College Basketball Wins Predictability

STA6543 Algorithms II

March 26, 2020

Andrew Locker, Matthew Forey, Nathan Keckley, Ricardo Munoz

***** Note***

Due in large part to the Coronavirus (COVID-19) the 2020 edition of the college basketball tournament has been canceled. Since the teams that were supposed to participate in the tournament were not officially selected, the group decided to look at various online resources to closely replicate what teams were going to play.

Abstract

Predicting the outcome of a basketball game can be a difficult task, especially when there are so many factors to be considered. Still, predicting the outcome of games during the NCAA Men's Basketball tournament is a popular tradition for basketball fans each year, despite the challenges it presents. Our group will propose an approach comparing the accuracies of 10 different models with data from college basketball games played this year, and select the best performing model to try to predict the results of the 2020 Men's NCAA Tournament.

Background

Every March, basketball fans around the country do their best to predict the outcomes of college basketball games during the annual NCAA Men's Basketball Tournament. Fans compete to predict the most wins correctly by filling out a bracket for all 63 games that are played over 3 weeks. Historically, many people have been able to predict a large number of early round games, but it is extremely rare for anyone to continue picking correct winners as upsets occur and the variations of possible matchups increase in later rounds of the tournament.

There are many reasons why one basketball team might be selected to win over another team when predicting who will win a game. The basketball statistics that teams accumulate throughout the season can create a descriptive framework of how a team may perform better in some areas than others. For example, statistics like points per game or field goal percentage can give a good description of how well a team performs on offense, just as statistics like points allowed per game or forced turnovers could be indicators of how strong a team is on defense.

We will present a model that performs well using team statistics variables for two opposing teams to predict game winners for all 63 predicted matchups in the 2020 NCAA Basketball tournament.

Variable Introduction and Definitions

The selected dataset consisted of scraping the results from the NCAA website of every Division 1 Men's Basketball game from the start of the 2019 season to March 8, 2020 (the end of the regular season for most Division 1 teams), which included 10,518 observations. This allowed us to have a complete representation of the observed outcomes from the teams we are trying to predict will succeed in the tournament. In addition to each matchup, we also downloaded the basketball statistics of all 353 Division 1 teams and joined these statistics to the observed season matchups by team name. This included both the selected team's and their opponent's accumulated season stats from the website Basketball Reference as described in the table below.

Variable Name	Description	Variable Name	Description
year	Year the game was played	Away_W	Away wins on season
month	Month the game was played	Away_L	Away losses on season
day	Day the game was played	Tm.Total_Pts	Total points scored on season
team	NCAA Division 1 school	Tm.Pts_Allowed	Total points allowed on season
opponent	NCAA Division 1 school opponent	Pace	Estimate of possessions per 40 minutes
teamscore	Points scored in individual matchup	ORtg	Offensive Rating - an estimate of points scored per 100 possessions
oppscore	Opponent points in individual matchup	Ftr	Free Throw Attempt Rate: Number of FT attempts per FG attempts
D1	2 = Division 1 opponent 1 = Division 2 opponent	3PAr	3-point Attempt Rate: Percentage of Field Goal Attempts that are 3-point Field Goal Attempts
Win	0 = teamscore < oppscore 1 = teamscore > oppscore	TS%	True Shooting Percentage - A measure of efficiency using weighted calculations of 2-point field goals, 3-point field goals and free throws
G	Team total season games played	TRB%	Total Rebound Percentage - Estimate of the percent of available rebounds grabbed
Overall_W	Team total season wins	AST%	Assist Percentage - An estimate of the percentage of teammate field goals assisted while on the court
Overall_L	Team total season losses	STL%	Steal Percentage - An estimate of the percentage of opponent possessions that end with a steal
W-L%	Team season win-loss percentage	BLK%	Block Percentage - An estimate of the percentage of opponent two-point field goal attempts blocked
SRS	Simple Rating System: a calculated rating system taking into account average point differential and strength of schedule (0 = average)	eFG%	Effective Field Goal Percentage - adjusts field goal percentage for the fact that a 3-point field goal is worth one more point than a 2-point field goal
SOS	Strength of schedule: a rating of a team's schedule difficulty (0=average)	TOV%	Turnover Percentage - an estimate of turnovers per 100 possessions
Home_W	Home wins on season	ORB%	Offensive Rebound Percentage - an estimate of the percentage of available offensive rebounds grabbed
Home_L	Home losses on season	FT/FGA	Free Throws Per Field Goal Attempt

An additional variable, "Win Margin" was created by taking the difference of "teamscore" and "oppscore". Our response variable needed to be a predicted probability that the selected team would win against their opponent, so the "Win" column served as the dependent variable, and the selected predictor statistics were chosen based on their levels of significance.

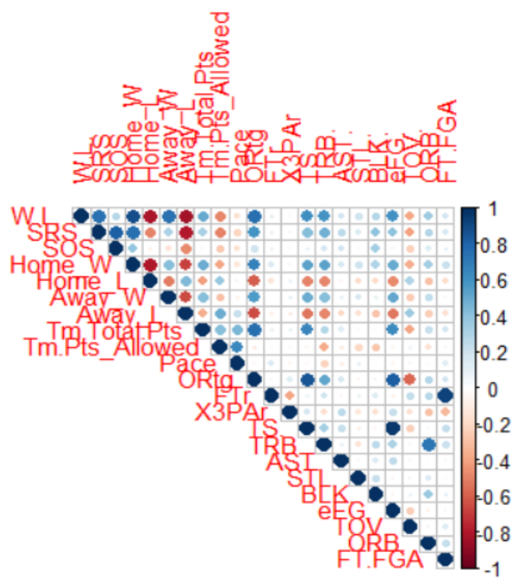
Preprocessing

Combining the two datasets required much cleaning before the data could be used. The first step required rewriting team names from the scraped NCAA website dataset to match team names on the other dataset from Basketball Reference. By manually changing the team names of 191 basketball teams on one dataset to match the correct name on the other dataset, we could append the correct basketball statistics to the correct basketball team.

The NCAA games dataset also included games where a Division-1 school played against a Division-2 school. Because we were only interested in statistics for games between two Division-1 opponents, the games that featured a Division-2 opponent were removed from the dataset.

Finally, the "Win" column was originally represented by "0" for a loss and "1" for a win, so we renamed the values in that column to substitute "Loss" for "0" and "Win" for "1", and then set the column data as factors, as it served as our dependent variable.

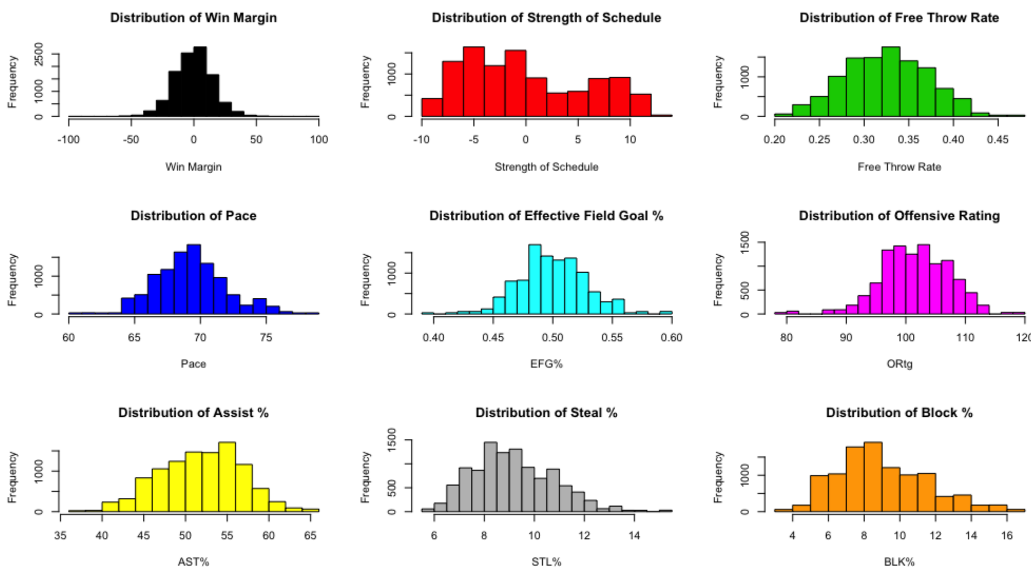
Between-Predictor Correlations



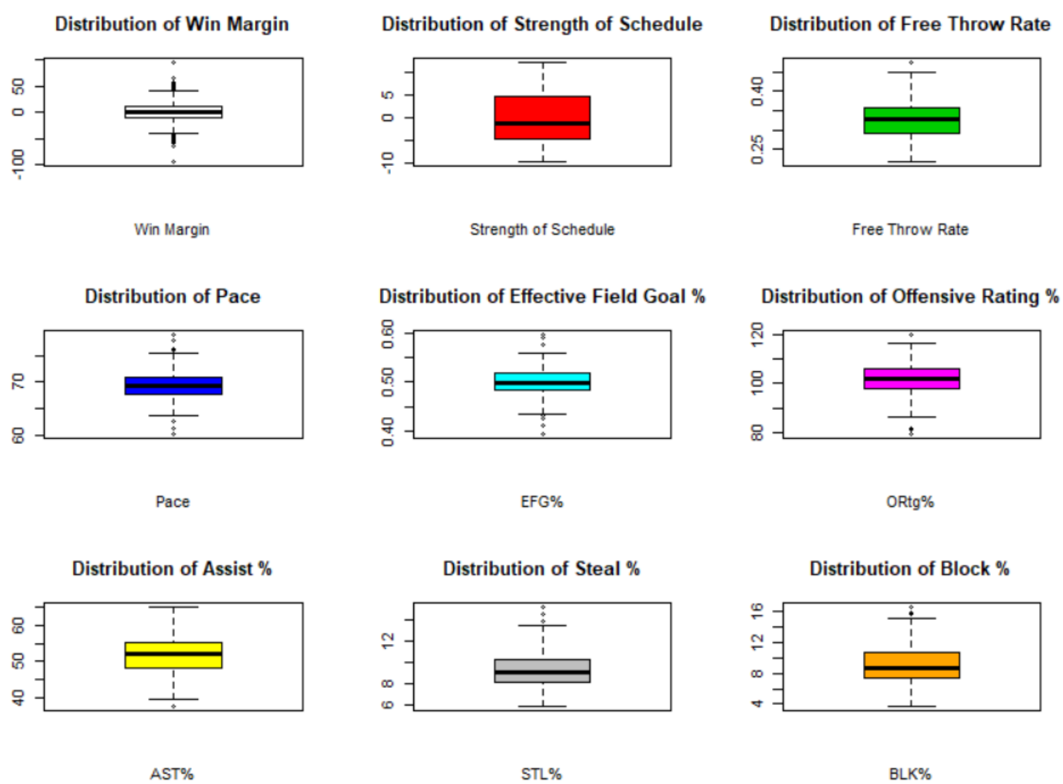
In the correlation matrix plot seen above, the dark blue dots indicate a strong positive correlation between predictors, and the dark red dots indicate a strong negative correlation between predictors. We can see from the visual of all 22 predictors that free-throws per field goal attempt has a strong positive correlation with free-throw rate, so free-throws per field goal attempt was not considered for the model. Similarly, true shooting percentage and effective field goal percentage also had a strong positive correlation, so true shooting percentage was not considered for the model either.

Transforming Data/ Normality/Skewness/Boxplots

Normality -



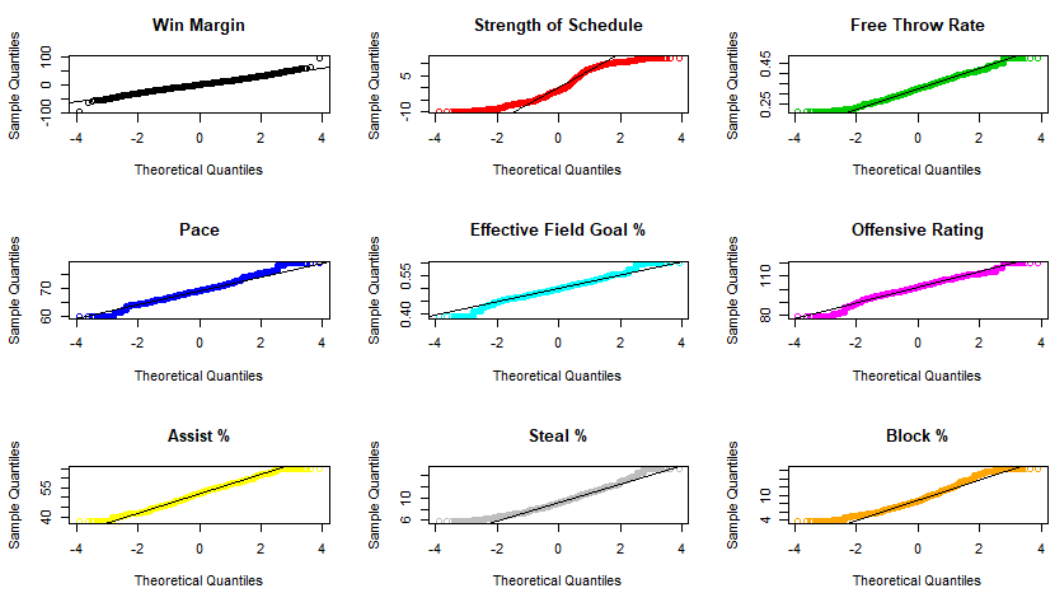
Describe normality



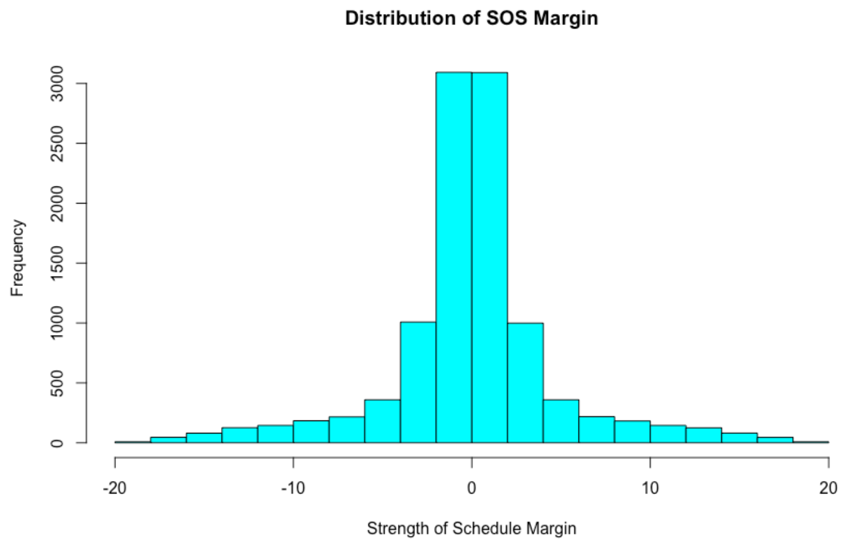
As seen in our distributions above, most of our data is for the most part, normal. There are a few outliers, some more than others such as Distribution of Win Margin, Pace, and Effective Field Goal %, but in general we are seeing normal distributions inside of the first quartiles, and little skewness.

