MACHINE LEARNING PROJECT

WINSHARES

The Science and Art of Winning Basketball

Raymond Chun

student id: chun1

Contents

Topics to be Covered

Why Win Shares?

“The search for understanding, wherever it roams, is a search for better simplifications. Simplifications which explain more and distort less. All human understanding is based on simplifications of more complex realities.”

— Bill James

Why Win Shares?

Win Shares is an advanced basketball statistic that allows us to give credit to individual NBA players towards the wins of a team. With a very complex and nuanced formula, Win Shares assigns a fully attributed number to the individual on how much they helped win games. This makes Win Shares a desirable stat for prediction.

In section 1 and 2 of the Analysis Report, I demonstrate how I filtered and adjusted for the data set from the NBA. This particular data set covers 1950-2019, but several filters were applied for this project.

On the following page, I reference how the Win Shares formula is calculated and how it is scaled for the NBA.

The Win Shares Formula

Offensive Win Shares:
Points produced for each player
Offensive possessions for each player
Calculate Marginal offense: This is equal to (points produced) - 0.92 * (league points per possession) * (offensive possessions)
*Note that this formula may produce a negative result for some players
Calculate marginal points per win. Marginal points per win reduces to 0.32 * (league points per game) * ((team pace) / (league pace))
Credit Offensive Win Shares to the players.
Offensive Win Shares are credited using the following formula: (marginal offense) / (marginal points per win)

Defensive Win Shares:
Defensive Rating for each player
Calculate marginal defense for each player. Marginal defense is equal to (player minutes played / team minutes played) * (team defensive possessions) * (1.08 * (league points per possession) - ((Defensive Rating) / 100))
Note that this formula may produce a negative result for some players.
Calculate Marginal points: this is per win reduces to 0.32 * (league points per game) * ((team pace) / (league pace))
Credit Defensive Win Shares to the players.
Defensive Win Shares are credited using the following formula: (marginal defense) / (marginal points per win).

Win Shares = Offensive Win Shares + Defensive Win Shares

https://www.basketball-reference.com/about/ws.html

The Data

The Data Set: NBA stats until 2018-2019: Players stats from 1950 to 2019
link: https://www.kaggle.com/lancharro5/seasons-stats-50-19

This file is in a robust data set compiled once again through Basketball Reference and its terrific resources online. It recorded stats and information dating back to 1950 to the most recent complete NBA season in 2018-2019. It is arranged by year, then player. For each row, the yearly season for each player is considered it's own entry.

While it is highly developed, it also acknowledges that there are missing statistics from before the NBA-ABA merger in the 1970's and the early 50's and 60's where stats were not fully recorded. As such, I filtered the data set to acknowledge this.

Also there are some strange data inconsistencies with certain names from international, foreign born players, and hall of famers. The text characters sometimes appear broken or misplaced. I accounted for this using functions to track the players.

III Analysis Report

Current, Next Seasons, All Stars

Outline:

Section 1: Predicting the Current season's Win Share statistic
A. Exploratory Analysis: Correlations, Plots, Histograms, etc
B. Models: Lasso, Linear Regression, Random Forest
C. **Problems, Proposed Adjustments**

Section 2: Predicting the NEXT season's Win Share statistic
A. Data Filter: Mutating the data to lead the Win Shares stat to the next yearly season
B. Models: Lasso, Linear Regression, Random Forest ModelRidge, Elastic Net, ANN

Section 3: Predicting All Stars from their Rookie Season
A. Data Filter: Classifying All Stars
B. Models: Lasso, Ridge, Enet, Logistic Regression

