What Nba Players Statistics Best Predict Scoring Output.[*]

A Look Into The Metrics Which Best Predit Scoring Output Among NBA Players From The 2020-21 Season.

Adam Labas

27 April 2022

Abstract

First sentence. Second sentence. Third sentence. Fourth sentence.

Keywords:NBA, PTS, REB, AST, MIN, FGA, X3PM

Contents

[*] Code and data are available at: https://github.com/adam-labas/Which-NBA-Stats-Best-Predict-Scoring-Output.

1 Introduction

The NBA has for long been a widely admired and celebrated facet of American popular culture. Every season, the average fan has the opportunity to watch the best players on the face of the earth go at each other, night after night, in nail-biting intensity. As an avid fan of the NBA, I too am a consumer of the NBA and its professional basketball content. This became foundational in my life in 2019 when my hometown Toronto Raptors had an amazing run in the playoffs and won the NBA championship. The NBA is well known for being a league full of tall players and as my data suggests, this is true. However, the Toronto Raptors team that won the NBA finals had two relatively short players: Fred VanVleet and Kyle Lowry. Being a relatively short man myself, I was mesmerized by their abilities to perform at the highest level and their abilities to score the basketball and impact the game. As such, in this Paper, I attempt to explore the true parameters which contribute to players scoring output.

As the new 2021-2022 NBA season started on October 19th 2021, it was reminiscing and was thinking about traditionally short players and their ability to score the basketball with ease. It is counter intuitive to me to think that a player that is 185 cm and 89kg like VanVleet can score with ease on a player that is much taller and heavier than him.

If a player like VanVleet, lacking in height can score the ball with ease, what are actually the qualities and traits which contribute statistically to players being able to score more points per game? In statistical terms what predictors best describe the points per game of an NBA player in the 2020-21 season? We aim to answer this question by developing a simple, yet effective and easy to understand model. I had originally decided to study the relationship between the Age, Height and Weight and the number of points scored. When plotting my data, I noticed that there was not a linear relationship between the predictor variables I chose to study and the response variable. As a result, I modified my research question to instead study the relationship between points per game (PTS) to minutes per game and assists per game as this will indirectly answer the question I had originally intended to investigate. I will be able to make this link once I find data on the positions of players and demonstrate that players with smaller heights are almost always players with positions that traditionally have a lot of assists like the Point Guard.

I think that any analysis which gives an insight on which players are prone to producing the most points is always useful and an have impacts in many fields such as the world of sports gambling and especially fantasy sports. Although I am not a gambler myself, I am an aspiring actuary and data analyst and I find pleasure in being able to bring forth simple results from large complex datasets.

2 Data

In this Data Section 2, I will provide a look into the data acquisition and processing methodology as well as a deep dive into the contents of the data. We will also touch on our exploratory data analysis as it pertains to variable selection and lastly we will discuss the reach of the data.

First of, Tables 1 and 2 give us a glimpse of the data.

Table 1: First ten rows of a dataset of shelture usage without All Population

Player	Points	Rebounds	Age	Minutes	Field Goal Attempts	3PM	Weight	Height
Aaron Gordon	12.4	5.7	25	27.7	10.0	1.2	107	203
Aaron Holiday	7.2	1.3	24	17.8	6.6	1.0	84	183
Aaron Nesmith	4.7	2.8	21	14.5	3.9	0.9	98	196
Abdel Nader	6.7	2.6	27	14.8	4.8	0.8	102	196
Adam Mokoka	1.1	0.4	22	4.0	1.4	0.1	86	193
Al Horford	14.2	6.7	35	27.9	12.9	2.0	109	206
Al-Farouq Aminu	4.4	4.8	30	18.9	4.3	0.3	97	198
Alec Burks	12.7	4.6	29	25.6	10.2	2.1	86	213
Aleksej Pokusevski	8.2	4.7	19	24.2	9.1	1.3	98	208
Alen Smailagic	1.9	1.1	20	5.6	1.8	0.3	84	193

