Final Project - NBA MVP Predictions

Doug Marcum

02/26/2020

## Introduction

The National Basketball Association (NBA) is a professional basketball league in North America comprising 30 franchises, of which 29 are in the United States and one in Canada. The league has grown from its humble beginnings to a global enterprise generating in excess of $8 billion in annual revenue. A major component to the rise in popularity of the sport has been directly attributed to the aggressive player centric marketing initiative. Fans often discuss and debate the accolades of specific players with greater passion than they do for the franchises they represent.

The league does a tremendous job of staying ahead of the curve in terms of advanced analytics, but when deciding the most valuable player (MVP), they leave the decision to the opinions of 100 NBA selected media members. Additionally, the league does not provide any criteria in the selection or voting process. Without a defined criterion, the argument for MVP is debated year after year. The purpose of this project is to see if there is an unwritten criterion that is being followed, and once determined, can it be used to predict future MVPs.

### Import Necessary Libraries

library(readr)  
library(dplyr)  
library(corrplot)  
library(car)  
library(caTools)  
library(lmtest)  
library(skimr)  
library(corpcor)

### Data

The data sets used in this project are:

[Kaggle - Full NBA Seasonal Stats](https://www.kaggle.com/lancharro5/seasons-stats-50-19#Seasons_stats_complete.csv) - This data set contains all player statistics from 1950 – 2019. The season from 2010-2019 will be the focus. Additionally, offensive statistics are the concentration of the analysis.

[MVP Totals 2010 - 2019](https://www.basketball-reference.com/awards/mvp.html#mvp_NBA::none) – List of all previous MVP award winners by season from basketball-reference.com.

[2020 Season Stats](https://www.basketball-reference.com/leagues/NBA_2020_per_game.html) - Current season stats from basketball-reference.com.

# load in csv files  
stats\_all <- read.csv('Seasons\_stats\_complete.csv')  
mvp\_votes <- read.csv('MVP\_Voting\_2010to2019.csv', header=TRUE)  
  
# quick inspection of loaded data   
head(stats\_all)

## X Year Player Pos Age Tm G MP PER TS. X3PAr FTr ORB. DRB. TRB.  
## 1 0 1950 Curly Armstrong G-F 31 FTW 63 0 0 0.368 0 0.467 0 0 0  
## 2 1 1950 Cliff Barker SG 29 INO 49 0 0 0.435 0 0.387 0 0 0  
## 3 2 1950 Leo Barnhorst SF 25 CHS 67 0 0 0.394 0 0.259 0 0 0  
## 4 3 1950 Ed Bartels F 24 TOT 15 0 0 0.312 0 0.395 0 0 0  
## 5 4 1950 Ed Bartels F 24 DNN 13 0 0 0.308 0 0.378 0 0 0  
## 6 5 1950 Ed Bartels F 24 NYK 2 0 0 0.376 0 0.750 0 0 0  
## AST. STL. BLK. TOV. USG. OWS DWS WS WS.48 OBPM DBPM BPM VORP FG FGA  
## 1 0 0 0 0 0 -0.1 3.6 3.5 0 0 0 0 0 144 516  
## 2 0 0 0 0 0 1.6 0.6 2.2 0 0 0 0 0 102 274  
## 3 0 0 0 0 0 0.9 2.8 3.6 0 0 0 0 0 174 499  
## 4 0 0 0 0 0 -0.5 -0.1 -0.6 0 0 0 0 0 22 86  
## 5 0 0 0 0 0 -0.5 -0.1 -0.6 0 0 0 0 0 21 82  
## 6 0 0 0 0 0 0.0 0.0 0.0 0 0 0 0 0 1 4  
## FG. X3P X3PA X3P. X2P X2PA X2P. eFG. FT FTA FT. ORB DRB TRB AST STL  
## 1 0.279 0 0 0 144 516 0.279 0.279 170 241 0.705 0 0 0 176 0  
## 2 0.372 0 0 0 102 274 0.372 0.372 75 106 0.708 0 0 0 109 0  
## 3 0.349 0 0 0 174 499 0.349 0.349 90 129 0.698 0 0 0 140 0  
## 4 0.256 0 0 0 22 86 0.256 0.256 19 34 0.559 0 0 0 20 0  
## 5 0.256 0 0 0 21 82 0.256 0.256 17 31 0.548 0 0 0 20 0  
## 6 0.250 0 0 0 1 4 0.250 0.250 2 3 0.667 0 0 0 0 0  
## BLK TOV PF PTS  
## 1 0 0 217 458  
## 2 0 0 99 279  
## 3 0 0 192 438  
## 4 0 0 29 63  
## 5 0 0 27 59  
## 6 0 0 2 4

head(mvp\_votes)