Section 1

Predicting Win Shares for each Current Season

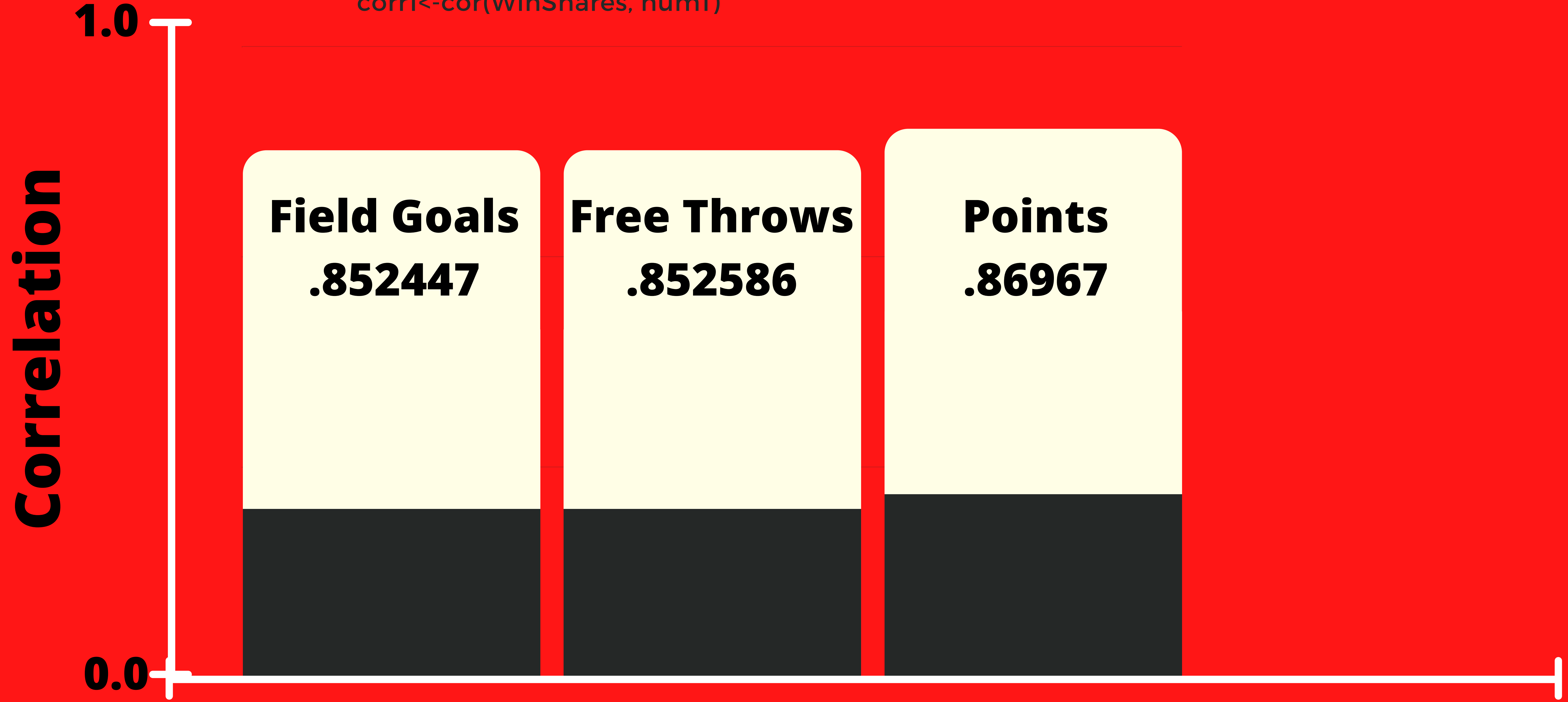
The Process / Algorithm

- Explore the data, correlate with Win Shares
- Filter the data
- Run models: Linear Regression, Random Forest, Lasso
- Compare Results

Exploratory Data

Top 3 correlated variables with Win Shares

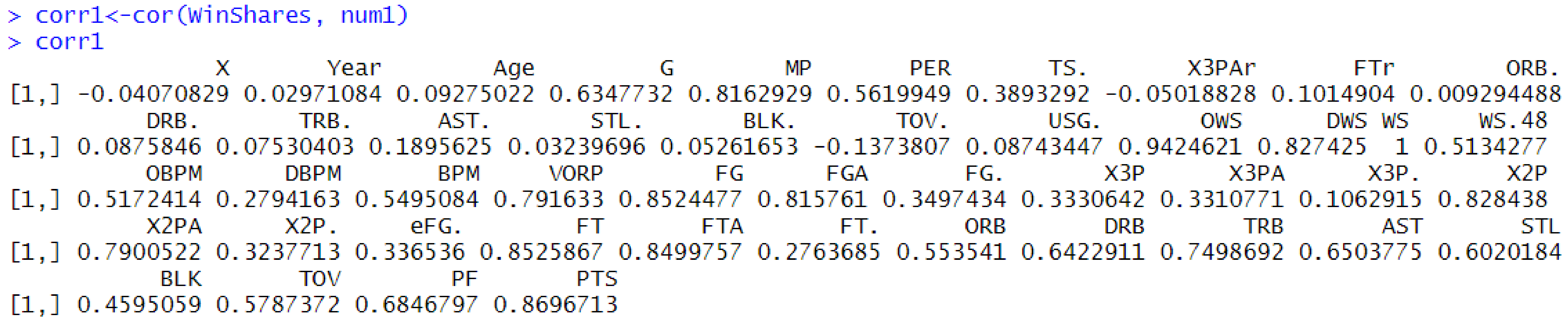
```
#Selecting correlated numerical variables
num1<-dplyr::select_if(NBA, is.numeric)
num1
WinShares=NBA$WS
corr1<-cor(WinShares, num1)
```



Exploratory Data

Full list of correlated variables with Win Shares

```
#Selecting correlated numerical variables
num1<-dplyr::select_if(NBA, is.numeric)
num1
WinShares=NBA$WS
corr1<-cor(WinShares, num1)
```



```
> corr1<-cor(WinShares, num1)
> corr1
               X       Year        Age         G        MP       PER       TS.       X3PAr       FTr        ORB.
[1,] -0.04070829 0.02971084 0.09275022 0.6347732 0.8162929 0.5619949 0.3893292 -0.05018828 0.1014904 0.009294488
          DRB.       TRB.      AST.       STL.       BLK.       TOV.       USG.       OWS      DWS WS     WS.48
[1,] 0.0875846 0.07530403 0.1895625 0.03239696 0.05261653 -0.1373807 0.08743447 0.9424621 0.827425  1 0.5134277
          OBPM      DBPM       BPM     VORP        FG      FGA       FG.       X3P      X3PA      X3P.      X2P
[1,] 0.5172414 0.2794163 0.5495084 0.791633 0.8524477 0.815761 0.3497434 0.3330642 0.3310771 0.1062915 0.828438
          X2PA      X2P.     eFG.        FT       FTA       FT.      ORB       DRB       TRB       AST       STL
[1,] 0.7900522 0.3237713 0.336536 0.8525867 0.8499757 0.2763685 0.553541 0.6422911 0.7498692 0.6503775 0.6020184
           BLK       TOV        PF       PTS
[1,] 0.4595059 0.5787372 0.6846797 0.8696713
```