2.1 Data Collection & Processing

All the data used in this analysis was retrieved directly from the NBA's website and used in accordance with their terms and conditions (more on this in section 5.1). The data on the NBA's website is displayed on 11 different pages each containing 50 players except for the last page which only contains 36 players. The data is available online as an HTML table. As a result, I used a Google Chrome web browser extension titled **Dowload Table as CSV** to extract the data into 11 CSV files. More information on the Google Chrome extension and the source code that makes it work can be found at the following git repository: https://github.com/arktiv/table-csv-chrome. After appending all the player data into one data frame, we now have a dataset with all the official NBA regular season data for the 2020-2021 season.

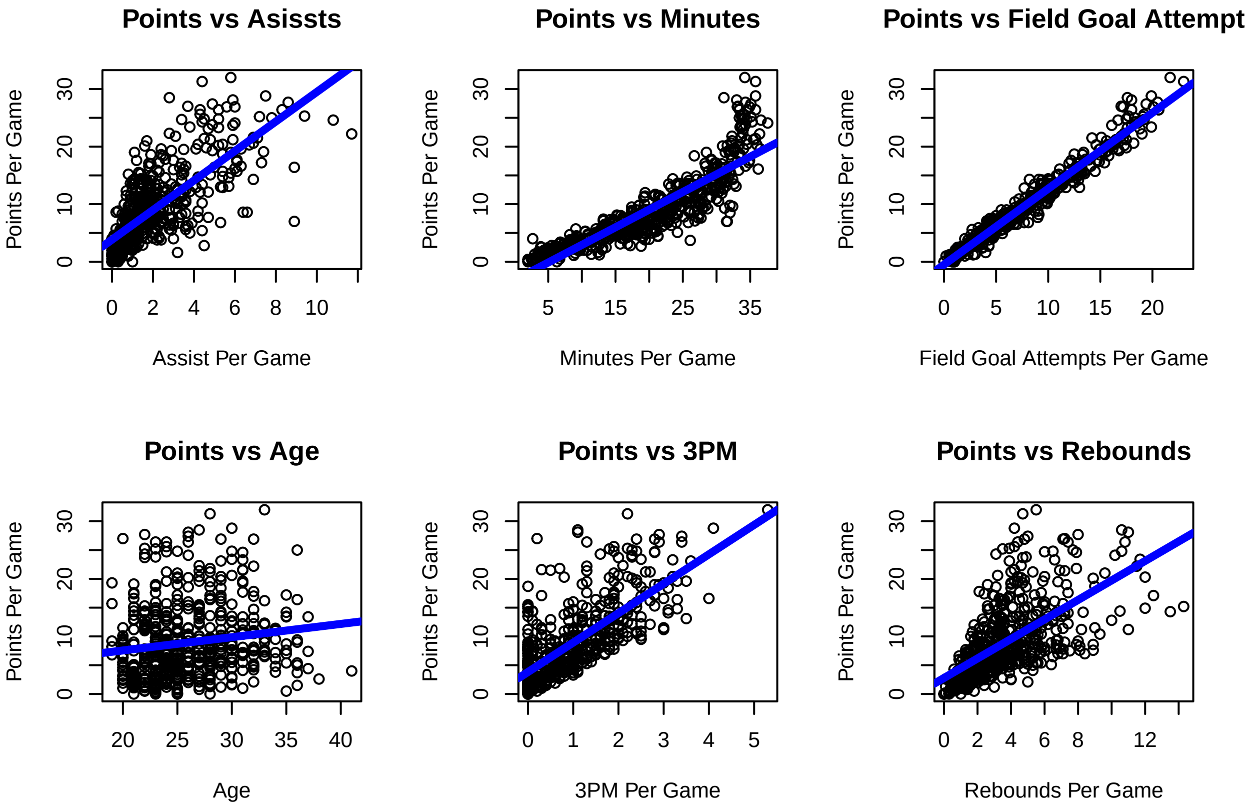
The data collected is listed as *Traditional Stats* on the NBA website. In total, there are 31 variables. The type of data that the NBA considers traditional are the data categories which are simple and straight forward to understand. These variables include but are not limited to player name, team, the points per game, games played, number of wins and losses, minutes played, field goals made, field goals attempted, number of 3 points made and rebounds per game. Some variables like the latter are also further broken down into sub-categories like offensive rebounds and defensive rebounds. In Section 2.2, I will elaborate on the exploratory data analysis, the driving factor which contributed to variable selection.