## X X.1 X.2 X.3 X.4 Voting X.5 X.6 X.7 X.8  
## 1 Season Rank Player Age Tm First Pts Won Pts Max Share G  
## 2 2019 1 Giannis Antetokounmpo 24 MIL 78 941 1010 0.932 72  
## 3 2019 2 James Harden 29 HOU 23 776 1010 0.768 78  
## 4 2019 3 Paul George 28 OKC 0 356 1010 0.352 77  
## 5 2019 4 Nikola Jokic 23 DEN 0 212 1010 0.21 80  
## 6 2019 5 Stephen Curry 30 GSW 0 175 1010 0.173 69  
## Per.Game X.9 X.10 X.11 X.12 X.13 Shooting X.14 X.15 Advanced X.16  
## 1 MP PTS TRB AST STL BLK FG% 3P% FT% WS WS/48  
## 2 32.8 27.7 12.5 5.9 1.3 1.5 0.578 0.256 0.729 14.4 0.292  
## 3 36.8 36.1 6.6 7.5 2 0.7 0.442 0.368 0.879 15.2 0.254  
## 4 36.9 28 8.2 4.1 2.2 0.4 0.438 0.386 0.839 11.9 0.201  
## 5 31.3 20.1 10.8 7.3 1.4 0.7 0.511 0.307 0.821 11.8 0.226  
## 6 33.8 27.3 5.3 5.2 1.3 0.4 0.472 0.437 0.916 9.7 0.199

# correct header for mvp\_votes  
names(mvp\_votes) <- as.matrix(mvp\_votes[1, ])  
mvp\_votes <- mvp\_votes[-1, ]  
mvp\_votes[] <- lapply(mvp\_votes, function(x) type.convert(as.character(x)))  
head(mvp\_votes)

## Season Rank Player Age Tm First Pts Won Pts Max Share G MP  
## 2 2019 1 Giannis Antetokounmpo 24 MIL 78 941 1010 0.932 72 32.8  
## 3 2019 2 James Harden 29 HOU 23 776 1010 0.768 78 36.8  
## 4 2019 3 Paul George 28 OKC 0 356 1010 0.352 77 36.9  
## 5 2019 4 Nikola Jokic 23 DEN 0 212 1010 0.210 80 31.3  
## 6 2019 5 Stephen Curry 30 GSW 0 175 1010 0.173 69 33.8  
## 7 2019 6 Damian Lillard 28 POR 0 69 1010 0.068 80 35.5  
## PTS TRB AST STL BLK FG% 3P% FT% WS WS/48  
## 2 27.7 12.5 5.9 1.3 1.5 0.578 0.256 0.729 14.4 0.292  
## 3 36.1 6.6 7.5 2.0 0.7 0.442 0.368 0.879 15.2 0.254  
## 4 28.0 8.2 4.1 2.2 0.4 0.438 0.386 0.839 11.9 0.201  
## 5 20.1 10.8 7.3 1.4 0.7 0.511 0.307 0.821 11.8 0.226  
## 6 27.3 5.3 5.2 1.3 0.4 0.472 0.437 0.916 9.7 0.199  
## 7 25.8 4.6 6.9 1.1 0.4 0.444 0.369 0.912 12.1 0.205

### Data Cleaning

Begin filtering and cleaning the player stats data.

# only include seasons equal to 2010 and later, since the project is focusing on the past ten (10) NBA seasons  
df\_2010 <- stats\_all %>% filter(Year >= 2010)  
  
# minimum of 39 games played is the threshold in the past ten (10) MVP voting years (Derek Rose, 39 Games, 11th in MVP voting)  
df\_2010 <- df\_2010 %>% filter(G >= 39)  
  
# no mvp has played for more than one (1) team in a season, remove TOT team for players playing for more than one team over the course of one season (this also helps eliminate duplicate player entries in a specific season)  
df\_2010 <- df\_2010 %>% filter(Tm != 'TOT')  
  
# determine if any values are missing  
sum(is.na(df\_2010))

## [1] 0

# no missing values found

A new data frame is created with selected variables for evaluation. This is done by narrowing the variables from 50 to 15.

# Season = Season Year, Player = Player's Name, Team = Team, Games = Games Played  
# MinPG = Minutes Per Game, PtsPG = Points Per Game, RebPG = Total Rebounds Per Game  
# AstPG = Assists Per Game, StlPG = Steals Per Game, BlkPG = Blocks Per Game  
# FGPct = Field Goal Percentage, ThreePct = 3 Point Field Goal Percentage  
# PER = Player Efficiency Rating, WS = Player Win Share, VORP = Value Over Replacement Player  
  
clean\_stats\_df <- data.frame('Season' = as.character(df\_2010$Year),   
 'Player' = df\_2010$Player,   
 'Team' = df\_2010$Tm,   
 'Games' = as.numeric(df\_2010$G),   
 'MinPG' = round((df\_2010$MP / df\_2010$G), 1),   
 'PtsPG' = round((df\_2010$PTS / df\_2010$G), 1),   
 'RebPG' = round((df\_2010$TRB / df\_2010$G), 1),   
 'AstPG' = round((df\_2010$AST / df\_2010$G), 1),  
 'StlPG' = round((df\_2010$STL / df\_2010$G), 1),  
 'BlkPG' = round((df\_2010$BLK / df\_2010$G), 1),   
 'FGPct' = round((df\_2010$FG.), 2),   
 'ThreePct' = round((df\_2010$X3P.), 2),  
 'PER' = round((df\_2010$PER), 2),  
 'WS' = df\_2010$WS,   
 'VORP' = df\_2010$VORP)  
  
head(clean\_stats\_df)