Predictor	**Skewness**	**Category**
Win Margin	4.60e-05	Symmetric
Strength of Schedule	0.39385	Slightly Skewed
Free Throw Rate	0.09909	Symmetric
Pace	0.26607	Slightly Skewed
Effective Field Goal %	0.04918	Symmetric
Offensive Rating	-0.3108	Slightly Skewed
Assist %	-0.0943	Symmetric
Steal %	0.49829	Slightly Skewed
Block %	0.55113	Slightly Skewed

QQ Plots



In our displayed 9 variables above, 8 of them are not skewed and normally distributed. Strength of schedule is our most skewed with a skewness value of 0.39. We end up using the margin of strength of schedule between the two teams, and this alleviates the skewness issue and it is now normally distributed.



After checking variable distributions and cleaning the data set, we needed to determine which basketball team stats variables were most important for our prediction model. The six variables that were revealed to be the most significant predictors for our models were Strength of Schedule, Free Throw Rate, Pace, Effective Field Goal Percentage, Turnover Percentage, and Offensive Rebound Percentage. Each of these statistics were used to create six new variables which held the margin between the statistic of that team and its opponent (Strength of Schedule Margin, Pace Margin, Free Throw Rate Margin, etc.). These six new margin variables captured the difference in stats between the two teams and were used for building each model.

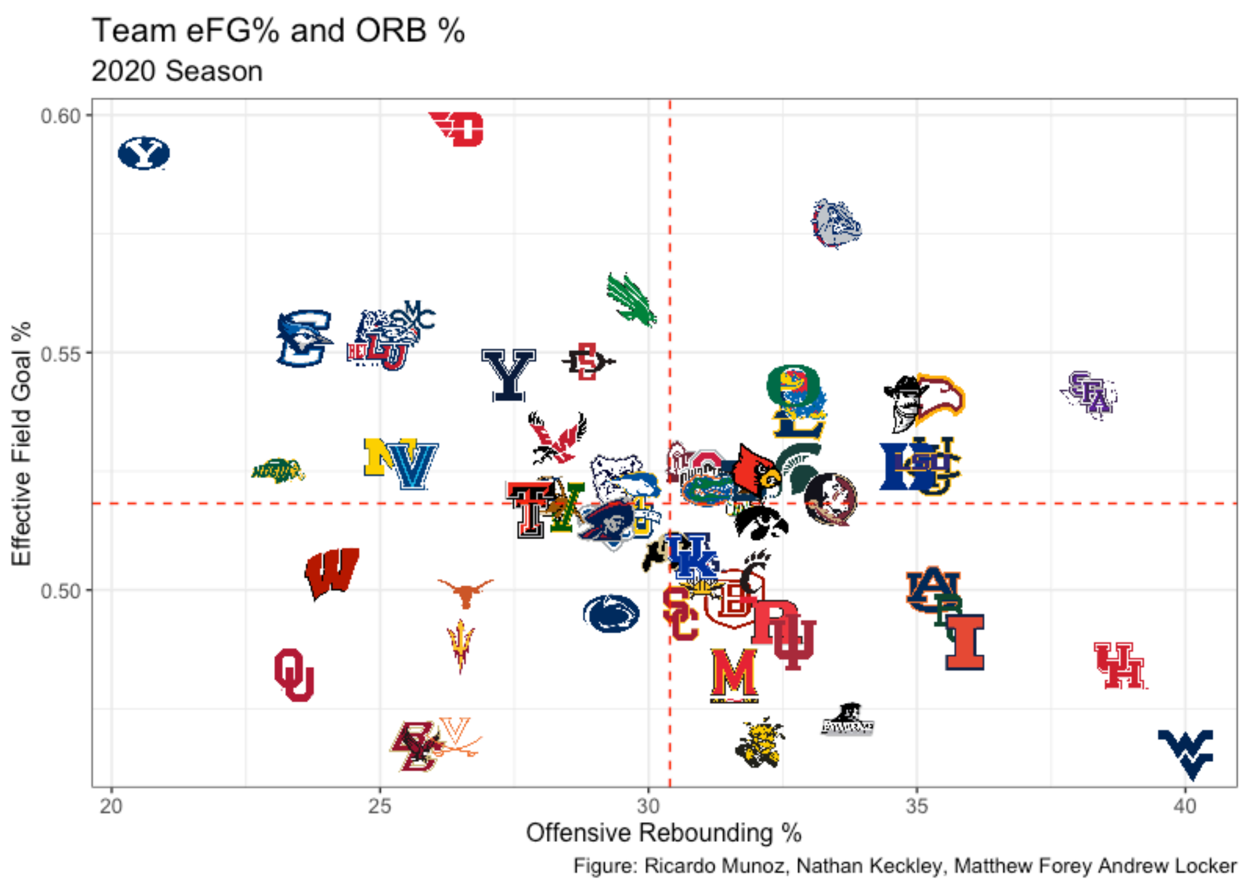
Data Splitting

To evaluate our dataset, we wanted to find a way to split the data to ensure that we were creating the best models. In practice, the training set is usually used to build our different models, and the testing set used to validate the performance of the models. After some consideration the group decided that it would be best if we used the first four months of game data from this season as our training set and the last two months as our testing set. When the split was conducted, we ended up with 8,760 observations in our train set and 1,758 observations in our test. The 83-17 split was right about what we were hoping to get, we just needed to ensure that the same distribution existed. Afterwards, we built our models on our training set and compared the respective results to the testing set. If the group was content with the results, we would then use the model to predict the tournament matchups.

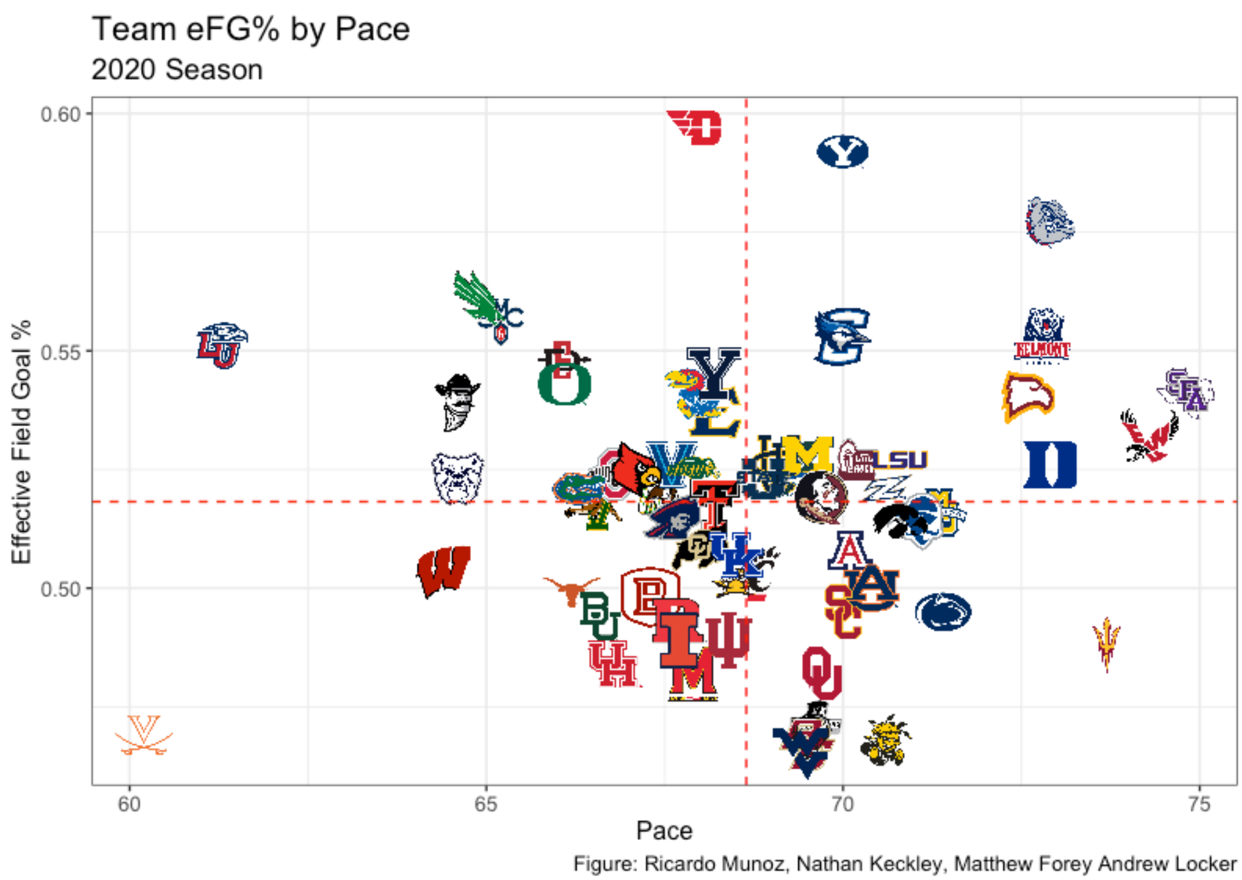
Team Statistical Graphs

The following graphs were created to try and get a better understanding of the relationships between different stats for every team in the tournament. The only variables used were the ones deemed to be significant by our models.

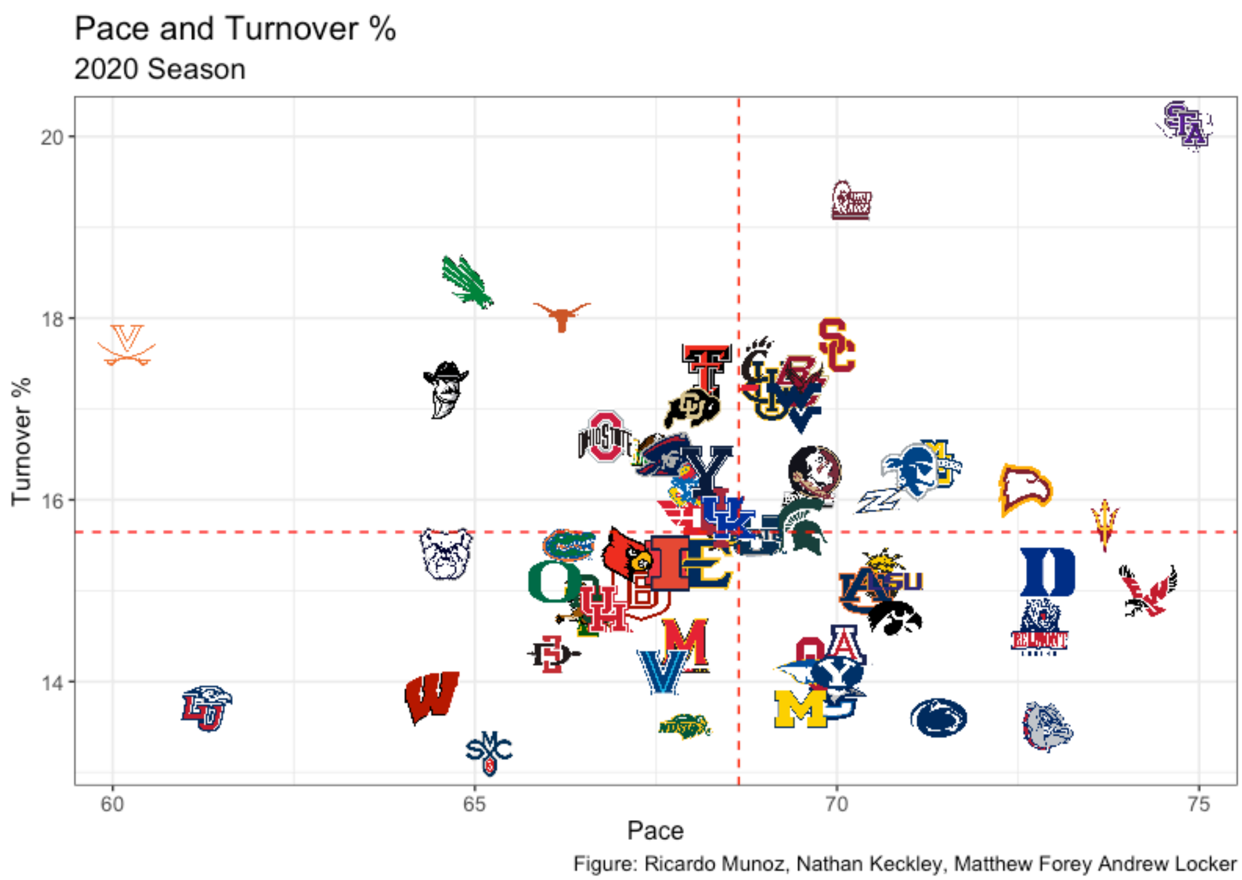
The graph below shows the effective field goal percentage and the offensive rebounding percentage for every team. The reasoning behind this graph was that some teams may not be effective shooters but gain second chance possessions due to their rebounding. In the bottom right corner, we see that West Virginia is among the worst teams in terms of eFG % but rebounds the ball at a very high rate. In contrast we see BYU in the top left corner with a very high eFG%. BYU in the past is known to produce very good shooters but typically struggle to recruit big men that facilitate offensive rebounding.



Another metric that the team wanted to explore was how did pace correlate with how well teams shot the ball. Although we don't see any linear correlation, we a good understanding of how different teams approach the game. The team that stands out the most is Virginia in the bottom left corner. Virginia, the defending national champion, is known to play a really slow game a really on rebounding and defense. In contrast we do see that some of the better teams, Duke and Gonzaga actually play a much faster game. In reality we don't learn much from this graph other than pace doesn't really dictate how well you can shoot the ball. We see a decent number of teams that play fast (bottom right) but don't necessarily take efficient shots.



The last graph we wanted to explore dealt with how pace could affect turnover percentage. Our train of thought was that if a team plays quicker than they are more likely to turn the ball over and thus limit their number of possessions. Our hypothesis again fails because it appears there is no true relationship between the two variables. Other than Stephen F. Austin, which appears to be an outlier, turns the ball over much more due to their pace.