As far as data processing goes, we began by loading all the necessary libraries like haven (Zhu 2021) into R (R Core Team 2020). Additionally, we used a wide variety of libraries like knitr (Xie 2021), tidyverse (Wickham et al. 2019), tidyr (Wickham 2021), janitor (Firke 2021), patchwork (Pedersen 2020), readr (Wickham, Hester, and Bryan 2021),dplyr (Wickham 2021) as well as tinytex (Xie 2022) at some points in the process for data processing and pdf document generation. We removed the unwanted columns as discussed in Section 2.2 and then rearranged them for purely aesthetic purposes. Lastly, we created two new data: the training and test data. This will help us extensively in Section 3 when we are creating the parsimonious model we desire. To do this, we used the initial_split function to determine the proportion of data that would be in the training and test data. There are a few decisions which needed to be made at this step. The first being the seed and the second being the proportion. I decided to set the seed with a value of 866 which are simply the last three digits of my university student number and hold no other meaning within the analysis. For the proportion, i decided on a 80:20 test to training split. There was no math involved in this decision; i simply wanted to give the training model enough data to produce healthy estimates for the model coefficients.

2.2 EDA: Exploratory Data Analysis

As was stated in the Introduction Section 1, the aim of this analysis is to find a parsimonious model which can predict the scoring output of NBA players in the relevant season. It was very clear to me from the beginning of this analysis that a model with 31 variables is far from parsimonious and had that many variables would have to be eliminated from inclusion in the final model. Ultimately, I decided on 8 predictors, those in the final_raw_data dataset. These variables are age, minutes, field goal attempts, number of 3 points made, rebounds, assists, weight and height. These predictors in my opinion will cover a lot of ground and will explain a reasonable amount of the variance in the data.

Below are a few graphs which demonstrate that there is a clear linear correlation between points and many of the selected variables.



Clearly, we can see that for all the variables plotted above with the exception of age, there is a strong positive linear correlation between the variables and points. For this reason, we have decided that in section **??** when we discuss the model that we come up with, we elected to solely use simple linear models.

2.3 Population, Frame or Sample

It is important to discuss the reach of our data as this will have an effect on the generalizability of our Model. As previously stated, the data we have gathered is of all 540 players which were listed on any teams roster for the 2020-2021 season. As a result, I put forth the claim that the findings of this analysis will have strong generalizability and the model should be reliable to predict the scoring output of any player.

3 Model

In this section, we will go through the process of sifting through the remaining variables to simplify our model slightly and come up with the final model. As we mentioned previously, we have split our data into two datasets: the training set and the test set. The first thing we wish to do is compare the two using summary statistics to see if we can observe any abnormalities or extreme differences between the data. Because we segregated the total data using a set seed, the process was absolutely random and thus we expect to see little to no difference between the two datasets. Below is a table displaying summary statistics for both datasets.

Table 2: Some summary data for the training and test data.

Variable	Mean (Standard Deviation) In Training	Mean (Standard Deviation) In Test
PTS	9.0162 (6.4557)	8.64722 (6.61758)
AGE	26.00926 (4.09174)	25.93519 (4.27834)
MIN	20.01968 (9.12071)	19.03519 (10.01606)
FGA	7.2294 (4.82537)	6.89537 (4.86414)
X3PM	1.01181 (0.91524)	0.96574 (0.86556)
REB	3.68819 (2.39793)	3.42222 (2.36873)
AST	1.99699 (1.88067)	1.95 (1.80878)
KG	98.49537 (11.36323)	97.23148 (10.79253)
CM	199.1088 (8.38542)	198.43519 (9.04844)