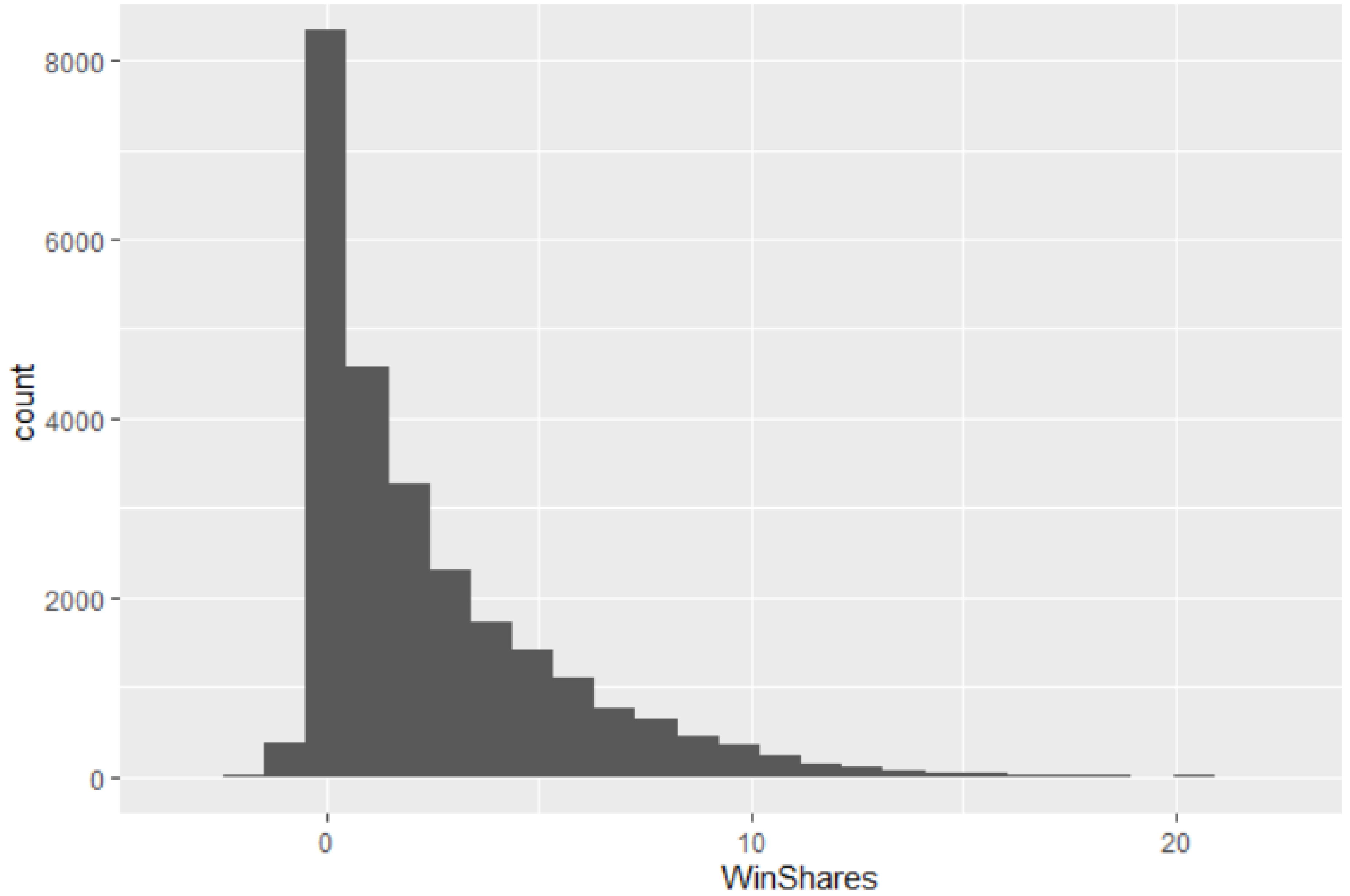
Exploratory Data

Histogram of the Win Shares Variable

```
#Histogram: This shows that Win Shares are very right skewed. Many players do not contribute
at all due to lack of playing time, injuries, etc.
NBA1<-select(NBA, -contains("WinShares"))
ggplot(NBA1, aes(x=WinShares)) + geom_histogram()

summary(NBA$MP)
#Summary of the WinShares variable
summary(WinShares)
```