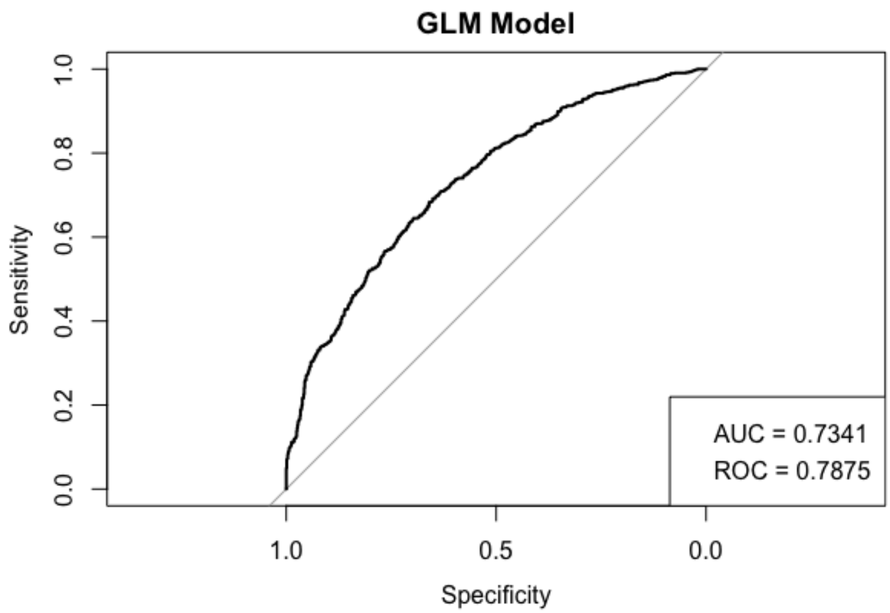


From the graphs above we learn that there is no one perfect way to play the game. It appears that how fast you play has no direct effect on how well you shoot the ball or how often you turn the ball over. In the current world basketball is becoming a very high paced game and it will be interesting to revisit this in a few years and see if the results have changed.

Model Building

GLM Model:

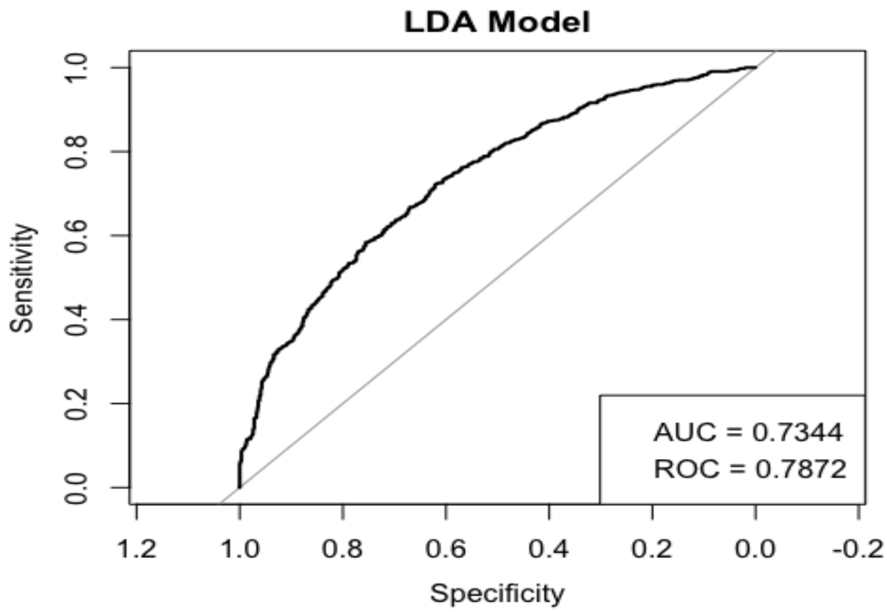
	ROC	AUC	Training Accuracy	Testing Accuracy
GLM	**.7875**	**.7341**	**0.7003**	**0.6661**




```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.005100   0.024839   0.205    0.837
SOS_Mar      0.175494   0.006652  26.383  < 2e-16 ***
FTr_Mar      2.713956   0.418181   6.490 8.59e-11 ***
Pace_Mar    -0.046430   0.007004  -6.629 3.38e-11 ***
eFG_Mar     20.402136   0.767367  26.587  < 2e-16 ***
TOV_Mar     -0.168614   0.012136 -13.893  < 2e-16 ***
ORB_Mar      0.081083   0.004827  16.799  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

LDA Model

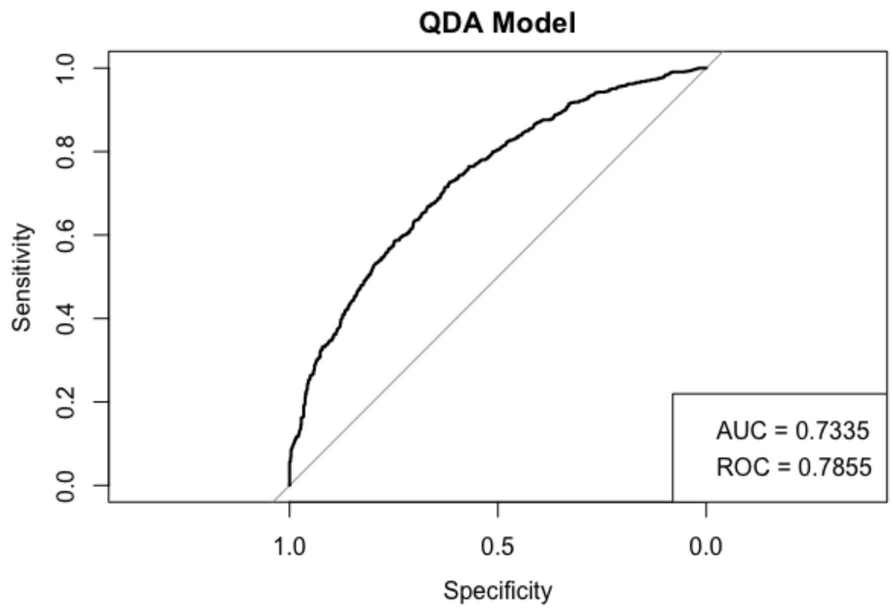
	ROC	AUC	Training Accuracy	Testing Accuracy
LDA	**.7872**	**.7344**	**0.7074**	**0.6672**




```
Linear Discriminant Analysis

8760 samples
   6 predictor
   2 classes: 'Loss', 'Win'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results:

  ROC        Sens       Spec
  0.7872555  0.7035165  0.7019143
```

QDA Model

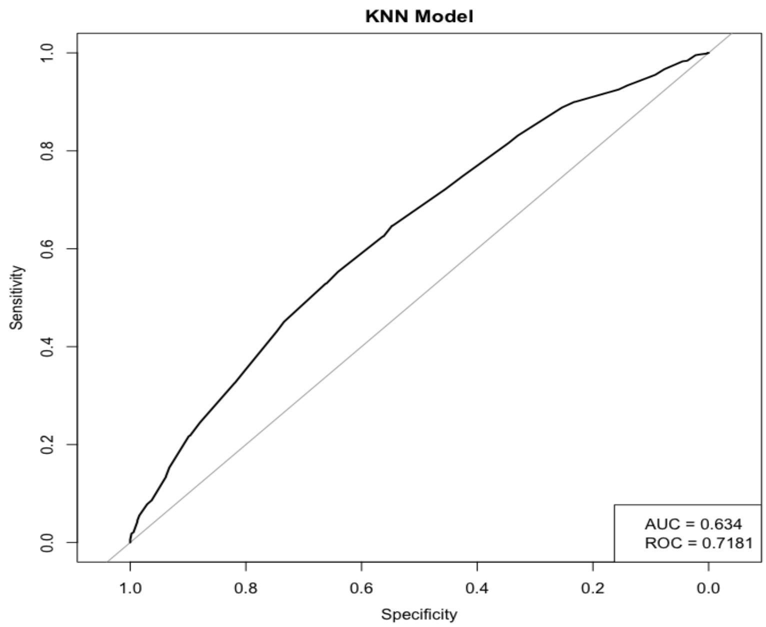
	ROC	AUC	Training Accuracy	Testing Accuracy
QDA	**.7855**	**.7335**	**0.7064**	**0.6678**




```
Quadratic Discriminant Analysis

8760 samples
   6 predictor
   2 classes: 'Loss', 'Win'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results:

  ROC        Sens       Spec
  0.7855829  0.7032601  0.7011486
```

KNN Model

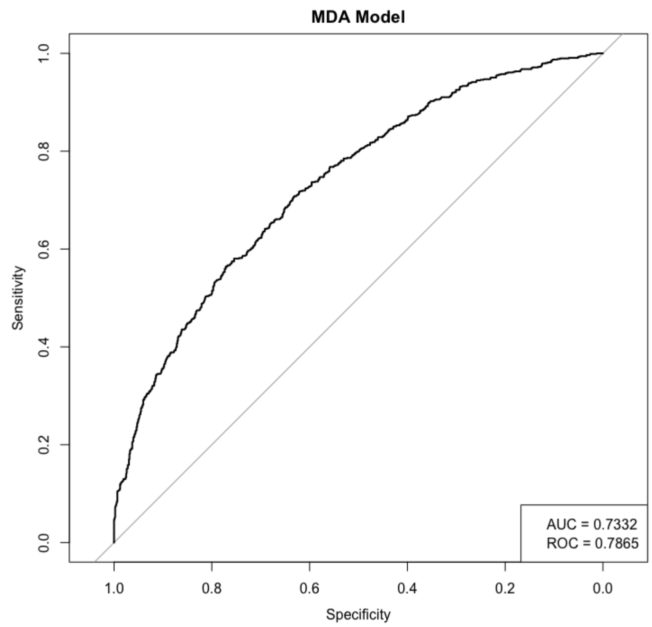
	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
KNN	**.7181**	**0.634**	**0.6979**	**0.6007**	**K = 20**




```
k-Nearest Neighbors

8760 samples
   6 predictor
   2 classes: 'Loss', 'Win'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results across tuning parameters:

  k   ROC        Sens       Spec
  20  0.7181127  0.6504029  0.6556062

ROC was used to select the optimal model using the
 largest value.
The final value used for the model was k = 20.
```

MDA Model

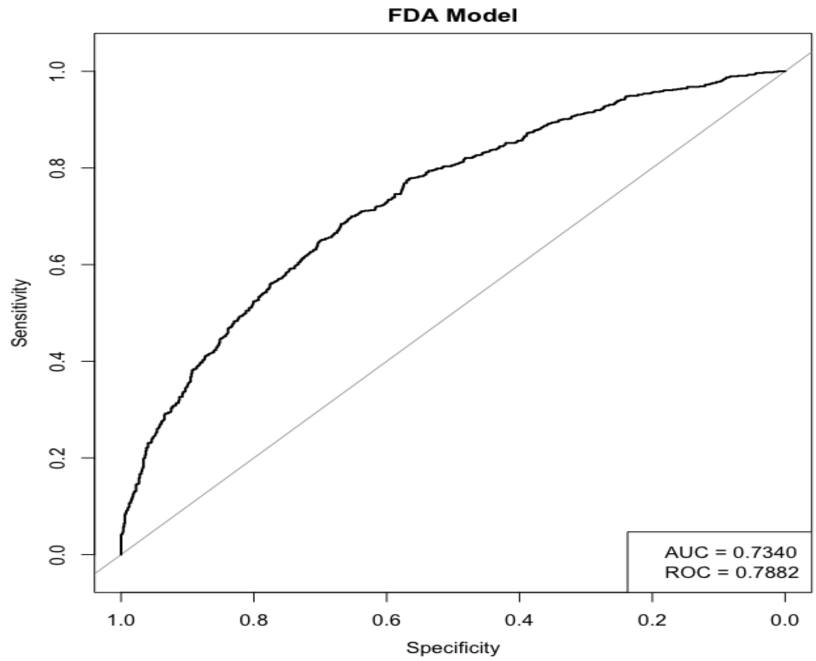
	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
MDA	**0.7865**	**0.7332**	**0.7054**	**0.6615**	**Subclass = 2**




```
  subclasses  ROC        Sens       Spec
  2           0.7865044  0.7044322  0.7023519
  3           0.7846769  0.7027473  0.7001276
  4           0.7845253  0.7030037  0.7004923

ROC was used to select the optimal model using the largest value.
The final value used for the model was subclasses = 2.
```

FDA Model

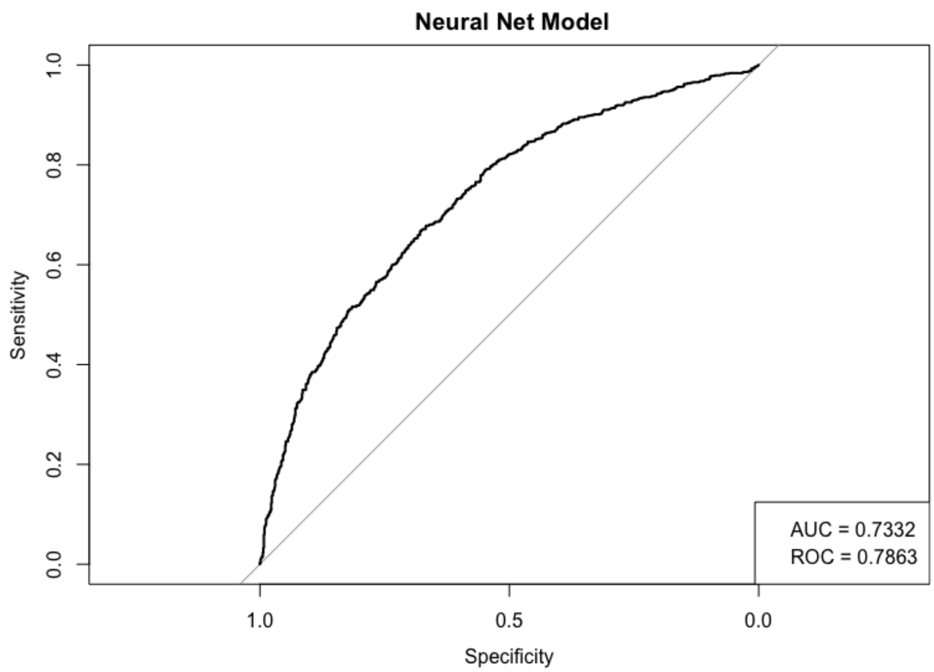
	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
FDA	**0.7882**	**0.7340**	**0.7340**	**0.6746**	**Nprune =14**




```
  nprune  ROC        Sens       Spec
   2      0.6534059  0.6237363  0.5712671
   8      0.7842148  0.7001099  0.7043209
  14      0.7882667  0.7041758  0.7075661