## Season Player Team Games MinPG PtsPG RebPG AstPG StlPG BlkPG FGPct  
## 1 2010 Arron Afflalo DEN 82 27.1 8.8 3.1 1.7 0.6 0.4 0.46  
## 2 2010 LaMarcus Aldridge POR 78 37.5 17.9 8.0 2.1 0.9 0.6 0.50  
## 3 2010 Malik Allen DEN 51 8.9 2.1 1.6 0.3 0.2 0.1 0.40  
## 4 2010 Ray Allen BOS 80 35.2 16.3 3.2 2.6 0.8 0.3 0.48  
## 5 2010 Tony Allen BOS 54 16.5 6.1 2.7 1.3 1.1 0.4 0.51  
## 6 2010 Lou Amundson PHO 79 14.8 4.7 4.4 0.4 0.3 0.9 0.55  
## ThreePct PER WS VORP  
## 1 0.43 10.9 4.3 1.3  
## 2 0.31 18.2 8.8 2.6  
## 3 0.17 5.9 0.1 -0.3  
## 4 0.36 15.2 7.9 2.6  
## 5 0.00 14.2 1.9 0.6  
## 6 0.00 14.4 2.9 0.7

Data structures are evaluated, to see if there is anything odd with the new data frame.

# check to make certain structure is correct  
str(clean\_stats\_df)

## 'data.frame': 3285 obs. of 15 variables:  
## $ Season : Factor w/ 10 levels "2010","2011",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Player : Factor w/ 4173 levels "0","A.C. Green",..: 228 2506 2689 3285 3892 2632 679 984 3512 609 ...  
## $ Team : Factor w/ 70 levels "0","AND","ATL",..: 20 50 20 6 6 48 20 25 46 20 ...  
## $ Games : num 82 78 51 80 54 79 76 63 63 69 ...  
## $ MinPG : num 27.1 37.5 8.9 35.2 16.5 14.8 22.3 14.1 14.5 38.2 ...  
## $ PtsPG : num 8.8 17.9 2.1 16.3 6.1 4.7 5.9 5.8 7.7 28.2 ...  
## $ RebPG : num 3.1 8 1.6 3.2 2.7 4.4 6.4 3.3 3.2 6.6 ...  
## $ AstPG : num 1.7 2.1 0.3 2.6 1.3 0.4 0.4 0.7 0.6 3.2 ...  
## $ StlPG : num 0.6 0.9 0.2 0.8 1.1 0.3 0.6 0.2 0.4 1.3 ...  
## $ BlkPG : num 0.4 0.6 0.1 0.3 0.4 0.9 1.9 0.2 0.2 0.4 ...  
## $ FGPct : num 0.46 0.5 0.4 0.48 0.51 0.55 0.57 0.43 0.44 0.46 ...  
## $ ThreePct: num 0.43 0.31 0.17 0.36 0 0 0 0.35 0.37 0.32 ...  
## $ PER : num 10.9 18.2 5.9 15.2 14.2 14.4 15.8 12.1 18.1 22.2 ...  
## $ WS : num 4.3 8.8 0.1 7.9 1.9 2.9 5.9 1.3 3.1 7.9 ...  
## $ VORP : num 1.3 2.6 -0.3 2.6 0.6 0.7 1.8 -0.4 0.8 2.3 ...

# Check on odd level team names. All are correct, once accounting for team name changes.  
sort(unique(clean\_stats\_df$Team))

## [1] ATL BOS BRK CHA CHI CHO CLE DAL DEN DET GSW HOU IND LAC LAL MEM MIA MIL MIN  
## [20] NJN NOH NOP NYK OKC ORL PHI PHO POR SAC SAS TOR UTA WAS  
## 70 Levels: 0 AND ATL BAL BLB BOS BRK BUF CAP CHA CHH CHI CHO CHP CHS ... WSC

# Check on odd level player names. All are correct.  
which(clean\_stats\_df['Player'] == '0', arr.ind=TRUE)

## row col

Here the data is *skimmed* to take a overarching evaluation of it. No data is missing and the distributions are as expected for the subject matter.

skim(clean\_stats\_df)

Data summary

|  |  |
| --- | --- |
| Name | clean\_stats\_df |
| Number of rows | 3285 |
| Number of columns | 15 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 3 |
| numeric | 12 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: factor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| Season | 0 | 1 | FALSE | 10 | 201: 343, 201: 339, 201: 338, 201: 337 |
| Player | 0 | 1 | FALSE | 852 | Ami: 10, And: 10, Ant: 10, C.J: 10 |
| Team | 0 | 1 | FALSE | 33 | SAS: 120, IND: 118, ORL: 116, ATL: 114 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Games | 0 | 1 | 65.11 | 12.97 | 39.00 | 54.00 | 67.00 | 77.00 | 82.00 | ▃▃▃▅▇ |
| MinPG | 0 | 1 | 23.86 | 7.90 | 3.60 | 17.50 | 24.20 | 30.50 | 41.40 | ▂▆▇▇▃ |
| PtsPG | 0 | 1 | 10.13 | 5.63 | 0.50 | 5.90 | 9.00 | 13.40 | 36.10 | ▇▇▃▁▁ |
| RebPG | 0 | 1 | 4.23 | 2.46 | 0.40 | 2.50 | 3.60 | 5.30 | 16.00 | ▇▆▂▁▁ |
| AstPG | 0 | 1 | 2.20 | 1.88 | 0.00 | 0.90 | 1.60 | 2.90 | 11.70 | ▇▂▁▁▁ |
| StlPG | 0 | 1 | 0.75 | 0.41 | 0.00 | 0.40 | 0.70 | 1.00 | 2.50 | ▇▇▃▁▁ |
| BlkPG | 0 | 1 | 0.48 | 0.46 | 0.00 | 0.20 | 0.30 | 0.60 | 3.70 | ▇▂▁▁▁ |
| FGPct | 0 | 1 | 0.46 | 0.06 | 0.26 | 0.42 | 0.45 | 0.49 | 0.74 | ▁▇▅▁▁ |
| ThreePct | 0 | 1 | 0.28 | 0.15 | 0.00 | 0.24 | 0.33 | 0.37 | 1.00 | ▂▇▁▁▁ |
| PER | 0 | 1 | 14.33 | 4.34 | 3.30 | 11.30 | 13.90 | 16.70 | 31.60 | ▂▇▆▁▁ |
| WS | 0 | 1 | 3.51 | 2.91 | -2.10 | 1.40 | 2.80 | 4.90 | 19.30 | ▆▇▂▁▁ |
| VORP | 0 | 1 | 0.91 | 1.47 | -2.20 | -0.10 | 0.50 | 1.50 | 12.40 | ▇▅▁▁▁ |