#Because the formula allows for negative numbers for Win Shares, it makes it difficult to take the log of that variable



Filtering the Data

The Super Set of Variables

```
library(dplyr)
NBA=NBA %>%
  arrange(Year, Player) %>%
  group_by(Player) %>%
  arrange(Year) %>%
   select(Tm, Pos, Player, Year, Age, MP, G, PER, TS., X3PAr, FTr, ORB., TRB., DRB., STL., BLK.,
TOV., USG., OBPM, BPM, DBPM, VORP, FG, FGA, FG., X3P, X3PA, X3P., X2P, X2PA, X2P., eFG.,
FT, FTA, FT., ORB, DRB, TRB, AST, STL, BLK, WS) %>%
filter(Year >= 1977)%>%
filter(MP >= 294)
```

#Not including Defensive or Offensive Win Shares because they add up Win Shares directly.

#Chose the year 1977 as the starting point because they started recording full stats that season

#294 minutes per game is the 1st quantile of minutes played per season. Many players do not play a single minute in the NBA but are on a team.

Models

Linear Regression, Random Forest Model, Lasso

LINEAR REGRESSION

```
set.seed(1)
library(caret)
train.control = trainControl(method = "cv",
number = 10)
linearRegression <- train(WinShares ~.,
data = NBA_train, method = "lm",
trControl = train.control)
print(linearRegression)
summary(linearRegression)

#Checking the RMSE of the WinShares of
the current year
predictionsLR= predict(linearRegression,
NBA_test)
RMSE(predictionsLR,
NBA_test$WinShares)
R2(predictionsLR, NBA_test$WinShares)
```

RMSE: 0.5852
R2: 0.9618

RANDOM FOREST

```
library(caret)
library(randomForest)
set.seed(1)
ctrl <- trainControl(method = "cv", number
= 10)
grid_rf <- expand.grid(mtry = c(2, 4, 8, 16))

rf<- train(WinShares ~ ., data = NBA_train,
importance=T,
method = "rf", trControl = ctrl, tuneGrid =
grid_rf)
varImp(rf)

predictionsRF <- predict(rf, NBA_test)
RMSE(predictionsRF,
NBA_test$WinShares)
R2(predictionsRF, NBA_test$WinShares)
```

RMSE: 0.6065193
R2: 0.9596143

LASSO

```
library(glmnet)
set.seed(1)
WinShares = NBA$WinShares
lasso <- train(
WinShares ~. , data=NBA_train, method =
"glmnet",
trControl = trainControl("cv", number = 10),
tuneGrid = expand.grid(alpha = 1, lambda =
10^seq(-3, 1, length =
100)))
lasso

coef(lasso$finalModel,
lasso$bestTune$lambda)
predictionsL <- predict(lasso, NBA_test)
RMSE(predictionsL, NBA_test$WinShares)
R2(predictionsL, NBA_test$WinShares)
```

RMSE: 0.6153303
R2: 0.9581337

The Problems

DIRECT, INDIRECT CONNECTION

The accuracy numbers are **very high** because Win Shares has a connection with the offensive and defensive numbers, DESPITE having its own formula. While it is not directly in the formula, the relationship is clear.

SKEWNESS

Most players in the NBA do not contribute enough. While it is viewed as the best league in the world, we have to understand that the best players get the most of the minutes, stats, and Win Shares.

USEFULNESS

In past assignments, we predicted income, applications, etc. Those are often influenced by human choices, behavior. Win Shares is a formula for individual players. Using the formula might be better than creating a model in certain situations. Also most of the data is from a different era in time.

Proposed Adjustments

PREDICT NEXT YEAR'S WS

Let's predict next year's Win Shares instead. While the formula does not change, the concept does. We are using any given season and its statistics to see if we can predict it's following NEXT season's Win Shares.

CHANGE THE THRESHOLD

Increase the threshold of minutes played and games played. If a player played all 48 minutes of all 82 games per season, that would be 3936 minutes. The median amount of games played was 57. Let's also use Lasso to improve variable selection.

