pos ~ PTS + FGM + FGA + X3PM + X3PA +       FTM + FTA + OREB + DREB
 + AST + STL + BLK + TOV + PF + Age +       Height + Weight, data = trn_rf)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 24.57%
Confusion matrix:
    C PF PG SF SG class.error
C  35 14  0  0  0   0.2857143
PF  9 46  0  9  0   0.2812500
PG  0  0 47  0  9   0.1607143
SF  0  9  0 37  5   0.2745098
SG  0  1 12  4 56   0.2328767
```

B7: Random Forest Confusion Matrix

```
Confusion Matrix and Statistics

            Reference
Prediction   C PF PG SF SG
        C   15  1  0  0  0
        PF   7 22  1  3  0
        PG   0  0 21  0  3
        SF   0  3  0 17  2
        SG   0  0  5  5 21

Overall Statistics

               Accuracy : 0.7619
                 95% CI : (0.6779, 0.8332)
    No Information Rate : 0.2143
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7015

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: C Class: PF Class: PG Class: SF Class: SG
Sensitivity            0.6818    0.8462    0.7778    0.6800    0.8077
Specificity            0.9904    0.8900    0.9697    0.9505    0.9000
Pos Pred Value         0.9375    0.6667    0.8750    0.7727    0.6774
Neg Pred Value         0.9364    0.9570    0.9412    0.9231    0.9474
Prevalence             0.1746    0.2063    0.2143    0.1984    0.2063
Detection Rate         0.1190    0.1746    0.1667    0.1349    0.1667
Detection Prevalence   0.1270    0.2619    0.1905    0.1746    0.2460
Balanced Accuracy      0.8361    0.8681    0.8737    0.8152    0.8538
```

B8: Logistic Regression outcomes

```
Generalized Linear Model

294 samples
 16 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 235, 235, 235, 236, 235
Resampling results:

  Accuracy   Kappa
  0.9762127  0.9515617
```

B9: Logistic Regression Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 70  4
         1  0 51

               Accuracy : 0.968
                 95% CI : (0.9201, 0.9912)
    No Information Rate : 0.56
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9346

 Mcnemar's Test P-Value : 0.1336

            Sensitivity : 1.0000
            Specificity : 0.9273
         Pos Pred Value : 0.9459
         Neg Pred Value : 1.0000
             Prevalence : 0.5600
         Detection Rate : 0.5600
   Detection Prevalence : 0.5920
      Balanced Accuracy : 0.9636

       'Positive' Class : 0
```

B10: Random Forest RQ2 outcomes

```
 randomForest(formula = PTS ~ FGM + FGA + X3PM + X3PA + FTM +      FTA + OREB + DREB + AST
 + STL + BLK + TOV + PF + Age + Height +      Weight, data = trn_pts_rf)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 1.7%
Confusion matrix:
    0   1 class.error
0 162   3  0.01818182
1   2 127  0.01550388
```

B11: Random Forest RQ2 Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 67  2
         1  3 53

               Accuracy : 0.96
                 95% CI : (0.9091, 0.9869)
    No Information Rate : 0.56
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.919

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9571
            Specificity : 0.9636
         Pos Pred Value : 0.9710
         Neg Pred Value : 0.9464
             Prevalence : 0.5600
         Detection Rate : 0.5360
   Detection Prevalence : 0.5520
      Balanced Accuracy : 0.9604

       'Positive' Class : 0
```