Tuning parameter 'degree' was held constant at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were degree = 1 and nprune = 14.
```

Neural Net Model

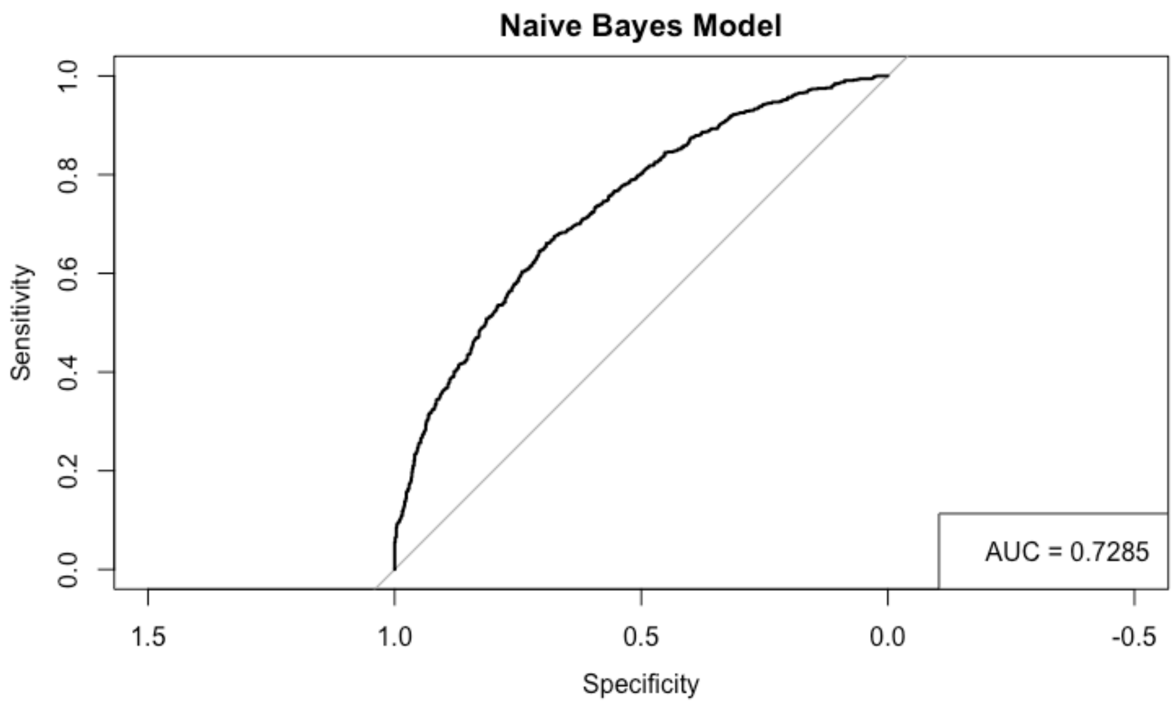
	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
NNet	.78638	0.7332	0.7139	0.6718	Size = 8 Decay = .1




```
Neural Network

8760 samples
   6 predictor
   2 classes: 'Loss', 'Win'

Pre-processing: centered (6), scaled (6), spatial sign transformation (6)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results across tuning parameters:

  size  decay  ROC        Sens       Spec
   8    0.10   0.7863867  0.7108791  0.7178122
   8    0.50   0.7841693  0.7044322  0.7110665
   9    0.00   0.7791183  0.7021245  0.7108478
   9    0.01   0.7835396  0.7068132  0.7153692
   9    0.10   0.7861692  0.7089377  0.7159891
   9    0.50   0.7840278  0.7038095  0.7102644
  10    0.00   0.7755882  0.6942491  0.7101185
  10    0.01   0.7830910  0.7025641  0.7148222
  10    0.10   0.7852361  0.7080586  0.7162078
  10    0.50   0.7846813  0.7050549  0.7108113

ROC was used to select the optimal model using the largest value.
The final values used for the model were size = 8 and decay = 0.1.
```

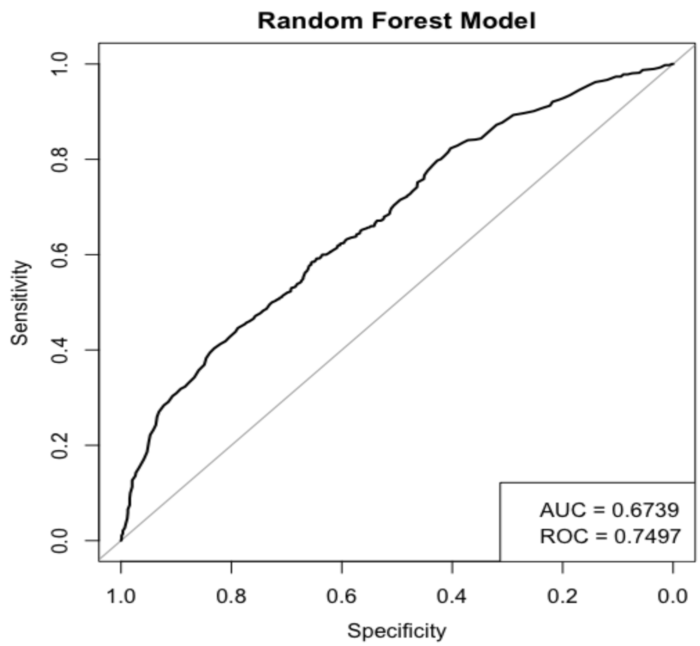
Naïve Bayes

	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
Naïve Bayes	**N/A**	**.7354**	**0.7061**	**0.6746**	**Accuracy = .7058 Kappa = .4116 (moderate)**



Random Forest Model

	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
RF	**0.7497**	**0.6739**	**0.9325**	**0.6138**	**Mtry =8**

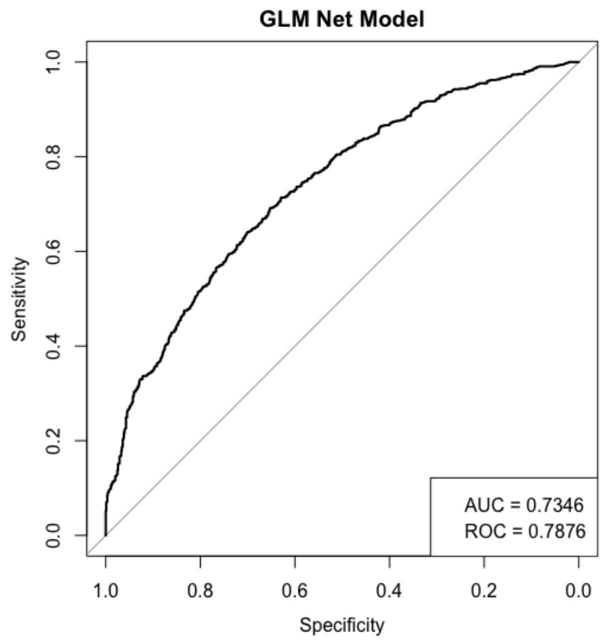



```
Random Forest

8760 samples
   8 predictor
   2 classes: 'Loss', 'Win'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   1    0.7405203  0.6998901  0.6928715
   8    0.7497694  0.6949451  0.6852871
  16    0.7492572  0.6946520  0.6822972
  23    0.7494957  0.6933333  0.6851048
  31    0.7494975  0.6958608  0.6844850
  38    0.7494124  0.6946886  0.6825524
  46    0.7491475  0.6957875  0.6837922
  53    0.7493846  0.6943956  0.6858341
  61    0.7493413  0.6950916  0.6852871
  69    0.7497414  0.6960073  0.6855424

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 8.
```

****Here we see the Random Forest model being biased towards the training set*

GLM Net Model

	ROC	AUC	Training Accuracy	Testing Accuracy	Notes
GLMNet	**0.7876**	**0.7346**	**0.7064**	**0.6684**	**Alpha = 0** **Lambda = 0.01487**

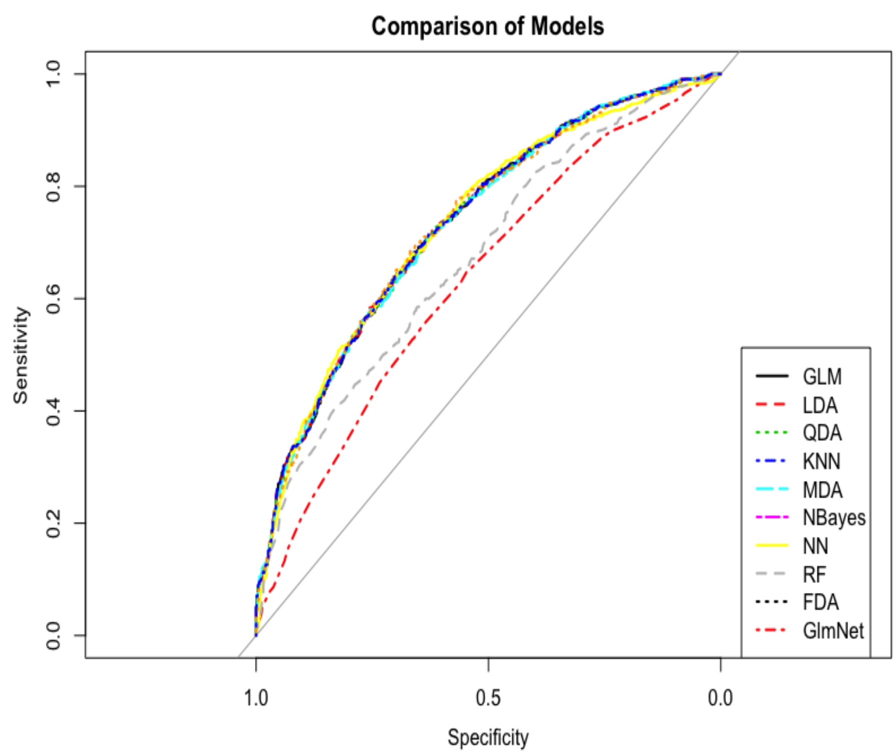


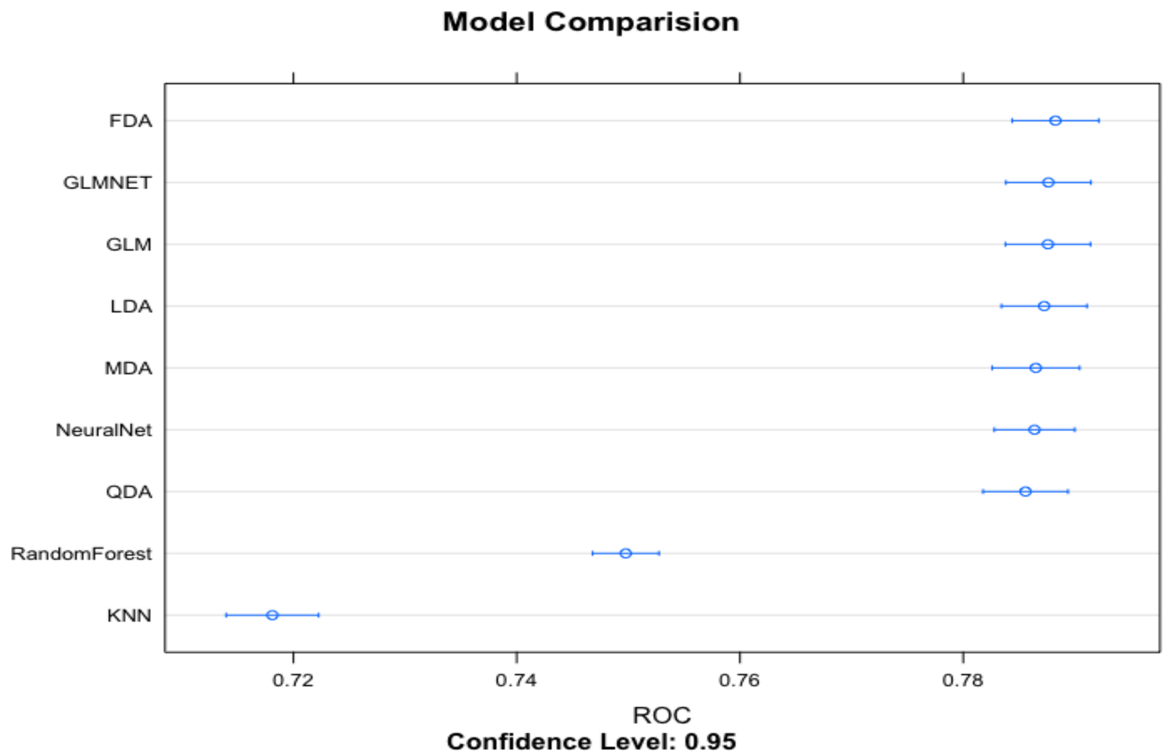

```
glmnet

8760 samples
   6 predictor
   2 classes: 'Loss', 'Win'

No pre-processing
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 6571, 6571, 6571, 6571, 6571, 6571, ...
Resampling results across tuning parameters:

  alpha  lambda      ROC        Sens       Spec
  0.0    0.01000000  0.7876321  0.7030037  0.7031176
  0.0    0.01487179  0.7876321  0.7030037  0.7031176
```

Model Selection

Model	**Train AUC**	**Test AUC**	**Train Accuracy**	**Test Accuracy**	**Notes**
GLM	.7875	.7341	0.7003	0.6661	
LDA	**.7872**	**.7344**	**0.7074**	**0.6672**	Final model chosen
QDA	.7855	.7335	0.7064	0.6678	
KNN	.7181	.634	0.6979	0.6007	K = 20
MDA	.7865	.7332	0.7054	0.6615	Subclass = 2
FDA	.7882	.7340	0.7340	0.6746	Nprune =14
Naïve Bayes		.7354	0.7061	0.6746	
Neural Net	.78638	0.7332	0.7139	0.6718	Size = 8 Decay = .1
Random Forest	0.7497	0.6739	0.9325	0.6138	Mtry =8
GLM Net	0.7876	0.7346	0.7064	0.6684	Alpha = 0,Lambda=0.01487

Comparison of Models





Most of our models, apart from the Random Forest and KNN models, performed at similar levels as seen by the graphs above. Taking this into consideration, we chose to proceed with our LDA model. An LDA model will perform better than a logistic regression due to the assumption of normal distribution and common variance in each class.

Final Model

Model	Train ROC	Train Accuracy	Test Accuracy	Notes
LDA	.7872	0.7074	0.6672	Final model chosen

Training Set

	Reference	
Prediction	**Loss**	**Win**
Loss	**3086**	**1278**
Win	**1285**	**3111**

Testing Set

	Reference	
Prediction	**Loss**	**Win**
Loss	**592**	**289**
Win	**296**	**581**

Tournament Simulation
First Round Matchups

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Siena	0.0458	0.9542
Michigan State	UC-Irvine	0.1254	0.8745
Creighton	Belmont	0.0670	0.9329
Wisconsin	Vermont	0.1376	0.8623
Auburn	East Tennessee State	0.2625	0.7374
West Virginia	Wichita State	0.2191	0.7808
Virginia	**Texas Tech**	0.7496	0.2503
Florida	USC	0.2163	0.7836
Dayton	Winthrop	0.1126	0.8874
Villanova	North Dakota State	0.1139	0.8860
Duke	Arkansas-Little Rock	0.1254	0.8745
Maryland	Akron	0.1207	0.8793
Butler	New Mexico State	0.1867	0.8133
Iowa	Cincinnati	0.2706	0.7294
Michigan	Utah State	0.3550	0.6450
Saint Mary's (CA)	Oklahoma	0.2701	0.7299
Baylor	Boston University	0.0888	0.9112
Florida State	Northern Kentucky	0.1071	0.8928
Seton Hall	Hofstra	0.1878	0.8121
Louisville	Yale	0.1827	0.8172
Ohio State	Stephen F. Austin	0.0947	0.9053
Penn State	Texas	0.3012	0.6988
Providence	**Rutgers**	0.6248	0.3752
Arizona	**LSU**	0.5908	0.4092
Gonzaga	Robert Morris	0.0292	0.9707
San Diego State	Bradley	0.2074	0.7925
Kentucky	Eastern Washington	0.1782	0.8218
Oregon	North Texas	0.1942	0.8058
BYU	Liberty	0.1485	0.8514
Houston	**Indiana**	0.5565	0.4434
Illinois	Arizona State	0.2063	0.7937
Colorado	**Marquette**	0.5725	0.4274

Second Round Matchups

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Florida	0.3221	0.6778
Michigan State	Texas Tech	0.2729	0.7270
Creighton	West Virginia	0.4064	0.5935
Wisconsin	**Auburn**	0.5859	0.4140
Dayton	Saint Mary’s (CA)	0.4165	0.5834
Villanova	**Michigan**	0.5283	0.4716
Duke	**Iowa**	0.5086	0.4913
Maryland	**Butler**	0.5473	0.4526
Baylor	**LSU**	0.5553	0.4446
Florida State	Rutgers	0.4850	0.5149
Seton Hall	Penn State	0.4610	0.5382
Louisville	**Ohio State**	0.5441	0.4558
Gonzaga	Marquette	0.2923	0.7076
San Diego State	**Illinois**	0.5443	0.4556
Kentucky	**Indiana**	0.5613	0.4386
Oregon	BYU	0.3527	0.6472

Third Round Matchups
"Sweet 16"

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Auburn	0.2940	0.7059
Michigan State	Creighton	0.4725	0.5274
Dayton	Butler	0.4219	0.5780
Michigan	**Iowa**	0.5535	0.4464
LSU	Ohio State	0.4898	0.5101
Florida State	Seton Hall	0.4950	0.5049
Gonzaga	Oregon	0.4945	0.5054
Illinois	Indiana	0.4888	0.5111

Fourth Round Matchups
"Elite 8"

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Michigan State	0.3570	0.6429
Dayton	Iowa	0.4805	0.5194
LSU	Florida State	0.3723	0.6277
Gonzaga	Illinois	0.3329	0.6670

Fifth Round Matchups
"Final 4"

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Dayton	0.3606	0.6393
LSU	**Gonzaga**	0.6037	0.3963

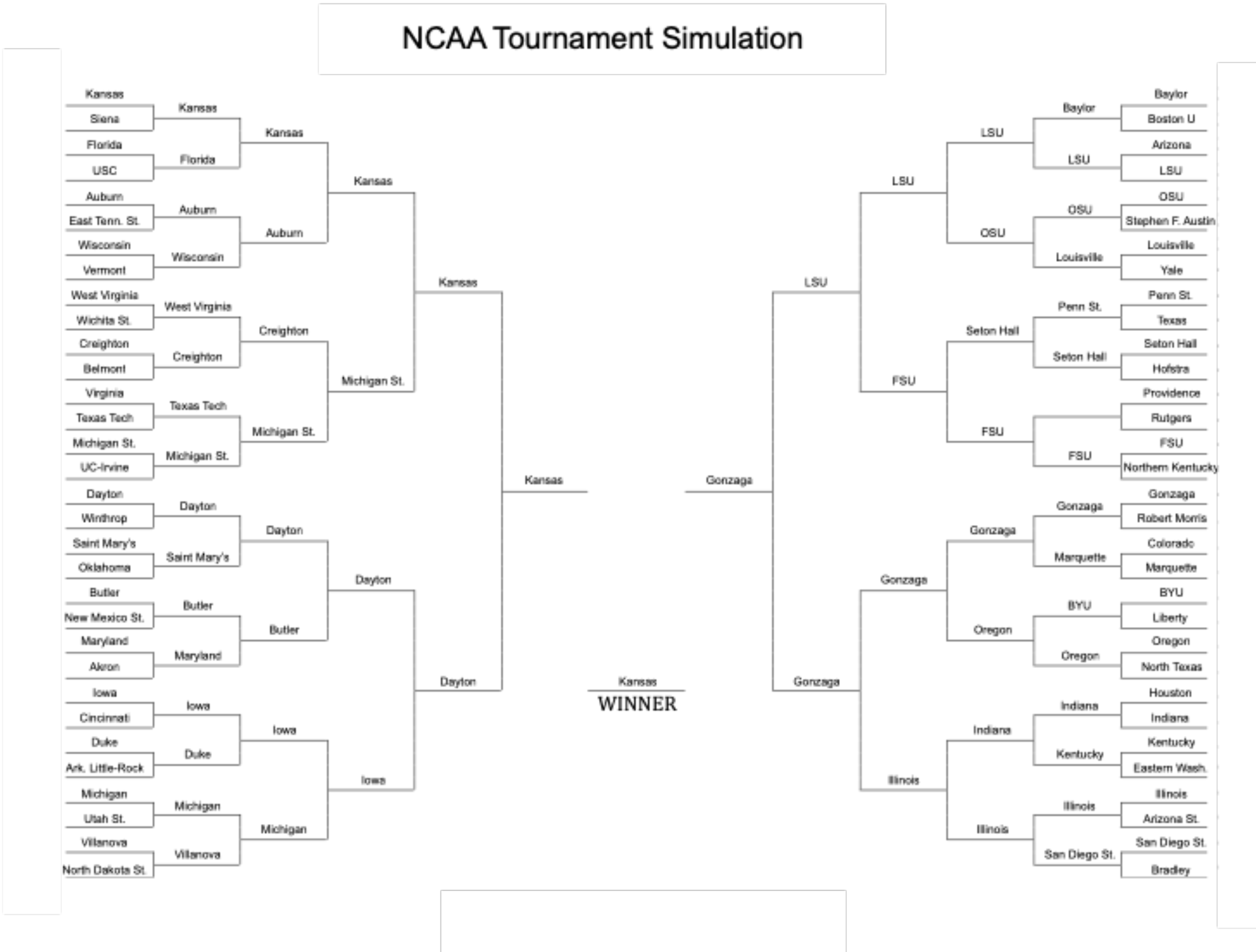
Sixth Round Matchups
"Championship Game"

*** Team in bold is projected to win*

Team	Opponent	Prob. of Team Loss	Prob. of Team Win
Kansas	Gonzaga	0.4376	0.5623

****Kansas is the project 2020 National Champion****

Tournament Bracket



Important Points

Most seasons there is one team that usually stands out as the best team in the nation. This year's college basketball season was considered to be "one of the widest fields" in a very long time, where there were many teams that could win the championship. Despite the distribution of talent throughout the nation many experts considered Kansas to be the best team this year. Our final model predicted that Kansas would win the championship and their most difficult test would come in the championship where the model predicted Kansas had a 56% of beating Gonzaga.

The most interesting revelation of the model was the fact that 9-seed LSU (Louisiana State) was projected to make it to the final four. Every year fans alike root for an underdog and are typically rewarded with a high-seeded team making a run in the tournament. Despite the impressive runs these teams rarely make it to the final four. In fact, even though high seeded teams have made the final four in the past, only Wichita State in 2013 reached that stage as a number 9 seed. The fact our model predicted LSU to make this run depicts that they may have been undervalued going into the tournament and or might have gotten a nice draw. In addition, this LSU team was projected to pull off one of the bigger upsets in the tournament by beating number 1 seed Baylor in the second round.

When the team concluded the bracket, we noticed a few things that made us believe that our model was working. In the first few rounds when the low seeded teams were playing the high seeded teams, the favorites were projected to win at a very high rate. As the tournament progressed, we also noticed that most of the games had a very thin margin in win disparity.

Lastly, the team was very pleased with the results of our model, other models actually predicted other winners. For instance, our GLM model which we agreed was our second-best model actually had number 4 seeded Oregon winning the championship. Oregon projected to beat Kansas, the projected winner in our final model, in the championship. As a team we concluded that Oregon may have been unlucky as it drew a tough path to the championship. In the final model Oregon actually lost in the third round to Gonzaga where Gonzaga had a 50.54% probability of winning that game.

R Code:

```
## Load the libraries need
library(caret)
library(e1071)
library(ggplot2)
library(corrplot)
library(partykit)
library(pROC)
library(tidyverse)
library(ggimage)
library(ggplot2)

### Set a seed
set.seed(69)

### Read in the data and remove the last two rows(Empty)
data <- read.csv("Dataset.csv", header= TRUE)
data <- data[1:10518,]

### Change the 0 and 1 to loss and win and set it as a factor
data$Win <- gsub("0", "Loss", data$Win)
data$Win <- gsub("1", "Win", data$Win)
data$Win <- as.factor(data$Win)

### Create a win margin column
data$WinMargin <- data$teamscore - data$oppscore

## Check the structure of the dataset
str(data)

## Attach the data to easily refer to it later
attach(data)

### Create a correlaion plot to show collinearity between predictors
corr_data <- data[,c(13:34)]
corr_matrix <- cor(corr_data)
corrplot(corr_matrix,type = "upper")

### Create Boxplots for the variables to show distribution
par(mfrow=c(3,3))
boxplot(data$WinMargin, main = "Distribution of Win Margin", xlab = "Win Margin")
boxplot(data$SOS, main = "Distribution of Strength of Schedule", xlab = "Strength of
Schedule")
boxplot(data$FTr, main = "Distribution of Free Throw Rate", xlab = "Free Throw Rate")
boxplot(data$Pace, main = "Distribution of Pace", xlab = "Pace")
boxplot(data$eFG., main = "Distribution of Effective Field Goal %", xlab = "EFG%")
```

```
boxplot(data$ORtg, main = "Distribution of Offensive Rating %", xlab = "ORtg%")
boxplot(data$AST., main = "Distribution of Assist %", xlab = "AST%")
boxplot(data$STL., main = "Distribution of Steal %", xlab = "STL%")
boxplot(data$BLK., main = "Distribution of Block %", xlab = "BLK%")
dev.off()

### Get the skewness of all the variables
skewness(data$WinMargin)
skewness(data$SOS)
skewness(data$FTr)
skewness(data$Pace)
skewness(data$eFG.)
skewness(data$ORtg)
skewness(data$AST.)
skewness(data$STL.)
skewness(data$BLK.)

### Create a QQ plot for all the variables to check for normaility
par(mfrow = c(3,3))
qqnorm(data$WinMargin, main = "Win Margin", col = 1)
qqline(data$WinMargin, main = "Win Margin")
qqnorm(data$SOS, main = "Strength of Schedule", col = 2)
qqline(data$SOS, main = "Strength of Schedule")
qqnorm(data$FTr, main = "Free Throw Rate", col = 3)
qqline(data$FTr, main = "Free Throw Rate")
qqnorm(data$Pace, main = "Pace", col = 4)
qqline(data$Pace, main = "Pace")
qqnorm(data$eFG., main = "Effective Field Goal %", col = 5)
qqline(data$eFG., main = "Effective Field Goal %")
qqnorm(data$ORtg, main = "Offensive Rating", col = 6)
qqline(data$ORtg, main = "Offensive Rating")
qqnorm(data$AST., main = "Assist %", col = 7)
qqline(data$AST., main = "Assist %")
qqnorm(data$STL., main = "Steal %", col = 8)
qqline(data$STL., main = "Steal %")
qqnorm(data$BLK., main = "Block %", col = "orange")
qqline(data$BLK., main = "Block %")
dev.off()

### Create Histograms for the variables to show distribution
par(mfrow = c(3,3))
hist(data$WinMargin, main = "Distribution of Win Margin", col = 1, xlab = "Win Margin")
hist(data$SOS, main = "Distribution of Strength of Schedule", col = 2, xlab = "Strength of
Schedule")
hist(data$FTr, main = "Distribution of Free Throw Rate", col = 3, xlab = "Free Throw Rate")
hist(data$Pace, main = "Distribution of Pace", col = 4, xlab = "Pace")
```

```
hist(data$eFG., main = "Distribution of Effective Field Goal %", col = 5, xlab = "EFG%")
hist(data$ORtg, main = "Distribution of Offensive Rating", col = 6, xlab = "ORtg")
hist(data$AST., main = "Distribution of Assist %", col = 7, xlab = "AST%")
hist(data$STL., main = "Distribution of Steal %", col = 8, xlab = "STL%")
hist(data$BLK., main = "Distribution of Block %", col = 'orange', xlab = "BLK%")
dev.off()

### Create plots that show the relationships in stats for every team
teams <- read.csv("TournamentTeams.csv", header = TRUE)

### THE FOLLOWING LINES HAVE BEEN COMMENTED OUT (Lines 108-178) AS THE
CSV FILE
### "TournamentTeams.csv" INCLUDES THE LOGO OF EVERY TEAM WHICH WERE
### STORED ON A LOCAL COMPUTER. THE RESULTS ARE DISPLAYED IN THE
PAPER AND
### THE FILE IS NOT USED ANYWHERE ELSE




### Pace by Effective field goals
#teams %>% ggplot(aes(x=teams$Pace, y=teams$eFG.)) + geom_image(image = teams$Logo,
asp = 16/9) +
 # geom_vline(xintercept = mean(teams$Pace), linetype = "dashed", color = "red") +
  #geom_hline(yintercept = mean(teams$eFG.), linetype = "dashed", color = "red") +
  #labs(y = "Effective Field Goal %",
   #    x = "Pace",
    #   caption = "Figure: Ricardo Munoz, Nathan Keckley, Matthew Forey Andrew Locker",
     #  title = "Team eFG% by Pace",
      # subtitle = "2020 Season") +
 # theme_bw() +
  #theme(
   # axis.text = element_text(size = 10),
    #axis.title.x = element_text(size = 12),
    #axis.title.y = element_text(size = 12),
    #plot.title = element_text(size = 16),
    #plot.subtitle = element_text(size = 14),
    #plot.caption = element_text(size = 10))

### Offensive rebounding % by effective field goal %
#teams %>% ggplot(aes(x=teams$ORB., y=teams$eFG.)) + geom_image(image = teams$Logo,
asp = 16/9) +
 # geom_vline(xintercept = mean(teams$ORB.), linetype = "dashed", color = "red") +
  #geom_hline(yintercept = mean(teams$eFG.), linetype = "dashed", color = "red") +
  #labs(y = "Effective Field Goal %",
   #    x = "Offensive Rebounding %",
```

```
   #   caption = "Figure: Ricardo Munoz, Nathan Keckley, Matthew Forey Andrew Locker",
    #  title = "Team eFG% and ORB %",
     # subtitle = "2020 Season") +
 # theme_bw() +
  #theme(
   # axis.text = element_text(size = 10),
    #axis.title.x = element_text(size = 12),
    #axis.title.y = element_text(size = 12),
    #plot.title = element_text(size = 16),
    #plot.subtitle = element_text(size = 14),
    #plot.caption = element_text(size = 10))