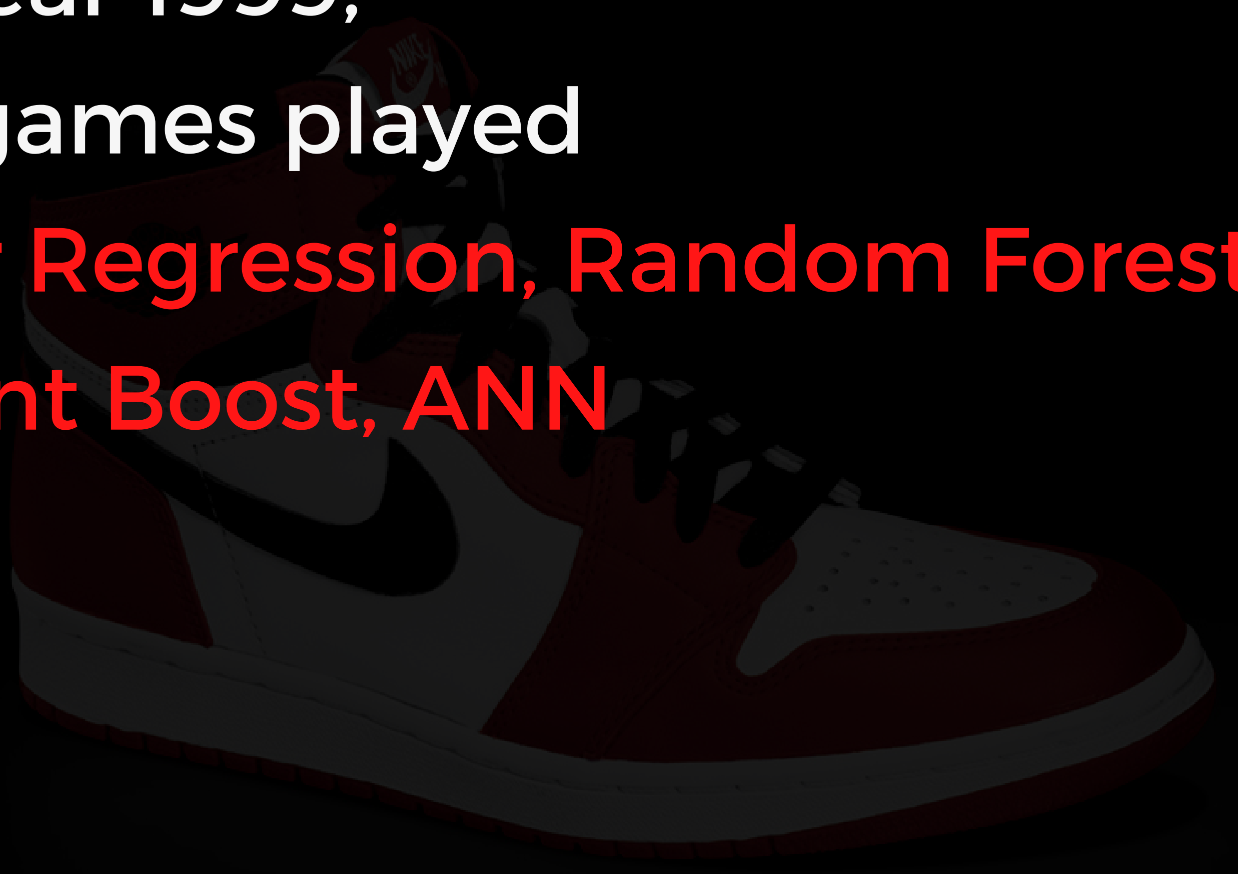
TIME PERIOD

Let's use the last 20 years as a time frame. Big data sets are useful, but the NBA has changed considerably since 1977. In 2019, two active players Vince Carter and Dirk Nowitzki were drafted in 1999. This shows relevance in time and style of play.

Section 2

PREDICTING WIN SHARES FOR THE NEXT SEASON

THE PROCESS / ALGORITHM



- Filter the data from the year 1999, 1003 minutes played, 57 games played
- Run models: Lasso, Linear Regression, Random Forest, Ridge, Elastic Net, Gradient Boost, ANN
- Compare Results

Filtering the Data

The Super Set of Variables

#Used Dplyr to lead Win Shares to the next season.

```
library(dplyr)
NBA=NBA %>%
  arrange(Year, Player) %>%
  group_by(Player) %>%
  arrange(Year) %>%
  #Leading WS to the following year
  mutate(WS_next_year=lead(WS),Year_Count= dplyr::row_number()) %>%
  ungroup() %>%
  filter(Year_Count > 1) %>%
  filter(!is.na(WS_next_year)) %>%
   select(Tm, Pos, Player, Year, Year_Count, Age, MP, G, PER, TS., X3PAr, FTr, ORB.,
TRB., DRB., STL., BLK., TOV., USG., OBPM, BPM, DBPM, VORP, FG, FGA, FG., X3P, X3PA,
X3P., X2P, X2PA, X2P., eFG., FT, FTA, FT., ORB, DRB, TRB, AST, STL, BLK, WS,
WS_next_year) %>%
filter(Year >= 1999) %>%
filter(MP >= 1003) %>%
filter(G >= 57)
```

Model for Next Year's WS

Lasso, Linear Regression, Random Forest Model, Ridge, Elastic Net, GBM, ANN

LASSO

```
library(glmnet)
set.seed(1)
WinShares_Next_Year = NBA$WinShares_Next_Year
lasso <- train(
WinShares_Next_Year ~. , data=NBA_train, method =
"glmnet",
trControl = trainControl("cv", number = 10),
tuneGrid = expand.grid(alpha = 1, lambda = 10^seq(-3, 1,
length =
100)))
lasso

coef(lasso$finalModel, lasso$bestTune$lambda)
predictionsL <- predict(lasso, NBA_test)
RMSE(predictionsL, NBA_test$WinShares_Next_Year)
R2(predictionsL, NBA_test$WinShares_Next_Year)
```