### Data Normalization

Normalize the data

minmax <- function(x) {(x-min(x))/ (max(x) - min(x))}  
  
clean\_norm <- as.data.frame((lapply(clean\_stats\_df[4:15], minmax)))  
  
clean\_norm <- data.frame('Season' = clean\_stats\_df$Season, 'Player' = clean\_stats\_df$Player, 'Team' = clean\_stats\_df$Team, clean\_norm)  
  
head(clean\_norm)

## Season Player Team Games MinPG PtsPG RebPG  
## 1 2010 Arron Afflalo DEN 1.0000000 0.6216931 0.23314607 0.17307692  
## 2 2010 LaMarcus Aldridge POR 0.9069767 0.8968254 0.48876404 0.48717949  
## 3 2010 Malik Allen DEN 0.2790698 0.1402116 0.04494382 0.07692308  
## 4 2010 Ray Allen BOS 0.9534884 0.8359788 0.44382022 0.17948718  
## 5 2010 Tony Allen BOS 0.3488372 0.3412698 0.15730337 0.14743590  
## 6 2010 Lou Amundson PHO 0.9302326 0.2962963 0.11797753 0.25641026  
## AstPG StlPG BlkPG FGPct ThreePct PER WS VORP  
## 1 0.14529915 0.24 0.10810811 0.4166667 0.43 0.26855124 0.2990654 0.2397260  
## 2 0.17948718 0.36 0.16216216 0.5000000 0.31 0.52650177 0.5093458 0.3287671  
## 3 0.02564103 0.08 0.02702703 0.2916667 0.17 0.09187279 0.1028037 0.1301370  
## 4 0.22222222 0.32 0.08108108 0.4583333 0.36 0.42049470 0.4672897 0.3287671  
## 5 0.11111111 0.44 0.10810811 0.5208333 0.00 0.38515901 0.1869159 0.1917808  
## 6 0.03418803 0.12 0.24324324 0.6041667 0.00 0.39222615 0.2336449 0.1986301

Now the MVP data is cleaned and a new data frame is created.

# MVPRank = Player Final MVP Rank for the Season (1 = Winner)  
# MVPPts = Total Voting Points Received by Player  
clean\_mvp\_df <- data.frame('Season' = as.character(mvp\_votes$Season),  
 'Player' = mvp\_votes$Player,  
 'MVPRank' = mvp\_votes$Rank,  
 'MVPPts' = mvp\_votes$`Pts Won`)  
head(clean\_mvp\_df)

## Season Player MVPRank MVPPts  
## 1 2019 Giannis Antetokounmpo 1 941  
## 2 2019 James Harden 2 776  
## 3 2019 Paul George 3 356  
## 4 2019 Nikola Jokic 4 212  
## 5 2019 Stephen Curry 5 175  
## 6 2019 Damian Lillard 6 69

The two data frames are merged and NA values are converted to zeros (0’s).

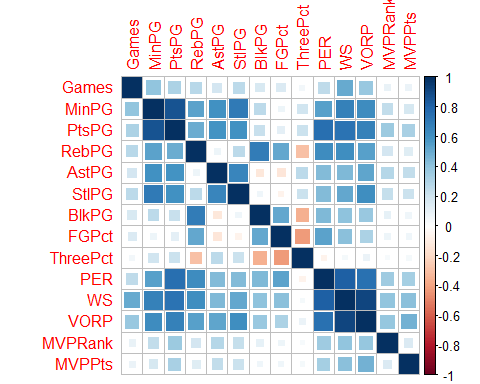
cleaned\_df <- merge(clean\_norm, clean\_mvp\_df, by=c('Season','Player'), all=TRUE, incomparables = NULL)  
  
# converting NAs for MVPRank and MVPPts for players not receiving votes  
cleaned\_df[is.na(cleaned\_df)] <- 0  
  
head(cleaned\_df, 7)