### turnover by Pace%
#teams %>% ggplot(aes(x=teams$Pace, y=teams$TOV.)) + geom_image(image = teams$Logo,
asp = 16/9) +
 # geom_vline(xintercept = mean(teams$Pace), linetype = "dashed", color = "red") +
  #geom_hline(yintercept = mean(teams$TOV.), linetype = "dashed", color = "red") +
  #labs(y = "Turnover %",
   #    x = "Pace",
    #   caption = "Figure: Ricardo Munoz, Nathan Keckley, Matthew Forey Andrew Locker",
     #  title = "Pace and Turnover %",
      # subtitle = "2020 Season") +
  #theme_bw() +
  #theme(
   # axis.text = element_text(size = 10),
    #axis.title.x = element_text(size = 12),
    #axis.title.y = element_text(size = 12),
    #plot.title = element_text(size = 16),
    #plot.subtitle = element_text(size = 14),
    #plot.caption = element_text(size = 10))


## Change the factors to numeric columns
data$Win <- as.factor(data$Win)
data$G <- as.numeric(data$G)
data$Overall_W <- as.numeric(data$Overall_W)
data$Overall_L <- as.numeric(data$Overall_L)
data$SRS <- as.numeric(data$SRS)
data$SOS <- as.numeric(data$SOS)
data$Home_W <- as.numeric(data$Home_W)
data$Home_L <- as.numeric(data$Home_L)
data$Away_W <- as.numeric(data$Away_W)
data$Away_L <- as.numeric(data$Away_L)
data$Tm.Total.Pts <- as.numeric(data$Tm.Total.Pts)
data$Tm.Pts_Allowed <- as.numeric(data$Tm.Pts_Allowed)
data$Pace <- as.numeric(data$Pace)
```

```
data$ORtg <- as.numeric(data$ORtg)
data$FTr <- as.numeric(data$FTr)
data$X3PAr<- as.numeric(data$X3PAr)
data$TS.<- as.numeric(data$TS.)
data$TRB.<- as.numeric(data$TRB.)
data$AST.<- as.numeric(data$AST.)
data$STL.<- as.numeric(data$STL.)
data$BLK.<- as.numeric(data$BLK.)
data$eFG.<- as.numeric(data$eFG.)
data$TOV.<- as.numeric(data$TOV.)
data$ORB.<- as.numeric(data$ORB.)
data$FT.FGA<- as.numeric(data$FT.FGA)

## Change the factors to numeric columns
data$G_2 <- as.numeric(data$G_2)
data$Overall_W_2 <- as.numeric(data$Overall_W_2)
data$Overall_L_2 <- as.numeric(data$Overall_L_2)
data$SRS_2 <- as.numeric(data$SRS_2)
data$SOS_2 <- as.numeric(data$SOS_2)
data$Home_W_2 <- as.numeric(data$Home_W_2)
data$Home_L_2 <- as.numeric(data$Home_L_2)
data$Away_W_2 <- as.numeric(data$Away_W_2)
data$Away_L_2 <- as.numeric(data$Away_L_2)
data$Tm.Total.Pts_2 <- as.numeric(data$Tm.Total.Pts_2)
data$Tm.Pts_Allowed_2 <- as.numeric(data$Tm.Pts_Allowed_2)
data$Pace_2 <- as.numeric(data$Pace_2)
data$ORtg_2 <- as.numeric(data$ORtg_2)
data$FTr_2 <- as.numeric(data$FTr_2)
data$X3PAr_2 <- as.numeric(data$X3PAr_2)
data$TS._2 <- as.numeric(data$TS._2)
data$TRB_2. <- as.numeric(data$TRB._2)
data$AST._2 <- as.numeric(data$AST._2)
data$STL._2 <- as.numeric(data$STL._2)
data$BLK._2 <- as.numeric(data$BLK._2)
data$eFG._2 <- as.numeric(data$eFG._2)
data$TOV._2 <- as.numeric(data$TOV._2)
data$ORB._2<- as.numeric(data$ORB._2)
data$FT.FGA_2<- as.numeric(data$FT.FGA_2)

### Change the names of all the teams

### NOT REQUIRED TO RUN THIS PORTION (LINES 150 - 436)

# data$team <- gsub("A&M-Corpus Christi", "Texas A&M Corpus Christi", data$team)
# data$team <- gsub("Alabama St.", "Alabama State", data$team)
# data$team <- gsub("Alcorn", "Alcorn State", data$team)
```

```
# data$team <- gsub("Appalachian St.", "Appalachian State", data$team)
# data$team <- gsub("Arizona St.", "Arizona State", data$team)
# data$team <- gsub("Ark-Pine Bluff", "Arkansas-Pine Bluff", data$team)
# data$team <- gsub("Arkansas St.", "Arkansas State", data$team)
# data$team <- gsub("Army West Point", "Army", data$team)
# data$team <- gsub("Ball St.", "Ball State", data$team)
# data$team <- gsub("Boise St.", "TBoise State", data$team)
# data$team <- gsub("Boston U.", "Boston University", data$team)
# data$team <- gsub("Bowling Green", "Bowling Green State", data$team)
# data$team <- gsub("BYU", "Brigham Young", data$team)
# data$team <- gsub("Cal St. Fullerton", "Cal State Fullerton", data$team)
# data$team <- gsub("California", "University of California", data$team)
# data$team <- gsub("Central Ark.", "Central Arkansas", data$team)
# data$team <- gsub("Central Conn. St.", "Central Connecticut State", data$team)
# data$team <- gsub("Central Mich.", "Central Michigan", data$team)
# data$team <- gsub("Charleston So.", "Charleston Southern", data$team)
# data$team <- gsub("Chicago St.", "Chicago State", data$team)
# data$team <- gsub("Cleveland St.", "Cleveland State", data$team)
# data$team <- gsub("Col. of Charleston", "College of Charleston", data$team)
# data$team <- gsub("Colorado St.", "Colorado State", data$team)
# data$team <- gsub("Copponentin St.", "TCopponentin State", data$team)
# data$team <- gsub("CSU Bakersfield", "Cal State Bakersfield", data$team)
# data$team <- gsub("CSUN", "Cal State Northridge", data$team)
# data$team <- gsub("Delaware St.", "Delaware State", data$team)
# data$team <- gsub("Eastern Ill.", "Eastern Illinois", data$team)
# data$team <- gsub("Eastern Ky.", "Eastern Kentucky", data$team)
# data$team <- gsub("Eastern Mich.", "Eastern Michigan", data$team)
# data$team <- gsub("Eastern Wash.", "Eastern Washington", data$team)
# data$team <- gsub("ETSU", "East Tennessee State", data$team)
# data$team <- gsub("FGCU", "Florida Gulf Coast", data$team)
# data$team <- gsub("FIU", "Florida International", data$team)
# data$team <- gsub("Fla. Atlantic", "Florida Atlantic", data$team)
# data$team <- gsub("Florida St.", "Florida State", data$team)
# data$team <- gsub("Fresno St.", "Fresno State", data$team)
# data$team <- gsub("Ga. Southern", "Georgia Southern", data$team)
# data$team <- gsub("Georgia St.", "Georgia State", data$team)
# data$team <- gsub("Idaho St.", "Idaho State", data$team)
# data$team <- gsub("Illinois St.", "Illinois State", data$team)
# data$team <- gsub("Indiana St.", "Indiana State", data$team)
# data$team <- gsub("Iowa St.", "Iowa State", data$team)
# data$team <- gsub("Jackson St.", "Jackson State", data$team)
# data$team <- gsub("Jacksonville St.", "Jacksonville State", data$team)
# data$team <- gsub("Kansas City", "Missouri-Kansas City", data$team)
# data$team <- gsub("Kansas St.", "Kansas State", data$team)
# data$team <- gsub("Kennesaw St.", "Kennesaw State", data$team)
# data$team <- gsub("Kent St.", "Kent State", data$team)
```

```
# data$team <- gsub("La-Monroe", "Louisiana-Monroe", data$team)
# data$team <- gsub("Lamar University", "Lamar", data$team)
# data$team <- gsub("LIU", "Long Island University", data$team)
# data$team <- gsub("LMU (CA)", "Loyola Marymount", data$team)
# data$team <- gsub("Long Beach St.", "Cal State Longbeach", data$team)
# data$team <- gsub("Loyola Chicago", "Loyola (IL)", data$team)
# data$team <- gsub("Loyola Maryland", "Loyola (MD)", data$team)
# data$team <- gsub("LSU", "Louisana State", data$team)
# data$team <- gsub("McNeese", "McNeese State", data$team)
# data$team <- gsub("Michigan St.", "Michigan State", data$team)
# data$team <- gsub("Middle Tenn.", "Middle Tennessee", data$team)
# data$team <- gsub("Mississippi St.", "Mississippi State", data$team)
# data$team <- gsub("Mississippi Val.", "Mississippi Valley State", data$team)
# data$team <- gsub("Missouri St.", "Missouri State", data$team)
# data$team <- gsub("Montana St.", "Montana State", data$team)
# data$team <- gsub("Morehead St.", "Morehead State", data$team)
# data$team <- gsub("Morgan St.", "Morgan State", data$team)
# data$team <- gsub("Murray St.", "Murray State", data$team)
# data$team <- gsub("N.C. A&T", "North Carolina A&T", data$team)
# data$team <- gsub("N.C. Central", "North Carolina Central", data$team)
# data$team <- gsub("NC State", "North Carolina State", data$team)
# data$team <- gsub("New Mexico St.", "New Mexico State", data$team)
# data$team <- gsub("Nicholls St.", "Nicholls State", data$team)
# data$team <- gsub("Norfolk St.", "Norfolk State", data$team)
# data$team <- gsub("North Ala.", "North Alabama", data$team)
# data$team <- gsub("North Dakota St.", "North Dakota State", data$team)
# data$team <- gsub("Northern Ariz.", "Northern Arizona", data$team)
# data$team <- gsub("Northern Col.", "Northern Colorado", data$team)
# data$team <- gsub("Northern Ill.", "Northern Illinois", data$team)
# data$team <- gsub("Northern Ky.", "Norther Kentucky", data$team)
# data$team <- gsub("Northwestern St.", "Northwestern State", data$team)
# data$team <- gsub("Ohio St.", "Ohio State", data$team)
# data$team <- gsub("Oklahoma St.", "Oklahoma State", data$team)
# data$team <- gsub("Ole Miss", "Mississippi", data$team)
# data$team <- gsub("Omaha", "Omaha", data$team)
# data$team <- gsub("Oregon St.", "Oregon State", data$team)
# data$team <- gsub("Penn", "Pennsylvania", data$team)
# data$team <- gsub("Penn St.", "Penn State", data$team)
# data$team <- gsub("Portland St.", "Portland State", data$team)
# data$team <- gsub("Purdue Fort Wayne", "Purdue-Fort Wayne", data$team)
# data$team <- gsub("Sacramento St.", "Sacramento State", data$team)
# data$team <- gsub("Sam Houston St.", "Sam Houston State", data$team)
# data$team <- gsub("San Diego St.", "San Diego State", data$team)
# data$team <- gsub("San Jose St.", "San Jose State", data$team)
# data$team <- gsub("Seattle U", "Seattle University", data$team)
# data$team <- gsub("SFA", "Stephen F. Austin", data$team)
```

```
# data$team <- gsub("SIUE", "SIU Edwardsville", data$team)
# data$team <- gsub("SMU", "Southern Methodist", data$team)
# data$team <- gsub("South Carolina St.", "South Carolina State", data$team)
# data$team <- gsub("South Dakota St.", "South Dakota State", data$team)
# data$team <- gsub("South Fla.", "South Florida", data$team)
# data$team <- gsub("Southeast Mo. St.", "Southeast Missouri State", data$team)
# data$team <- gsub("Southeastern La.", "Southeastern Louisiana", data$team)
# data$team <- gsub("Southern Ill.", "Southern Illinois", data$team)
# data$team <- gsub("Southern Miss.", "Southern Mississippi", data$team)
# data$team <- gsub("Southern U.", "Southern", data$team)
# data$team <- gsub("St. Francis Brooklyn", "St. Francis (NY)", data$team)
# data$team <- gsub("TCU", "Texas Christian", data$team)
# data$team <- gsub("Texas St.", "Texas State", data$team)
# data$team <- gsub("The Citadel", "Citadel", data$team)
# data$team <- gsub("UAB", "Alabama-Birmingham", data$team)
# data$team <- gsub("UC Davis", "UC-Davis", data$team)
# data$team <- gsub("UC Riverside", "UC-Riverside", data$team)
# data$team <- gsub("UC Santa Barbara", "UC-Santa Barbara", data$team)
# data$team <- gsub("UCF", "Central Florida", data$team)
# data$team <- gsub("UConn", "Connecticut", data$team)
# data$team <- gsub("UIC", "Illinois-Chicago", data$team)
# data$team <- gsub("UIW", "Incarnate Word", data$team)
# data$team <- gsub("Umass Lowell", "Massachusetts-Lowell", data$team)
# data$team <- gsub("UMBC", "Maryland-Baltimore County", data$team)
# data$team <- gsub("UMES", "Maryland-Eastern Shore", data$team)
# data$team <- gsub("UNC Asheville", "North Carolina-Asheville", data$team)
# data$team <- gsub("UNC Greensboro", "North Carolina-Greensboro", data$team)
# data$team <- gsub("UNCW", "North Carolina-Wilmington", data$team)
# data$team <- gsub("UNI", "Northern Iowa", data$team)
# data$team <- gsub("UNLV", "Nevada-Las Vegas", data$team)
# data$team <- gsub("USC Upstate", "South Carolina Upstate", data$team)
# data$team <- gsub("UT Arlington", "Texas-Arlington", data$team)
# data$team <- gsub("UT Martin", "Tennessee-Martin", data$team)
# data$team <- gsub("Utah St.", "Utah State", data$team)
# data$team <- gsub("UTEP", "Texas-El Paso", data$team)
# data$team <- gsub("UTRGV", "Texas- Rio Grande Valley", data$team)
# data$team <- gsub("UTSA", "Texas-San Antonio", data$team)
# data$team <- gsub("VCU", "Virginia Commonwealth", data$team)
# data$team <- gsub("Washington St.", "Washington State", data$team)
# data$team <- gsub("Weber St.", "Weber State", data$team)
# data$team <- gsub("Western Caro.", "Western Carolina", data$team)
# data$team <- gsub("Western Ill.", "Western Illinois", data$team)
# data$team <- gsub("Western Ky.", "Western Kentucky", data$team)
# data$team <- gsub("Western Mich.", "Western Michigan", data$team)
# data$team <- gsub("Wichita St.", "Wichita State", data$team)
# data$team <- gsub("Wright St.", "Wright State", data$team)
```