RMSE: 2.172565
R2: 0.5673374

LINEAR REGRESSION

```
set.seed(1)
library(caret)
train.control = trainControl(method = "cv", number = 10)
linearRegression <- train(WinShares_Next_Year ~., data
= NBA_train, method = "lm", trControl = train.control)
print(linearRegression)
summary(linearRegression)

#Checking the RMSE of the WinShares Next Year
predictionsLR= predict(linearRegression, NBA_test)
RMSE(predictionsLR, NBA_test$WinShares_Next_Year)
R2(predictionsLR, NBA_test$WinShares_Next_Year)
```

RMSE: 2.140996
R2: 0.6062071

Model for Next Year's WS

Lasso, Linear Regression, Random Forest, Ridge, Elastic Net, GBM, ANN

RANDOM FOREST

```
library(caret)
library(randomForest)
set.seed(1)
ctrl <- trainControl(method = "cv", number = 10)
grid_rf <- expand.grid(mtry = c(2, 4, 8, 16))

rf<- train(WinShares_Next_Year ~ ., data = NBA_train,
importance=T, method = "rf", trControl = ctrl, tuneGrid =
grid_rf)
varImp(rf)
predictionsRF <- predict(rf, NBA_test)
RMSE(predictionsRF, NBA_test$WinShares_Next_Year)
R2(predictionsRF, NBA_test$WinShares_Next_Year)
```

RMSE: 2.214575
R2: 0.5792119

RIDGE

```
set.seed(1)
ridge <- train(
WinShares_Next_Year ~., data = NBA_train, method =
"glmnet",
trControl = trainControl("cv", number = 10),
tuneGrid = expand.grid(alpha = 0, lambda = 10^seq(-3, 1,
length =
100)))

predictionsRidge <- predict(ridge,NBA_test)
RMSE(predictionsRidge,
NBA_test$WinShares_Next_Year)
R2(predictionsRidge, NBA_test$WinShares_Next_Year)
```

RMSE: 2.160676
R2: 0.599431

Model for Next Year's WS

Lasso, Linear Regression, Random Forest Model, Ridge, Elastic Net, GBM, ANN

ELASTIC NET

```
set.seed(1)
enet <- train(
WinShares_Next_Year ~., data = NBA_train, method =
"glmnet",
trControl = trainControl("cv", number = 10),
tuneGrid = expand.grid(alpha =seq(0,1, length=10),
lambda = 10^seq(-
3, 1, length = 100)))

predictionsElasticNet <- predict(enet, NBA_test)
RMSE(predictionsElasticNet,
NBA_test$WinShares_Next_Year)
R2(predictionsElasticNet,
NBA_test$WinShares_Next_Year)
```

RMSE: 2.146052
R2: 0.6046081

GBM

```
library(caret)
library(randomForest)
set.seed(1)
ctrl <- trainControl(method = "cv", number = 10)
grid_rf <- expand.grid(mtry = c(2, 4, 8, 16))

rf<- train(WinShares ~ ., data = NBA_train, importance=T,
method = "rf", trControl = ctrl, tuneGrid = grid_rf)
varImp(rf)

predictionsRF <- predict(rf, NBA_test)
RMSE(predictionsRF, NBA_test$WinShares)
R2(predictionsRF, NBA_test$WinShares)
```

RMSE: 2.20075
R2: 0.5833839

Model for Next Year's WS

Lasso, Linear Regression, Random Forest Model, Ridge, Elastic Net, GBM, ANN

ANN

```
library(keras)
set.seed(1)
model <- keras_model_sequential() %>%
layer_dense(units =150, activation = "relu", input_shape =
dim(NBA_train_re_x)[2]) %>%
layer_dropout(0.5) %>%
layer_dense(units = 1)

model %>%
compile(loss = "mse",
optimizer = "adam")

model %>% fit(NBA_train_rex,
NBA_train_rey,
batch_size=1000,
epochs = 500)
#Calculating RMSE on test data
predictions=model %>% predict(NBA_testx)

rmse= function(x,y){
return((mean((x - y)^2))^0.5)
}
rmse(predictions, NBA_testy)
R2(predictions, NBA_testy)
```

RMSE: 2.020652
R2: 0.6382309

SUMMARY RESAMPLES

MAE, RMSE, R2

```
MAE
       Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
L  1.596532 1.631928 1.682560 1.673864 1.699667 1.756370    0
E  1.608593 1.658396 1.673772 1.678621 1.709532 1.730372    0
G  1.646661 1.665475 1.685464 1.693879 1.711566 1.786192    0
R  1.626554 1.663242 1.682075 1.689784 1.728974 1.742934    0
RF 1.650868 1.685029 1.702841 1.708588 1.725543 1.780778    0

RMSE
       Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
L  2.119214 2.142207 2.184582 2.200744 2.248206 2.335300    0
E  2.145883 2.156852 2.201631 2.200224 2.240845 2.255018    0
G  2.139239 2.185971 2.222089 2.217331 2.246506 2.302499    0
R  2.149565 2.172583 2.209277 2.209158 2.248082 2.270339    0
RF 2.162579 2.202786 2.219517 2.226063 2.258232 2.285451    0

Rsquared
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
L  0.4818441 0.5254397 0.5598179 0.5565301 0.5846814 0.6542241    0
E  0.5100119 0.5390659 0.5500880 0.5534380 0.5566527 0.6131536    0
G  0.4932525 0.5199831 0.5385938 0.5472674 0.5665261 0.6155697    0
R  0.5094269 0.5383191 0.5457954 0.5500873 0.5491405 0.6053034    0
RF 0.4924231 0.5166680 0.5388595 0.5435849 0.5595005 0.6117547    0
```

Contrasts, lessons learned

While Section 1's prediction's of current Win Shares was more accurate, the usefulness of the problem is ambiguous. Section 2's next year's Win Shares had less accuracy for sure, but the model is more realistic and practical. Others have attempted to predict Win Shares based off current seasons, but this shows that it is not always the best parameter for prediction.