As we can see from Table 2, the values for both the mean and the standard deviation of all the predictors are almost identical and there is little to no difference as expected.Now, to create a baseline for our model, lets create a model with all of the predictors. We call this model mod_full.

$$\begin{aligned} \text{PTS} = \alpha + \beta_1(\text{AGE}) + \beta_2(\text{MIN}) + \beta_3(\text{FGA}) + \\ \beta_4(\text{X3PM}) + \beta_5(\text{REB}) + \beta_6(\text{AST}) + \beta_7(\text{KG}) + \\ \beta_8(\text{CM}) + \epsilon \end{aligned} \tag{1}$$

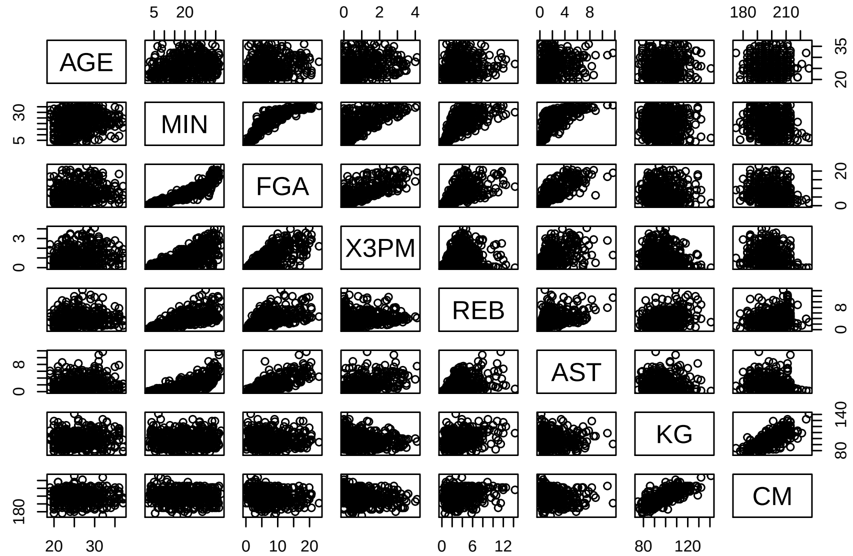
When we run a summary of this mod_full, we can extract the R squared and the adjusted R squared. The adjusted R Squared is the value we will primarily be looking at as we know that when comparing simple linear models with different numbers of predictors, the adjusted R squared is the desired quantity and the simple Rˆ2 will not provide valuable information.

Model Name	R Squared	Adjusted R Squared
mod_full	0.9722968	0.9717729

We see that the adjusted R squared is equal to 97.18%. This is a very high number and after simplifying the model from 31 variables to only 8 variables, we did not lose much information.