```
# data$team <- gsub("Youngstown St.", "Youngstown State", data$team)
#
#
# ### Change the name of opponentonenentent
# data$opponent <- gsub("A&M-Corpus Christi", "Texas A&M Corpus Christi",
data$opponent)
# data$opponent <- gsub("Alabama St.", "Alabama State", data$opponent)
# data$opponent <- gsub("Alcorn", "Alcorn State", data$opponent)
# data$opponent <- gsub("Appalachian St.", "Appalachian State", data$opponent)
# data$opponent <- gsub("Arizona St.", "Arizona State", data$opponent)
# data$opponent <- gsub("Ark-Pine Bluff", "Arkansas-Pine Bluff", data$opponent)
# data$opponent <- gsub("Arkansas St.", "Arkansas State", data$opponent)
# data$opponent <- gsub("Army West Point", "Army", data$opponent)
# data$opponent <- gsub("Ball St.", "Ball State", data$opponent)
# data$opponent <- gsub("Boise St.", "TBoise State", data$opponent)
# data$opponent <- gsub("Boston U.", "Boston University", data$opponent)
# data$opponent <- gsub("Bowling Green", "Bowling Green State", data$opponent)
# data$opponent <- gsub("BYU", "Brigham Young", data$opponent)
# data$opponent <- gsub("Cal St. Fullerton", "Cal State Fullerton", data$opponent)
# data$opponent <- gsub("California", "University of California", data$opponent)
# data$opponent <- gsub("Central Ark.", "Central Arkansas", data$opponent)
# data$opponent <- gsub("Central Conn. St.", "Central Connecticut State", data$opponent)
# data$opponent <- gsub("Central Mich.", "Central Michigan", data$opponent)
# data$opponent <- gsub("Charleston So.", "Charleston Southern", data$opponent)
# data$opponent <- gsub("Chicago St.", "Chicago State", data$opponent)
# data$opponent <- gsub("Cleveland St.", "Cleveland State", data$opponent)
# data$opponent <- gsub("Col. of Charleston", "College of Charleston", data$opponent)
# data$opponent <- gsub("Colorado St.", "Colorado State", data$opponent)
# data$opponent <- gsub("Copponentin St.", "TCopponentin State", data$opponent)
# data$opponent <- gsub("CSU Bakersfield", "Cal State Bakersfield", data$opponent)
# data$opponent <- gsub("CSUN", "Cal State Northridge", data$opponent)
# data$opponent <- gsub("Delaware St.", "Delaware State", data$opponent)
# data$opponent <- gsub("Eastern Ill.", "Eastern Illinois", data$opponent)
# data$opponent <- gsub("Eastern Ky.", "Eastern Kentucky", data$opponent)
# data$opponent <- gsub("Eastern Mich.", "Eastern Michigan", data$opponent)
# data$opponent <- gsub("Eastern Wash.", "Eastern Washington", data$opponent)
# data$opponent <- gsub("ETSU", "East Tennessee State", data$opponent)
# data$opponent <- gsub("FGCU", "Florida Gulf Coast", data$opponent)
# data$opponent <- gsub("FIU", "Florida International", data$opponent)
# data$opponent <- gsub("Fla. Atlantic", "Florida Atlantic", data$opponent)
# data$opponent <- gsub("Florida St.", "Florida State", data$opponent)
# data$opponent <- gsub("Fresno St.", "Fresno State", data$opponent)
# data$opponent <- gsub("Ga. Southern", "Georgia Southern", data$opponent)
# data$opponent <- gsub("Georgia St.", "Georgia State", data$opponent)
# data$opponent <- gsub("Idaho St.", "Idaho State", data$opponent)
# data$opponent <- gsub("Illinois St.", "Illinois State", data$opponent)
```

```
# data$opponent <- gsub("Indiana St.", "Indiana State", data$opponent)
# data$opponent <- gsub("Iowa St.", "Iowa State", data$opponent)
# data$opponent <- gsub("Jackson St.", "Jackson State", data$opponent)
# data$opponent <- gsub("Jacksonville St.", "Jacksonville State", data$opponent)
# data$opponent <- gsub("Kansas City", "Missouri-Kansas City", data$opponent)
# data$opponent <- gsub("Kansas St.", "Kansas State", data$opponent)
# data$opponent <- gsub("Kennesaw St.", "Kennesaw State", data$opponent)
# data$opponent <- gsub("Kent St.", "Kent State", data$opponent)
# data$opponent <- gsub("La-Monroe", "Louisiana-Monroe", data$opponent)
# data$opponent <- gsub("Lamar University", "Lamar", data$opponent)
# data$opponent <- gsub("LIU", "Long Island University", data$opponent)
# data$opponent <- gsub("LMU (CA)", "Loyola Marymount", data$opponent)
# data$opponent <- gsub("Long Beach St.", "Cal State Longbeach", data$opponent)
# data$opponent <- gsub("Loyola Chicago", "Loyola (IL)", data$opponent)
# data$opponent <- gsub("Loyola Maryland", "Loyola (MD)", data$opponent)
# data$opponent <- gsub("LSU", "Louisana State", data$opponent)
# data$opponent <- gsub("McNeese", "McNeese State", data$opponent)
# data$opponent <- gsub("Michigan St.", "Michigan State", data$opponent)
# data$opponent <- gsub("Middle Tenn.", "Middle Tennessee", data$opponent)
# data$opponent <- gsub("Mississippi St.", "Mississippi State", data$opponent)
# data$opponent <- gsub("Mississippi Val.", "Mississippi Valley State", data$opponent)
# data$opponent <- gsub("Missouri St.", "Missouri State", data$opponent)
# data$opponent <- gsub("Montana St.", "Montana State", data$opponent)
# data$opponent <- gsub("Morehead St.", "Morehead State", data$opponent)
# data$opponent <- gsub("Morgan St.", "Morgan State", data$opponent)
# data$opponent <- gsub("Murray St.", "Murray State", data$opponent)
# data$opponent <- gsub("N.C. A&T", "North Carolina A&T", data$opponent)
# data$opponent <- gsub("N.C. Central", "North Carolina Central", data$opponent)
# data$opponent <- gsub("NC State", "North Carolina State", data$opponent)
# data$opponent <- gsub("New Mexico St.", "New Mexico State", data$opponent)
# data$opponent <- gsub("Nicholls St.", "Nicholls State", data$opponent)
# data$opponent <- gsub("Norfolk St.", "Norfolk State", data$opponent)
# data$opponent <- gsub("North Ala.", "North Alabama", data$opponent)
# data$opponent <- gsub("North Dakota St.", "North Dakota State", data$opponent)
# data$opponent <- gsub("Northern Ariz.", "Northern Arizona", data$opponent)
# data$opponent <- gsub("Northern Col.", "Northern Colorado", data$opponent)
# data$opponent <- gsub("Northern Ill.", "Northern Illinois", data$opponent)
# data$opponent <- gsub("Northern Ky.", "Norther Kentucky", data$opponent)
# data$opponent <- gsub("Northwestern St.", "Northwestern State", data$opponent)
# data$opponent <- gsub("Ohio St.", "Ohio State", data$opponent)
# data$opponent <- gsub("Oklahoma St.", "Oklahoma State", data$opponent)
# data$opponent <- gsub("Ole Miss", "Mississippi", data$opponent)
# data$opponent <- gsub("Omaha", "Omaha", data$opponent)
# data$opponent <- gsub("Oregon St.", "Oregon State", data$opponent)
# data$opponent <- gsub("Penn", "Pennsylvania", data$opponent)
# data$opponent <- gsub("Penn St.", "Penn State", data$opponent)
```

```
# data$opponent <- gsub("Portland St.", "Portland State", data$opponent)
# data$opponent <- gsub("Purdue Fort Wayne", "Purdue-Fort Wayne", data$opponent)
# data$opponent <- gsub("Sacramento St.", "Sacramento State", data$opponent)
# data$opponent <- gsub("Sam Houston St.", "Sam Houston State", data$opponent)
# data$opponent <- gsub("San Diego St.", "San Diego State", data$opponent)
# data$opponent <- gsub("San Jose St.", "San Jose State", data$opponent)
# data$opponent <- gsub("Seattle U", "Seattle University", data$opponent)
# data$opponent <- gsub("SFA", "Stephen F. Austin", data$opponent)
# data$opponent <- gsub("SIUE", "SIU Edwardsville", data$opponent)
# data$opponent <- gsub("SMU", "Southern Methodist", data$opponent)
# data$opponent <- gsub("South Carolina St.", "South Carolina State", data$opponent)
# data$opponent <- gsub("South Dakota St.", "South Dakota State", data$opponent)
# data$opponent <- gsub("South Fla.", "South Florida", data$opponent)
# data$opponent <- gsub("Southeast Mo. St.", "Southeast Missouri State", data$opponent)
# data$opponent <- gsub("Southeastern La.", "Southeastern Louisiana", data$opponent)
# data$opponent <- gsub("Southern Ill.", "Southern Illinois", data$opponent)
# data$opponent <- gsub("Southern Miss.", "Southern Mississippi", data$opponent)
# data$opponent <- gsub("Southern U.", "Southern", data$opponent)
# data$opponent <- gsub("St. Francis Brooklyn", "St. Francis (NY)", data$opponent)
# data$opponent <- gsub("TCU", "Texas Christian", data$opponent)
# data$opponent <- gsub("Texas St.", "Texas State", data$opponent)
# data$opponent <- gsub("The Citadel", "Citadel", data$opponent)
# data$opponent <- gsub("UAB", "Alabama-Birmingham", data$opponent)
# data$opponent <- gsub("UC Davis", "UC-Davis", data$opponent)
# data$opponent <- gsub("UC Riverside", "UC-Riverside", data$opponent)
# data$opponent <- gsub("UC Santa Barbara", "UC-Santa Barbara", data$opponent)
# data$opponent <- gsub("UCF", "Central Florida", data$opponent)
# data$opponent <- gsub("UConn", "Connecticut", data$opponent)
# data$opponent <- gsub("UIC", "Illinois-Chicago", data$opponent)
# data$opponent <- gsub("UIW", "Incarnate Word", data$opponent)
# data$opponent <- gsub("Umass Lowell", "Massachusetts-Lowell", data$opponent)
# data$opponent <- gsub("UMBC", "Maryland-Baltimore County", data$opponent)
# data$opponent <- gsub("UMES", "Maryland-Eastern Shore", data$opponent)
# data$opponent <- gsub("UNC Asheville", "North Carolina-Asheville", data$opponent)
# data$opponent <- gsub("UNC Greensboro", "North Carolina-Greensboro", data$opponent)
# data$opponent <- gsub("UNCW", "North Carolina-Wilmington", data$opponent)
# data$opponent <- gsub("UNI", "Northern Iowa", data$opponent)
# data$opponent <- gsub("UNLV", "Nevada-Las Vegas", data$opponent)
# data$opponent <- gsub("USC Upstate", "South Carolina Upstate", data$opponent)
# data$opponent <- gsub("UT Arlington", "Texas-Arlington", data$opponent)
# data$opponent <- gsub("UT Martin", "Tennessee-Martin", data$opponent)
# data$opponent <- gsub("Utah St.", "Utah State", data$opponent)
# data$opponent <- gsub("UTEP", "Texas-El Paso", data$opponent)
# data$opponent <- gsub("UTRGV", "Texas- Rio Grande Valley", data$opponent)
# data$opponent <- gsub("UTSA", "Texas-San Antonio", data$opponent)
# data$opponent <- gsub("VCU", "Virginia Commonwealth", data$opponent)
```

```
# data$opponent <- gsub("Washington St.", "Washington State", data$opponent)
# data$opponent <- gsub("Weber St.", "Weber State", data$opponent)
# data$opponent <- gsub("Western Caro.", "Western Carolina", data$opponent)
# data$opponent <- gsub("Western Ill.", "Western Illinois", data$opponent)
# data$opponent <- gsub("Western Ky.", "Western Kentucky", data$opponent)
# data$opponent <- gsub("Western Mich.", "Western Michigan", data$opponent)
# data$opponent <- gsub("Wichita St.", "Wichita State", data$opponent)
# data$opponent <- gsub("Wright St.", "Wright State", data$opponent)
# data$opponent <- gsub("Youngstown St.", "Youngstown State", data$opponent)

### Create new variables to capture the difference in stats between both teams
attach(data)
data$FTr_Mar <- (FTr-FTr_2)
data$Pace_Mar <- (Pace-Pace_2)
data$TOV_Mar <- (TOV.-TOV._2)
data$eFG_Mar <- (eFG.-eFG._2)
data$ORB_Mar <- (ORB.-ORB._2)
data$SOS_Mar <- (SOS-SOS_2)
data$AST_Mar <- (AST.-AST._2)
data$STL_Mar <- (STL.-STL._2)

## Histogram that shows the distribution of Strength of Schedule
hist(data$SOS_Mar, main = "Distribution of SOS Margin", col = 5, xlab = "Strength of
Schedule Margin")

### Create a training and testing set using feburary and march games as testing sets
train <- which(month != c(2,3))
dataTrain <- data[train,]
dataTest  <- data[-train,]

### Control function for the models
ctrl <- trainControl(method = "LGOCV",
                 summaryFunction = twoClassSummary,
                 classProbs = TRUE,
                 savePredictions = TRUE)

### GLM Model
### ROC = 0.7875898
### Train Accuracy = 0.7003
### Testing Accuracy = 0.6661

set.seed(69)
glm <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
          data = dataTrain,
          method = "glm",
          metric = "ROC",
```

```
            trControl = ctrl)
summary(glm)
glm

### Training Set
confusionMatrix(data = predict(glm, dataTrain), reference = dataTrain$Win)

### Testing Set
confusionMatrix(data = predict(glm, dataTest), reference = as.factor(dataTest$Win))

### Get the ROC and AUC for the model and plot
dataTest$glm <- predict(glm, dataTest, type = "prob")[,1]
glmROC <- roc(dataTest$Win, dataTest$glm)
auc(glmROC)
plot(glmROC, main = "GLM Model")
legend("bottomright", c("AUC = 0.7341 ","ROC = 0.7875"))

### LDA Model
### ROC = 0.7872555
### Training Accuracy = 0.7074
### Testing Accuracy = 0.6672

set.seed(69)
lda <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
            data = dataTrain,
            method = "lda",
            metric = "ROC",
            trControl = ctrl)
lda

## Training Set
confusionMatrix(data = predict(lda, dataTrain), reference = as.factor(dataTrain$Win))

## Testing Set
confusionMatrix(data = predict(lda, dataTest), reference = as.factor(dataTest$Win))

### Get the ROC and AUC for the model and plot
dataTest$lda <- predict(lda, dataTest, type = "prob")[,1]
ldaROC <- roc(dataTest$Win, dataTest$lda)
auc(ldaROC)
plot(ldaROC, main = "LDA Model")
legend("bottomright", c("AUC = 0.7344 ","ROC = 0.7872"))