I wanted to demonstrate Section 1 to make a point on how to adjust, but feature Section 2 for the project.

The Artificial Neural Network had the best numbers in terms of RSME and R Squared, all the models were very close. It's important to note that I put the train data back together to train the ANN to compare to the other models.

Section 3

BONUS SECTION:
PREDICTING ALL STAR PLAYERS FROM THEIR ROOKIE SEASON

THE PROCESS / ALGORITHM

- Filter the data from the year 1999, inner join players that became All Stars
- Run models: Lasso, Ridge, Enet, Logistic for classification
- Compare results

Filtering the Data

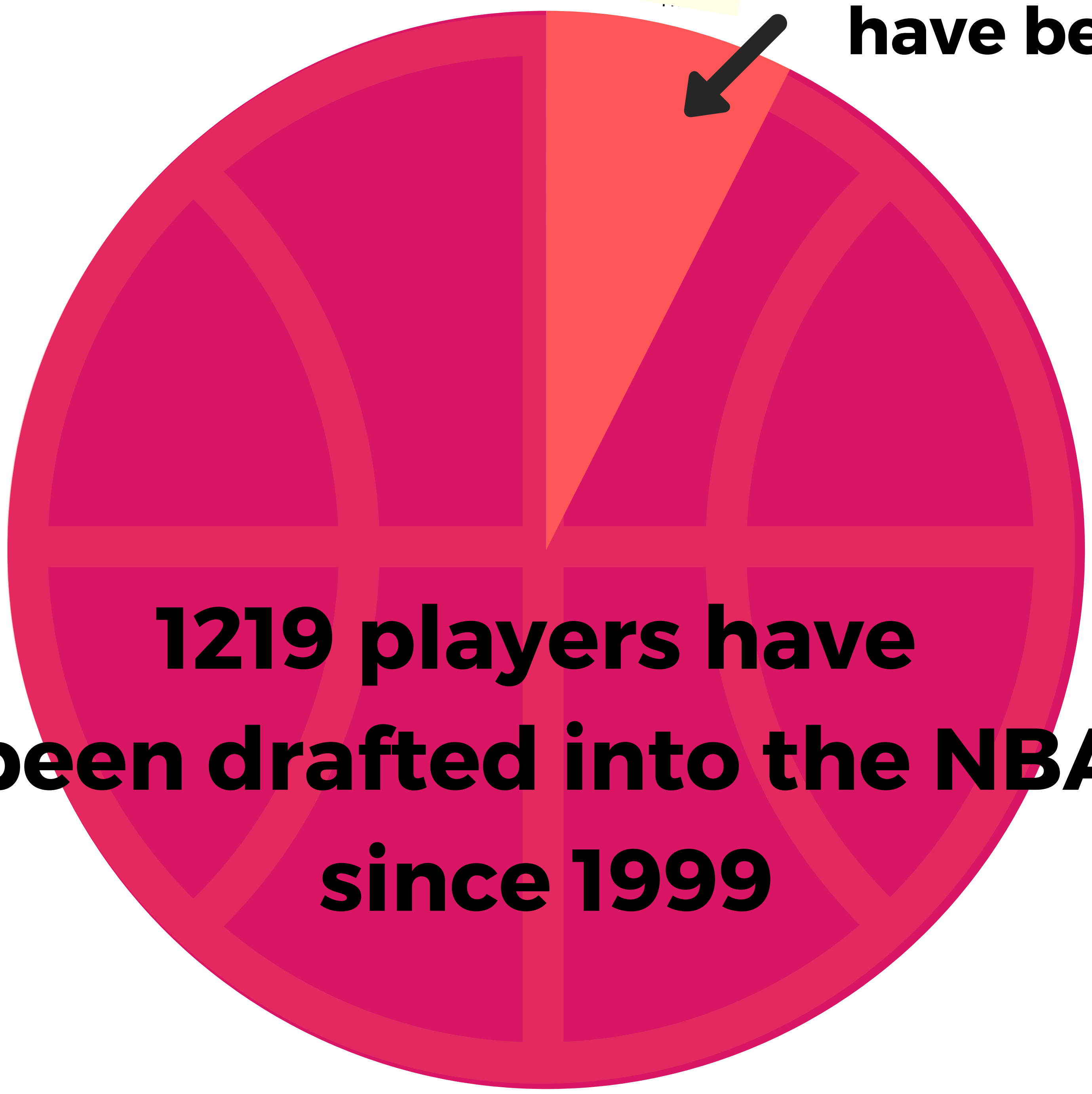
The Super Set of Variables

```
#Filtering for all players in their first rookie season since 1999
library(dplyr)
NBA=NBA %>%
  arrange(Year, Player) %>%
   mutate(Player =as.character(Player)) %>%
mutate(Player = case_when(stringr:: str_detect(Player, "Yao Ming*") ~ "Yao
Ming",TRUE~Player))%>%
  mutate(Player =as.character(Player)) %>%
mutate(Player = case_when(stringr:: str_detect(Player, "Nikola V") ~ "Nikola
Vucevic",TRUE~Player))%>%
   mutate(Player =as.character(Player)) %>%
mutate(Player = case_when(stringr:: str_detect(Player, "Nikola J") ~ "Nikola
Jokic",TRUE~Player))%>%
  group_by(Player) %>%
  arrange(Year) %>%
  mutate(Year_Count= dplyr::row_number()) %>%
  ungroup() %>%
   filter(Year_Count == 1) %>%
 filter(Year >= 1999)
```