## Season Player Team Games MinPG PtsPG RebPG  
## 1 2010 A.J. Price IND 0.3953488 0.3121693 0.1910112 0.07692308  
## 2 2010 Aaron Brooks HOU 1.0000000 0.8465608 0.5365169 0.14102564  
## 3 2010 Al Harrington NYK 0.7674419 0.7116402 0.4831461 0.33333333  
## 4 2010 Al Horford ATL 0.9767442 0.8333333 0.3848315 0.60897436  
## 5 2010 Al Jefferson MIN 0.8604651 0.7619048 0.4662921 0.56410256  
## 6 2010 Al Thornton LAC 0.2790698 0.6322751 0.2865169 0.21794872  
## 7 2010 Amar'e Stoudemire PHO 1.0000000 0.8201058 0.6348315 0.54487179  
## AstPG StlPG BlkPG FGPct ThreePct PER WS VORP  
## 1 0.16239316 0.24 0.02702703 0.3125000 0.34 0.3780919 0.1542056 0.1506849  
## 2 0.45299145 0.32 0.05405405 0.3541667 0.40 0.4487633 0.3551402 0.2602740  
## 3 0.12820513 0.36 0.10810811 0.3750000 0.34 0.4770318 0.2616822 0.1575342  
## 4 0.19658120 0.28 0.29729730 0.6041667 1.00 0.5689046 0.6074766 0.4109589  
## 5 0.15384615 0.32 0.35135135 0.5000000 0.00 0.5547703 0.3130841 0.2191781  
## 6 0.10256410 0.20 0.10810811 0.4583333 0.36 0.2932862 0.1682243 0.1232877  
## 7 0.08547009 0.24 0.27027027 0.6250000 0.17 0.6819788 0.5981308 0.2602740  
## MVPRank MVPPts  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0  
## 7 10 5

### Correlations

All correlations are plotted and list of correlations to MVPRank has been prepared. A number of interesting correlations present themselves in the data. With the focus on MVPRank, it is easy to see greater correlations with PtsPG, AstPG, PER, WS, and VORP. However, once examining all the correlations, it is discovered that the correlation between PtsPG and PER is approximately, .75, and would cause problems with collinearity with the data. Additionally, MinPG and PtsPG has a high correlation as well, .77, and it too would cause problems with collinearity with the data.

cleaned\_df\_num <- select\_if(cleaned\_df, is.numeric)  
corrplot(cor(cleaned\_df\_num), method = 'square')



MVPcorrelations <- cor(cleaned\_df$MVPRank, cleaned\_df\_num)  
MVPcorrelations

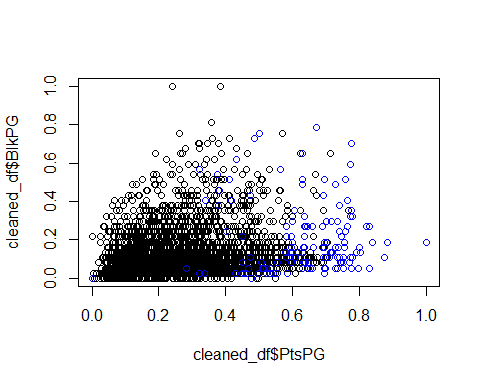
## Games MinPG PtsPG RebPG AstPG StlPG BlkPG  
## [1,] 0.09196662 0.246857 0.3664593 0.1817971 0.2886996 0.2324249 0.1066467  
## FGPct ThreePct PER WS VORP MVPRank MVPPts  
## [1,] 0.07156474 0.03433069 0.3570775 0.3905859 0.3843492 1 0.1510882

x <- cleaned\_df[, 4:15]  
cor\_chart <- cor2pcor(cov(x))  
cor\_chart

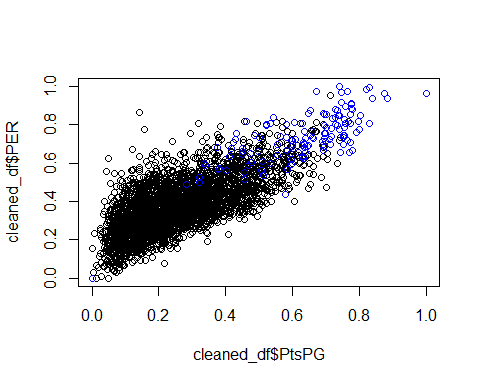
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.000000000 -0.007131723 0.002247464 -0.00767646 0.04094281  
## [2,] -0.007131723 1.000000000 0.773854134 0.44795894 0.43787232  
## [3,] 0.002247464 0.773854134 1.000000000 -0.21723150 -0.23224159  
## [4,] -0.007676460 0.447958937 -0.217231504 1.00000000 -0.33027821  
## [5,] 0.040942805 0.437872318 -0.232241590 -0.33027821 1.00000000  
## [6,] 0.107073587 0.500465316 -0.268881639 -0.14736320 0.04461220  
## [7,] 0.060801307 0.233122854 -0.199097586 0.31600191 -0.26930475  
## [8,] 0.030959499 0.260887174 -0.431800734 -0.04820080 -0.23873838  
## [9,] 0.017913517 0.159043482 0.005629577 -0.27362407 -0.11912928  
## [10,] -0.195653183 -0.653100090 0.746917150 0.30817052 0.41624033  
## [11,] 0.463439962 0.365154721 -0.121505203 -0.05154838 -0.24761376  
## [12,] -0.276159532 -0.337522336 0.110660697 0.12141786 0.30305506  
## [,6] [,7] [,8] [,9] [,10] [,11]  
## [1,] 0.10707359 0.06080131 0.03095950 0.017913517 -0.19565318 0.46343996  
## [2,] 0.50046532 0.23312285 0.26088717 0.159043482 -0.65310009 0.36515472  
## [3,] -0.26888164 -0.19909759 -0.43180073 0.005629577 0.74691715 -0.12150520  
## [4,] -0.14736320 0.31600191 -0.04820080 -0.273624069 0.30817052 -0.05154838  
## [5,] 0.04461220 -0.26930475 -0.23873838 -0.119129278 0.41624033 -0.24761376  
## [6,] 1.00000000 -0.15436575 -0.17096968 -0.072370915 0.27274808 -0.34028049  
## [7,] -0.15436575 1.00000000 0.02440422 -0.141755447 0.21830115 -0.21069466  
## [8,] -0.17096968 0.02440422 1.00000000 -0.240492452 0.53427327 0.06022094  
## [9,] -0.07237092 -0.14175545 -0.24049245 1.000000000 0.06548783 -0.02539564  
## [10,] 0.27274808 0.21830115 0.53427327 0.065487834 1.00000000 0.38525028  
## [11,] -0.34028049 -0.21069466 0.06022094 -0.025395640 0.38525028 1.00000000  
## [12,] 0.45242719 0.25512969 -0.02849732 0.113242171 -0.18920207 0.82271152  
## [,12]  
## [1,] -0.27615953  
## [2,] -0.33752234  
## [3,] 0.11066070  
## [4,] 0.12141786  
## [5,] 0.30305506  
## [6,] 0.45242719  
## [7,] 0.25512969  
## [8,] -0.02849732  
## [9,] 0.11324217  
## [10,] -0.18920207  
## [11,] 0.82271152  
## [12,] 1.00000000