### This is our best model so we all the games will be predicted based on the lDA model
### Read in the first round matchups
firstRound <- read.csv("FirstRound.csv", header = T)
```

```
attach(firstRound)

### Create variables that show the difference in useful variables
firstRound$FTr_Mar <- (FTr-FTr_2)
firstRound$Pace_Mar <- (Pace-Pace_2)
firstRound$TOV_Mar <- (TOV.-TOV._2)
firstRound$eFG_Mar <- (eFG.-eFG._2)
firstRound$ORB_Mar <- (ORB.-ORB._2)
firstRound$SOS_Mar <- (SOS-SOS_2)
firstRound$AST_Mar <- (AST.-AST._2)
firstRound$STL_Mar <- (STL.-STL._2)

### Make the predictions for the first round games
ldaPred <- predict(lda, firstRound, type = 'prob')
### Print the probabilites of the team to win that particular game
ldaPred

### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
firstRoundTeams <- firstRound[,5:6]
firstRoundTeams
firstRoundResults <- cbind(firstRoundTeams, ldaPred)

### These are the results for the first round, the probabilites
### in column 3 and 4 are the probabilites for the first team
firstRoundResults

### Read the second round data
secondRound <- read.csv("SecondRound.csv", header = T)

### Create variables that show the difference in useful variables
attach(secondRound)
secondRound$FTr_Mar <- (FTr-FTr_2)
secondRound$Pace_Mar <- (Pace-Pace_2)
secondRound$TOV_Mar <- (TOV.-TOV._2)
secondRound$eFG_Mar <- (eFG.-eFG._2)
secondRound$ORB_Mar <- (ORB.-ORB._2)
secondRound$SOS_Mar <- (SOS-SOS_2)
secondRound$AST_Mar <- (AST.-AST._2)
secondRound$STL_Mar <- (STL.-STL._2)

### Make the predictions for the second round games
ldaPredSecond <- predict(lda, secondRound, type = 'prob')
### Print the probabilites of the team to win that particular game
ldaPredSecond
```

```
### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
secondRoundTeams <- secondRound[,5:6]
secondRoundResults <- cbind(secondRoundTeams, ldaPredSecond)

#### These are the results for the second round, the probabilites
### in column 3 and 4 are the probabilites for the first team
secondRoundResults

### Read the third round data
thirdRound <- read.csv("Sweet16.csv", header = T)

### Create variables that show the difference in useful variables
attach(thirdRound)
thirdRound$FTr_Mar <- (FTr-FTr_2)
thirdRound$Pace_Mar <- (Pace-Pace_2)
thirdRound$TOV_Mar <- (TOV.-TOV._2)
thirdRound$eFG_Mar <- (eFG.-eFG._2)
thirdRound$ORB_Mar <- (ORB.-ORB._2)
thirdRound$SOS_Mar <- (SOS-SOS_2)
thirdRound$AST_Mar <- (AST.-AST._2)
thirdRound$STL_Mar <- (STL.-STL._2)

### Make the predictions for the third round games
ldaPredThirdRound <- predict(lda, thirdRound, type = 'prob')
ldaPredThirdRound

### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
thirdRoundTeams <- thirdRound[,5:6]
thirdRoundResults <- cbind(thirdRoundTeams, ldaPredThirdRound)

#### These are the results for the third round, the probabilites
### in column 3 and 4 are the probabilites for the first team to win
thirdRoundResults

### Read in the fourth round games
fourthRound <- read.csv("Elite8.csv", header = T)


### Create variables that show the difference in useful variables
attach(fourthRound)
fourthRound$FTr_Mar <- (FTr-FTr_2)
fourthRound$Pace_Mar <- (Pace-Pace_2)
fourthRound$TOV_Mar <- (TOV.-TOV._2)
fourthRound$eFG_Mar <- (eFG.-eFG._2)
```

```
fourthRound$ORB_Mar <- (ORB.-ORB._2)
fourthRound$SOS_Mar <- (SOS-SOS_2)
fourthRound$AST_Mar <- (AST.-AST._2)
fourthRound$STL_Mar <- (STL.-STL._2)

### Make the predictions for the fourth round games
ldaPredFourthRound <- predict(lda, fourthRound, type = 'prob')
ldaPredFourthRound

### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
fourthRoundTeams <- fourthRound[,5:6]
fourthRoundResults <- cbind(fourthRoundTeams, ldaPredFourthRound)

#### These are the results for the fourth round, the probabilites
### in column 3 and 4 are the probabilites for the first team
fourthRoundResults

### Read in the fifth round games
fifthRound <- read.csv("FinalFour.csv", header = T)

### Create variables that show the difference in useful variables
attach(fifthRound)
fifthRound$FTr_Mar <- (FTr-FTr_2)
fifthRound$Pace_Mar <- (Pace-Pace_2)
fifthRound$TOV_Mar <- (TOV.-TOV._2)
fifthRound$eFG_Mar <- (eFG.-eFG._2)
fifthRound$ORB_Mar <- (ORB.-ORB._2)
fifthRound$SOS_Mar <- (SOS-SOS_2)
fifthRound$AST_Mar <- (AST.-AST._2)
fifthRound$STL_Mar <- (STL.-STL._2)

### Make the predictions for the fifth round games
ldaPredFifthRound <- predict(lda, fifthRound, type = 'prob')
ldaPredFifthRound

### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
fifthRoundTeams <- fifthRound[,5:6]
fifthRoundResults <- cbind(fifthRoundTeams, ldaPredFifthRound)

#### These are the results for the fifth round, the probabilites
### in column 3 and 4 are the probabilites for the first team
fifthRoundResults

### Read in the championship data
```

```
Championship <- read.csv("Championship.csv", header = T)

### Create variables that show the difference in useful variables
attach(Championship)
Championship$FTr_Mar <- (FTr-FTr_2)
Championship$Pace_Mar <- (Pace-Pace_2)
Championship$TOV_Mar <- (TOV.-TOV._2)
Championship$eFG_Mar <- (eFG.-eFG._2)
Championship$ORB_Mar <- (ORB.-ORB._2)
Championship$SOS_Mar <- (SOS-SOS_2)
Championship$AST_Mar <- (AST.-AST._2)
Championship$STL_Mar <- (STL.-STL._2)

### Make the predictions for the fifth round games
ldaPredChampionship <- predict(lda, Championship, type = 'prob')
ldaPredChampionship

### Create a dataset that simply has the two teams that played in this round and the
probability of the first team winning
championshipTeams <- Championship[,5:6]
championshipResults <- cbind(championshipTeams, ldaPredChampionship)

#### These are the results for the championship round, the probabilites
### in column 3 and 4 are the probabilites for the first team
championshipResults

"The winner of the 2020 Championship is predicted to be Kansas"

### QDA Model
### ROC = 0.7855829
### Training Accuracy = 0.7064
### Testing Accurarcy = 0.6678

set.seed(69)
qda <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
          data = dataTrain,
          method = "qda",
          metric = "ROC",
          trControl = ctrl)
qda

## Training Set
confusionMatrix(data = predict(qda, dataTrain), reference = as.factor(dataTrain$Win))

## Testing Set
confusionMatrix(data = predict(qda, dataTest), reference = as.factor(dataTest$Win))
```

```
### Get the ROC and AUC for the model and plot
dataTest$qda <- predict(qda, dataTest, type = "prob")[,1]
qdaROC <- roc(dataTest$Win, dataTest$qda)
auc(qdaROC)
plot(qdaROC, main = "QDA Model")
legend("bottomright", c("AUC = 0.7335 ","ROC = 0.7855"))

### KNN Model
### Training ROC = 0.7181127
### Training Accuracy = 0.6979
### Testing Accurarcy = 0.6007

set.seed(69)
knn <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
        data = dataTrain,
        method = "knn",
        metric = "ROC",
        tuneGrid = data.frame(k = 1:20),
        trControl = ctrl)
knn

## Training Set
confusionMatrix(data = predict(knn, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(knn, dataTest), reference = dataTest$Win)

### Get the ROC and AUC for the model and plot
dataTest$knn <- predict(knn, dataTest, type = "prob")[,1]
knnROC <- roc(dataTest$Win, dataTest$knn)
auc(knnROC)
plot(knnROC, main = "KNN Model")
legend("bottomright", c("AUC = 0.634 ","ROC = 0.7181"))


### MDA Model
### Training ROC = 0.7865
### Training Accuracy = 0.7054
### Testing Accurarcy = 0.6615

set.seed(69)
mda <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
        data = dataTrain,
        method = "mda",
        metric = "ROC",
```

```
          trControl = ctrl)
mda

## Training Set
confusionMatrix(data = predict(mda, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(mda, dataTest), reference = dataTest$Win)

### Get the ROC and AUC for the model and plot
dataTest$mda <- predict(mda, dataTest, type = "prob")[,1]
mdaROC <- roc(dataTest$Win, dataTest$mda)
auc(mdaROC)
plot(mdaROC, main = "MDA Model")
legend("bottomright", c("AUC = 0.7332 ","ROC = 0.7865"))

### FDA Model
### Training ROC = 0.7882667
### Training Accuracy = 0.7128
### Testing Accurarcy = 0.6746

set.seed(69)
fda <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
          data = dataTrain,
          method = "fda",
          metric = "ROC",
          trControl = ctrl)
fda

## Training Set
confusionMatrix(data = predict(fda, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(fda, dataTest), reference = dataTest$Win)

### Get the ROC and AUC for the model and plot
dataTest$fda <- predict(fda, dataTest, type = "prob")[,1]
fdaROC <- roc(dataTest$Win, dataTest$fda)
auc(fdaROC)
plot(fdaROC, main = "FDA Model")
legend("bottomright", c("AUC = 0.7340 ","ROC = 0.7882"))


### Neural Net Model
### Training ROC = 0.78638
### Training Accuracy = 0.7139
```

```
### Testing Accurarcy = 0.6718

##Neural Net takes apprx. 45 minutes to run

nnetGrid <- expand.grid(.size = 1:10,
                     .decay = c(0, .01, .1, 0.5))
maxSize <- max(nnetGrid$.size)

nnetFit <- train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
              data = dataTrain,
              method = "nnet",
              metric = "ROC",
              preProc = c("center", "scale", "spatialSign"),
              tuneGrid = nnetGrid,
              trace = FALSE,
              maxit = 1000,
              MaxNWts = 200,
              trControl = ctrl)

nnetFit

## Training Set
confusionMatrix(data = predict(nnetFit, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(nnetFit, dataTest), reference = dataTest$Win)


### Get the ROC and AUC for the model and plot
dataTest$nnetFit <- predict(nnetFit, dataTest, type = "prob")[,1]
nnetFitROC <- roc(dataTest$Win, dataTest$nnetFit)
auc(nnetFitROC)
plot(nnetFitROC, main = " Neural Net Model")
legend("bottomright", c("AUC = 0.7332 ","ROC = 0.7863"))



### Naive Bayes Model
### Training ROC =
### Training Accuracy = 0.7061
### Testing Accurarcy = 0.6746

nBayesFit = train(Win~ SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
                data = dataTrain,
                useKernel = TRUE,
                method = "nb")
```

```
nBayesFit

## Training Set
confusionMatrix(data = predict(nBayesFit, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(nBayesFit, dataTest), reference = dataTest$Win)

### Get the ROC and AUC for the model and plot
dataTest$nBayesFit <- predict(nBayesFit, dataTest, type = "prob")[,1]
nBayesFitROC <- roc(dataTest$Win, dataTest$nBayesFit)
auc(nBayesFitROC)
plot(nBayesFitROC, main = "Naive Bayes Model")
legend("bottomright", c("AUC = 0.7285 "))

### Random Forest Model
### Training ROC = 0.7497
### Training Accuracy = 0.9325
### Testing Accurarcy =  0.6138

mtryGrid <- data.frame(mtry = floor(seq(1,
                              ncol(dataTrain),
                              length = 10)))

set.seed(69)
rf <- train(Win ~
SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar+AST_Mar+STL_Mar,
         data=dataTrain,
         method = "rf",
         tuneGrid = mtryGrid,
         ntree = 200,
         importance = TRUE,
         trControl = ctrl)
rf

## Training Set
confusionMatrix(data = predict(rf, dataTrain), reference = dataTrain$Win)
## Testing Set
confusionMatrix(data = predict(rf, dataTest), reference = dataTest$Win)

### Get the ROC and AUC for the model and plot
dataTest$rf <- predict(rf, dataTest, type = "prob")[,1]
rfROC <- roc(dataTest$Win, dataTest$rf)
auc(rfROC)
plot(rfROC, main = "Random Forest Model")
```

```
legend("bottomright", c("AUC = 0.6739 ","ROC = 0.7497"))


### glmnet Model
### Training ROC = 0.7876321
### Training Accuracy = 0.7064
### Testing Accurarcy = 0.6684

glmnGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),
                    .lambda = seq(.01, .2, length = 40))
set.seed(69)
glmnTuned <- train(Win~
SOS_Mar+FTr_Mar+Pace_Mar+eFG_Mar+TOV_Mar+ORB_Mar,
               data = dataTrain,
               method = "glmnet",
               tuneGrid = glmnGrid,
               metric = "ROC",
               trControl = ctrl)
glmnTuned

## Training Set
confusionMatrix(data = predict(glmnTuned, dataTrain), reference = dataTrain$Win)

## Testing Set
confusionMatrix(data = predict(glmnTuned, dataTest), reference = dataTest$Win)


### Get the ROC and AUC for the model and plot
dataTest$glmnTuned <- predict(glmnTuned, dataTest, type = "prob")[,1]
glmnTunedROC <- roc(dataTest$Win, dataTest$glmnTuned)
auc(glmnTunedROC)
plot(glmnTunedROC, main = "GLM Net Model")
legend("bottomright", c("AUC = 0.7346 ","ROC = 0.7876"))

### Create a dot plot showing the results from all the models
res = resamples(list(MDA = mda,KNN = knn, LDA = lda, QDA = qda, GLM = glm, FDA = fda,
GLMNET = glmnTuned, RandomForest = rf, NeuralNet = nnetFit ))
dotplot(res, metric="ROC", main = "Model Comparision")



###ROC curves
plot(glmROC, col=1, lty=1, main = 'Comparison of Models')
lines(ldaROC, col=2, lty=2)
lines(qdaROC, col=3, lty=3)
lines(knnROC, col=10, lty=4)
```

```
lines(mdaROC, col=5, lty=5)
lines(nBayesROC, col=6, lty=6)
lines(nnetFitROC, col=7, lty=7)
lines(rfROC, col=8, lty=8)
lines(fdaROC, col = 'orange', lty=9)
lines(glmnTunedROC, col = 4, lty=10)
legend('bottomright',
c('GLM','LDA','QDA','KNN','MDA','NBayes','NN','RF','FDA','GlmNet'), col=1:10,
lty=1:10,lwd=2)
```