#Used Dplyr to pick rookies from 1999-2019

Exploratory Data

Among 1219 players, 101 All Stars have been selected (8.2%)



1219 players have been drafted into the NBA since 1999

Can we predict who will be an All Star based off their rookie season?

Models for All Star Prediction

Lasso, Ridge, Elastic Net, Logistic

LASSO

```
table(AllStarPred_test$AllStar)
predict.lasso = predict(lasso, AllStarPred_test)
confusionMatrix(predict.lasso, AllStarPred_test$AllStar)

lasso_predictions_prob=predict(lasso, AllStarPred_test,
type="prob")
head(lasso_predictions_prob)

pred_lasso = prediction(lasso_predictions_prob$`1`,
AllStarPred_test$AllStar)
performance(pred_lasso, measure = "auc")@y.values
perf <- performance(pred_lasso, measure = "tpr", x.measure =
"fpr")

#Plotting the ROC Curve
plot(perf, col = "blue")
```

AUC: 0.8371736
Accuracy : 0.94

RIDGE

```
ridge_predictions_prob=predict(ridge, AllStarPred_test,
type="prob")

pred_ridge = prediction(ridge_predictions_prob$`1`,
AllStarPred_test$AllStar)
performance(pred_ridge, measure = "auc")@y.values

perfR <- performance(pred_ridge, measure = "tpr",
x.measure = "fpr")

#Plotting the ROC Curve
plot(perfR, col = "green")
```

AUC: 0.8379416
Accuracy : 0.935

Models for All Star Prediction

Lasso, Ridge, Elastic Net, Logistic

ENET

```
enet_predictions_prob=predict(enet, AllStarPred_test,
type="prob")

pred_enet = prediction(enet_predictions_prob$`1`,
AllStarPred_test$AllStar)
performance(pred_enet, measure = "auc")@y.values


perfE <- performance(pred_enet, measure = "tpr",
x.measure = "fpr")

#Plotting the ROC Curve
plot(perfE, col = "purple")
```

AUC: 0.8379416
Accuracy : 0.935

LOGISTIC

```
library(pROC)

predict.logistic <- predict(model.logistic, AllStarPred_test,
type="response")
predict.logistic.label = factor(ifelse(predict.logistic > .1, "Yes",
"No"))
actual.label = AllStarPred_test$AllStar
table(actual.label, predict.logistic.label)
ROC <- roc(AllStarPred_test$AllStar, predict.logistic)

#Plotting the ROC Curve
ROCplot = plot(ROC, col = "red")

#AUC= The area under the curve
auc(ROC)
```

AUC: 0.8015
Accuracy: 0.865

All Stars, lessons learned

We seem to be able to reasonably predict if a player will become an All Star caliber player from their first season, their rookie year based off the AUC measurements. Ridge and Elastic net had the best numbers in terms of Area Under the Curve, Logistic had the lowest AUC number. The numbers were all fairly close.

An important detail to remember is that real-life All Star voting is a combination of fan votes, NBA selection, and coaches' votes. The rules and ratios of these inputs tend to change over time. Some players get selected for the right reasons, some for unpredictable ones.

Regardless, these amazing All Stars are a part of NBA history. At their peaks, they were among the best of the best and mattered in the story of basketball.

IV

Appendix, Notes

- **Full codes and details** are in the R notebook and html preview. The code in this presentation are only snippets for demonstration.

- For Section 1, I only produced 3 models to demonstrate how current season Win Shares only solves part of the problem. Section 2 has several more models to observe.

- For Section 3, I created a separate list of All Stars from NBA history by listing them in a CSV file. I copy and pasted them into a CSV file by name alone. Through **basketballreference.com**, I can verify that this list is accurate. I used it to inner join the main data set from kaggle.com. That list is from the following link: **https://en.wikipedia.org/wiki/List_of_NBA_All-Stars**

- In the zipped folder, I included my hmtl file and tuning flags R-script named 'test'.

- The attached project report also has a references page.