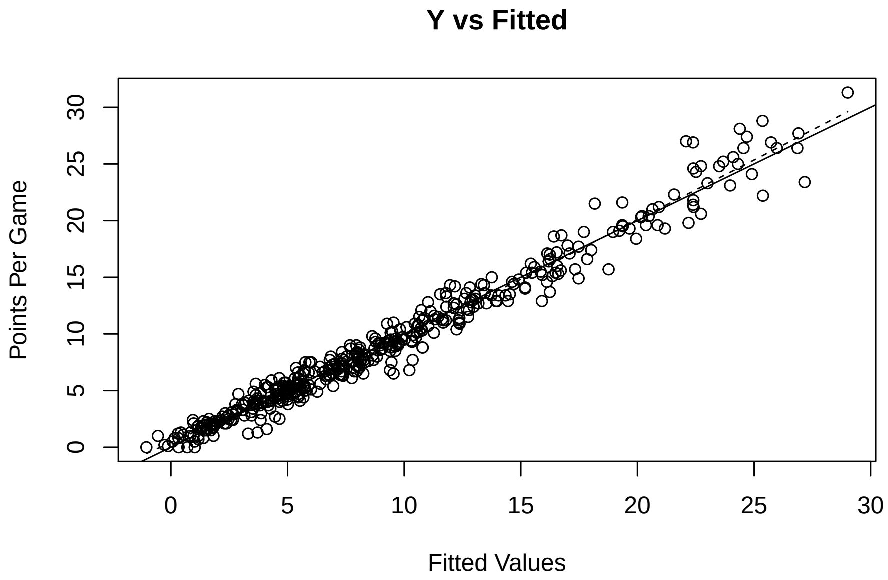
3.1 Checking for Model Violations

The next thing we will do is look for possible model violations. first, lets plot pairwise plots of the predictors and look for any possible evidence of colinearity.



We see very clearly that there is a strong colinearity between minutes played and field goal attempts and between weight and height. We can also see that there is no significant linear relationship between the other pairs of variables.

Lets conduct another test to determine if we will need to transform our data (response and/or predictors).



We see that the points of the Y versus Y hat (fitted values) graph follow very closely to the line of function f(x) = x.

With all the information gathered in the previous steps, we have come to our final model, mod_final:

$$\text{PTS} = \alpha + \beta_1(\text{FGA}) + \beta_2(\text{REB}) + \beta_3(\text{KG}) + \epsilon \tag{2}$$

Model Name	R Squared	Adjusted R Squared
mod_final	0.9718567	0.9716594

4 Results

In this Section 4, we we look at the information gathered in Section 3 and we will objectively describe the events that transpired.

Our first result from Section 3 is the fact that when we plotted the summaries in Table 2, there was little to no difference between the values of the means and standard deviations for the training and test dataset. This is what we expect. If we take a closer look into it, we see that the biggest different between two values from the training and test datasets is a value of the means of weight which differ by 1.26389 KG.

Next, we collected data for mod_full. We notice that the adjusted R squared fro mod_full is very high at 97.18%. If we run a summary for mod_full, we can see that rebounds and field goal attempts are two of the predictors with the three asterisk beside them. This means that they are being listed as having statistically significant p-values. This contributed to the two variables being implemented in the final model.

When we moved on to looking at the potential model violations, we started by examining the pairwise plots of the predictors. We notice that there is a strong linear correlation between minutes played and field goal attempts and between weight and height. Another thing we notice is that there is some fanning effect in some of the plots. This is often associated with a potential violation of constant error variance. This is known as a violation of Homoscedasticity. This can be observed in a lot of the plots such as the one between minutes and 3 points made, minutes and rebounds, minutes and assists and others.

Lastly, we compare the plot of of Y vs the fitted values with the function f(x) = x. We notice a very pure trend and the values follow very closely to the function f(x).

This brings us to the establishment of the final model, mod_final. We conclude that the final model is that in equation (2). The parameters in mod_final are as follows:

$$\alpha = -2.661466$$

$$beta1 = 1.266497$$

$$\beta_2 = 0.177978$$

$$\beta_3 = 0.018937$$

5 Discussion

When we have a look at the results from the later Section 4, we notice that a lot can be done to further the analysis. The following is my personal explanation and speculation for why many of the observations were observed as well as some discussion on the limitations and weaknesses of this analysis and finally some next steps that can be taken to improve the analysis in a future iteration.

We started this analysis because we wanted to determine weather or not height was a reliable predictor for determining the scoring production of NBA players. At surface value, if we simply look at the final model which we produced, we notice that

5.1 Limitations And Weaknesses

5.2 Next Steps

itll be interesting to see how our model changes potentially if we only include players that meet a certain threshold. Like maybe only players who have playes x many games or shoot at least 5 times.

Appendix

A Additional details

References

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://github.com/sfirke/janitor.

Pedersen, Thomas Lin. 2020. *Patchwork: The Composer of Plots*.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2021. *Tidyr: Tidy Messy Data*.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Jim Hester, and Jennifer Bryan. 2021. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Xie, Yihui. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

———. 2022. *Tinytex: Helper Functions to Install and Maintain Tex Live, and Compile Latex Documents*.

Zhu, Hao. 2021. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.