To illustrate the example of correlation difference in categories having an impact on MVPRank, two scatter plots are shown below. The first is PtsPG and BlkPG, and the second is PtsPG and PER. MVPRank is in blue.

plot(cleaned\_df$PtsPG, cleaned\_df$BlkPG)  
mvp\_dot <- cleaned\_df[which(cleaned\_df$MVPRank > 0),]  
points(mvp\_dot$PtsPG, mvp\_dot$BlkPG,col="blue")



plot(cleaned\_df$PtsPG, cleaned\_df$PER)  
mvp\_dot <- cleaned\_df[which(cleaned\_df$MVPRank > 0),]  
points(mvp\_dot$PtsPG, mvp\_dot$PER,col="blue")



### Regression

The linear regression model is built and processed.

mod\_1 <- lm(MVPRank ~ PtsPG + RebPG + AstPG + StlPG + WS + VORP, data=cleaned\_df)  
summary(mod\_1)

##   
## Call:  
## lm(formula = MVPRank ~ PtsPG + RebPG + AstPG + StlPG + WS + VORP,   
## data = cleaned\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.5899 -0.4840 -0.0558 0.2261 15.3971   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.96619 0.06418 -15.054 < 2e-16 \*\*\*  
## PtsPG 1.81582 0.28352 6.405 1.72e-10 \*\*\*  
## RebPG -0.82400 0.22184 -3.714 0.000207 \*\*\*  
## AstPG 0.96099 0.24276 3.959 7.70e-05 \*\*\*  
## StlPG -1.07717 0.23671 -4.551 5.54e-06 \*\*\*  
## WS 1.93828 0.55927 3.466 0.000536 \*\*\*  
## VORP 2.84013 0.72227 3.932 8.59e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.467 on 3279 degrees of freedom  
## Multiple R-squared: 0.1836, Adjusted R-squared: 0.1822   
## F-statistic: 122.9 on 6 and 3279 DF, p-value: < 2.2e-16

With an Adjusted R-squared of .1822 for mod\_2, it accounts for approximately 18.2% of the variation in the MVP voting. Considering there are no requirements in the formatting for MVP voting, this is to be expected. Opinion is the factor that cannot be accounted for, but this model is hoping to understand. Additionally, when looking at the adjusted R2 for mod\_1, we see that difference is small (approximately 0.02%). This shrinkage means that if the model were derived from the population rather than a sample it would account for approximately 0.02% less variance in the outcome.

### Confidence Intervals

None of the confidence intervals cross zero (0), thus it can be assumed the predictors are related to the outcome. The confidence intervals all seem to be small, which indicates a good model.

confint(mod\_1)

## 2.5 % 97.5 %  
## (Intercept) -1.0920310 -0.8403506  
## PtsPG 1.2599357 2.3717101  
## RebPG -1.2589547 -0.3890497  
## AstPG 0.4850136 1.4369663  
## StlPG -1.5412802 -0.6130666  
## WS 0.8417156 3.0348362  
## VORP 1.4239812 4.2562762

### Durbin Watson (DW) Statistic

The value is 2.015, which is so close to 2 that the assumption of independence has certainly been met. The p-value of .66 confirms this conclusion (it is larger than .05 and, therefore, not significant).

durbinWatsonTest(mod\_1)

## lag Autocorrelation D-W Statistic p-value  
## 1 -0.007505983 2.015008 0.738  
## Alternative hypothesis: rho != 0

### Multicollinearity

All of the VIF values are below 10, the statistical tolerances are above 0.1, and the average VIF is below 5. Based on these measures we can conclude that there may be moderate collinearity within our data.

vif(mod\_1)

## PtsPG RebPG AstPG StlPG WS VORP   
## 3.068793 1.871503 2.332270 2.326158 8.835258 8.077351

1/(vif(mod\_1))

## PtsPG RebPG AstPG StlPG WS VORP   
## 0.3258610 0.5343298 0.4287669 0.4298935 0.1131829 0.1238030

mean(vif(mod\_1))

## [1] 4.418556

### Creating and Testing the Prediction

Now the creation of the prediction function. This will create a new data frame, loop through the season, and return the MVP prediction for each season.

MVPprediction <- function(data, model){  
 pred <- predict(model, newdata = data, level = 0.90, type = "response")  
 mvp <- data[0,]  
 for (season in levels(data$Season)){  
 data\_temp <- data[which(data$Season==season),]  
 pred\_temp <- pred[which(data$Season==season)]  
 mvp\_temp <- data\_temp[which(pred\_temp==max(pred\_temp)),]  
 mvp <- data.frame(rbind(as.matrix(mvp), as.matrix(mvp\_temp)))  
 }  
 return(mvp)  
}  
  
MVPprediction(cleaned\_df, mod\_1)

## Season Player Team Games MinPG PtsPG RebPG  
## 190 2010 LeBron James CLE 0.8604651 0.9365079 0.8202247 0.4423077  
## 521 2011 LeBron James MIA 0.9302326 0.9312169 0.7359551 0.4551282  
## 828 2012 LeBron James MIA 0.5348837 0.8968254 0.747191 0.4807692  
## 1150 2013 LeBron James MIA 0.8604651 0.9074074 0.738764 0.4871795  
## 1461 2014 Kevin Durant OKC 0.9767442 0.9232804 0.8848315 0.4487179  
## 1750 2015 James Harden HOU 0.9767442 0.8783069 0.755618 0.3397436  
## 2239 2016 Stephen Curry GSW 0.9302326 0.8095238 0.8314607 0.3205128  
## 2557 2017 Russell Westbrook OKC 0.9767442 0.8201058 0.8735955 0.6602564  
## 2746 2018 James Harden HOU 0.7674419 0.8412698 0.8398876 0.3205128  
## 3085 2019 James Harden HOU 0.9069767 0.8783069 1 0.3974359  
## AstPG StlPG BlkPG FGPct ThreePct PER WS  
## 190 0.7350427 0.64 0.2702703 0.5 0.33 0.9823322 0.9626168  
## 521 0.5982906 0.64 0.1621622 0.5208333 0.33 0.8480565 0.8271028  
## 828 0.5299145 0.76 0.2162162 0.5625 0.36 0.9681979 0.7757009  
## 1150 0.6153846 0.68 0.2432432 0.625 0.41 1 1  
## 1461 0.4700855 0.52 0.1891892 0.5 0.39 0.9363958 0.9953271  
## 1750 0.5982906 0.76 0.1891892 0.375 0.38 0.8268551 0.864486  
## 2239 0.5726496 0.84 0.05405405 0.5 0.45 0.9964664 0.9345794  
## 2557 0.8888889 0.64 0.1081081 0.3333333 0.34 0.9646643 0.7102804  
## 2746 0.7521368 0.72 0.1891892 0.3958333 0.37 0.9363958 0.817757  
## 3085 0.6410256 0.8 0.1891892 0.375 0.37 0.9646643 0.8084112  
## VORP MVPRank MVPPts  
## 190 0.8972603 1 1205  
## 521 0.7123288 3 522  
## 828 0.6712329 1 1074  
## 1150 0.8219178 1 1207  
## 1461 0.7328767 1 1232  
## 1750 0.6849315 2 936  
## 2239 0.8219178 1 1310  
## 2557 1 1 888  
## 2746 0.7191781 1 965  
## 3085 0.8287671 2 776

From the final data set of approximately 280 players per season, the model was able to correctly select the league MVP 70% of the time. Beyond that, the incorrect selections were for players that were selected either second or third in the MVP voting.

Since several assumptions and modifications have been made along the way, it is necessary to break the data into test and train sets. This cross validation will help ensure that the model is not over-fitting. From this test, the accuracy of the model remains high at approximately 72%.

# set random seed  
set.seed(56)  
  
# split the data, set up train and test subsets (90/10 split)  
split <- sample.split(cleaned\_df, SplitRatio = 0.9)  
train <- subset(cleaned\_df, split == 'TRUE')  
test <- subset(cleaned\_df, split == 'FALSE')  
  
# test the data through the model  
res <- predict(mod\_1, train, type='response')  
  
# create confusion matrix to validate and determine accuracy  
cmatrix <- table(Actuals\_Value=train$MVPRank, Predicted\_Value = res > 0.5)  
((cmatrix[[1,1]] + cmatrix[[2,2]]) / sum(cmatrix)\*100)

## [1] 72.48276

### Introducing the Current Season

With an accurate model built, the data from the current 2019-2020 season can be loaded, cleaned, and tested to make a future prediction.

# load in 2020 season statistics to date (accurate to Feb 26 2020)  
stats\_2020 <- read.csv('Season\_2020\_Stats.csv')  
  
# no mvp has played for more than one (1) team in a season, remove TOT team for players playing for more than one team over the course of one season (this also helps eliminate duplicate player entries in a specific season)  
df\_2020 <- stats\_2020 %>% filter(Tm != 'TOT')  
  
# minimum of 39 games played is the threshold in the past ten (10) MVP voting years   
df\_2020 <- df\_2020 %>% filter(G >= 39)  
  
# clean and structure in the same manner/format as previous dataframe (seasons)  
clean\_2020 <- data.frame('Season' = as.character(df\_2020$Year),   
 'Player' = df\_2020$Player,   
 'Team' = df\_2020$Tm,   
 'Games' = as.numeric(df\_2020$G),   
 'MinPG' = round((df\_2020$MP / df\_2020$G), 1),   
 'PtsPG' = round((df\_2020$PTS / df\_2020$G), 1),   
 'RebPG' = round((df\_2020$TRB / df\_2020$G), 1),   
 'AstPG' = round((df\_2020$AST / df\_2020$G), 1),  
 'StlPG' = round((df\_2020$STL / df\_2020$G), 1),  
 'BlkPG' = round((df\_2020$BLK / df\_2020$G), 1),   
 'FGPct' = round((df\_2020$FG.), 2),   
 'ThreePct' = round((df\_2020$X3P.), 2),   
 'PER' = round((df\_2020$PER), 2),  
 'WS' = df\_2020$WS,   
 'VORP' = df\_2020$VORP)  
  
# a small number of players have not attempted a 3 point shot this season (10). Replaced NA with 0 values.   
clean\_2020[is.na(clean\_2020)] = 0  
  
# normalize data  
clean\_2020\_norm <- as.data.frame((lapply(clean\_2020[4:15], minmax)))  
  
# construct new dataframe  
clean\_2020\_norm <- data.frame('Season' = clean\_2020$Season, 'Player' = clean\_2020$Player, 'Team' = clean\_2020$Team, clean\_2020\_norm)

### 2020 NBA MVP Prediction

# run 2020 season MVP prediction  
MVPprediction(clean\_2020\_norm, mod\_1)

## Season Player Team Games MinPG PtsPG RebPG AstPG  
## 103 2020 James Harden HOU 0.7142857 0.9965753 1 0.3605442 0.6893204  
## StlPG BlkPG FGPct ThreePct PER WS VORP  
## 103 0.8 0.3225806 0.3023256 0.6 0.9054545 1 1

### Conclusion

From the beginning of the project, the only method available to attack the problem was from a direct statistical approach. With much of the sports world filled with hyperbole, when no criteria are given in determining a most valuable player, opinions can be tainted by hometown pride, disdain toward a player, or even voter fatigue. During Michael Jordan’s era, he was arguably the most valuable player each season. However, fans and media grew tired of being dominated by the same name and face. Thus, Charles Barkley and Karl Malone were awarded with an MVP trophy each. Much of the same has been talked about during LeBron James time.

Knowing that opinions and attitudes play major factors, it was critical to understand what variables weigh heaviest in voters’ minds. Through minor trial and error, the model was able to predict with a high accuracy the annual MVP. Even for the seasons in which the model’s prediction was incorrect, those are possible seasons where opinion outweighed the statistics.

2011 - The model predicted LeBron James (third in the actual vote). However, this was the season of his public announcement to leave the Cleveland Cavaliers and ‘take his talents’ to South Beach. That season, LeBron without doing anything other than switching teams, went from hero to villain in many minds. Derek Rose was the winner that season, and the ‘hometown hero’ played large for him nationally and in Chicago. His style of play and being undersized, gave him the improbable vote over LeBron.

2015 - The model predicted James Harden (second in the actual vote). While Harden was a fine selection, the nation was enamored with the way Steph Curry was dominating. Voter sentiment could be argued to have been the deciding factor over pure statistics.

2019 - The model predicted James Harden (second in actual vote). Again, Harden had a tremendous season, but a major topic over the course of the season was centered around the style of play in Houston. Many writers argued that Harden and the Rockets were altered the pace and style of the game and not for the better. The winner, Giannis Antetokounmpo, had an outstanding season by leading the Milwaukee Bucks to the number one seed in the Eastern Conference. While I cannot pin this outcome on opinions more than statistics, it was a factor to some degree.

While the project could be seen as a success, there are still limitations and areas where it could be built upon.

1. Social Media - With social media playing a substantial role in our daily lives, it could be presumed to have an influence on how voters might view a player. Whether that be the player’s direct social media account or the numerous postings by fans and media, the number of impressions per player in comparison to voting patterns could be telling.
2. Advanced Metrics/Analytics - While the project touched on two advanced statistics (PER and VORP), the league has fully embraced several additional measures. For purposes of this project, the advanced metrics were kept to a minimum in part due to the unfavorable opinion many veteran analysts and media members have toward their validity. I believe they are not currently being influenced by those numbers, but they could provide additional insight in the coming seasons.
3. Voting Records - Voting was considered as a collective entity itself, instead of individual voters. If access to each voters’ previous ballots were made available, then great trends could potentially be extrapolated.

Overall, the project does assist in helping to define a structure as to how voters look at a player’s body of work when voting for MVP. With the model predicting James Harden as the possible 2020 MVP, it could be argued that he is having a better statistical season than he did during his MVP season of 2018. Just like the voters though, my own opinions run deep, and I see Giannis Antetokounmpo repeating as MVP. Now we wait and see.