Using Statistics to Create the Perfect March Madness Bracket

Sarah Downs

1 Introduction

The goal of this project is to analyze data from NCAA Division One Men's basketball teams during the regular season to predict how they will perform during the National Championships, colloquially known as March Madness. I use a data set that ranks teams according to their Pomeroy College Basketball Ratings[1]. These ratings give in depth basketball statistics for each year from 2002 until present and use several different measures to help quantify how good or bad a team is. My analysis will take three parts: single linear analysis, multiple linear analysis, and polynomial regression. I start by attempting to do a single linear analysis on the data from the year 2016, first using Adjusted Offensive Efficiency as the predictor and then using Adjusted Defensive Efficiency as the predictor. Next, I attempt a multiple linear analysis and find that by using both the Adjusted Offensive Efficiency and Adjusted Defensive Efficiency, the predictions greatly improve, but still are not perfect. Finally, I attempt polynomial regression using Adjusted Offensive Efficiency as the predictor. After running each of these methods, I found that none of these can predict the perfect bracket, however the multiple linear

1 https://kenpom.com/

regression is able to perform surprisingly well, making the correct final ranking predictions approximately 62.33% of the time.

2 Data

For the purposes of my study, I will be analyzing data from the years 2016, 2017, and 2018. In each of these years, there were 351 teams that played and are thus being analyzed. While there more years of data available, I chose to model only these three years for the sake of simplicity and the fact that the data set was already large enough that I can assume that adding in another year of data will not greatly change the outcomes. Additionally, as the years go by, the average player continually improves, so if there is a change in the important factors in the data it might be due to this and not actually representative of what the data and the teams look and play like today. There are twenty predictors in the data set. Though my methods to not use all twenty predictors, are still important to understand and are thus defined in the table below.

Variable	**Variable Name**	**What it is**
Rank	Rank	This is the teams ranking for each season. For all previous seasons (2002-2018) this matches with who won the National Championship, so 1 was the winner of the Championship,

		2 was the second place, etc. For 2019 the National Championship has not occurred yet so this is largely based on their Win-Loss Record as well as other factors that will be later discussed.
Team	Team	This is the team's name.
Conf	Conference	This is the conference that the team plays in. There are 32 conferences in total that play, each with between 9 and 38 teams. To advance to the National Championships, a team must either win their conference or be granted an invitation to play known as an "at-large berth"[2] which are granted based on how the teams performed throughout the regular season.
AdjEM	Adjusted Efficiency Margin	This is the difference between a team's offensive and defensive efficiency. "It represents the number of points the team would be expected to outscore the average D-I team over 100 possessions and it

2 https://kenpom.com/blog/ratings-glossary/

		has the advantage of being a linear measure."[3] This is essentially the difference between how good a team's defense and offense is giving a full picture of how the team is overall.
AdjO	Adjusted Offensive Efficiency	The "estimate of the offensive efficiency (points scored per 100 possessions) a team would have against the average D-I defense."[4] This helps to measure how good a team's offense is.
AdjO_Rank	Adjusted Offensive Efficiency Ranking	This gives the ranking of teams with the highest to lowest adjusted offensive efficiency. The team ranked 1 will then be considered to have the best offense, with their offense being considered worse as the ranking drops.
AdjD	Adjusted Defensive Efficiency	The "estimate of the defensive efficiency (points allowed per 100 possessions) a team would have against the average D-I offense."[5]

3 https://kenpom.com/blog/ratings-glossary/
4 https://kenpom.com/blog/ratings-glossary/
5 https://kenpom.com/blog/ratings-glossary/

		This helps to measure how good a team's defense is.
AdjD_rank	Adjusted Defensive Efficiency Ranking	This gives the ranking of teams with the highest to lowest adjusted defensive efficiency. The team ranked 1 will then be considered to have the best defense, with their defense being considered worse as the ranking drops.
AdjT	Adjusted Tempo	The "estimate of the tempo (possessions per 40 minutes) a team would have against the team that wants to play at an average D-I tempo."[6] This tells how often the team has possession of the ball.
AdjT_Rank	Adjusted Tempo Ranking	This gives the ranking of teams with the highest to lowest adjusted tempo. The team ranked 1 will then have the highest average possessions per game while the team ranked the lowest will have the lowest average possessions per game.
Luck	Luck	This measures the difference

6 https://kenpom.com/blog/ratings-glossary/

		between a team's winning percentage and what is expected from its game efficiencies. "Essentially, a team involved in a lot of close games should not win (or lose) all of them. Those that do will be viewed as lucky (or unlucky)."[7]
Luck_Rank	Luck Ranking	This is the ranking of teams with the most to least luck. A ranking of 1 means that the team had the most luck in their season, and a worse ranking means they had worse luck throughout the season.
SoS_AdjEM	Strength of Schedule Adjusted Efficiency Margin	This is the efficiency margin adjusted to consider the strength of schedule that a team has. This makes it easier to "minimize the effect of outliers"[8] as it considers teams that had easier schedules than others. If a team plays mainly tough teams then this rating isn't as sensitive to the quality of the bad teams it plays. On the other side of this, if a team plays

7 https://kenpom.com/blog/ratings-glossary/
8 https://kenpom.com/blog/ratings-glossary/

		mainly easy teams it will not have a large effect.
SoS_AdjEM_Rank	Strength of Schedule Adjusted Efficiency Margin Ranking	This gives the ranking of teams with the highest to lowest adjusted efficiency margin.
SoS_OppO	Strength of Schedule Adjusted Offensive Efficiency	This is the Adjusted Offensive Efficiency with Strength of Schedule considered to better help make it, so outliers do not affect the data as strongly.
SoS_OppO_Rank	Strength of Schedule Adjusted Offensive Efficiency Rank	This gives the ranking of teams with the highest to lowest adjusted offensive efficiency. The team with the best offense will be ranked 1 and teams with worse offenses will be ranked lower.
SoS_OppD	Strength of Schedule Adjusted Defensive Efficiency	This is the Adjusted Defensive Efficiency with Strength of Schedule considered to better help make it, so outliers do not affect the data as strongly.
SoS_OppD_Rank	Strength of Schedule Adjusted	This gives the ranking of teams with the highest to lowest adjusted defensive efficiency. The team with

	Defensive Efficiency Rank	the best defense will be ranked 1 and teams with worse defenses will be ranked lower.
NCSOS_AdjEm	Non-Conference Strength of Schedule Adjusted Efficiency Margin	This is the efficiency margin adjusted to consider the strength of schedule for non-conference games. This considers how teams do at invitationals, showcases, and other games where teams can play other teams from outside their own conference.
NCSOS_AdjEM_Rank	Non-Conference Strength of Schedule Adjusted Efficiency Margin Rank	This is the ranking of teams with the highest to lowest non-conference strength of schedule adjusted efficiency margin.

3 Simple Linear Analysis

My first attempt at finding a correlation in the data is to take a simple linear approach. Using Rank as the response, and different variables as predictors, I would like to see if there are any variables that have a strong relation with the data and are able to explain a large amount of the data. Before attempting to do this though, I will plot the data to see if there

appears to be any obvious correlations. The result of this can be seen below in Figure 1. Here a strong correlation can be seen in the plots where the data takes a very linear form, whereas weaker correlations appear more scattered.

From this, the strongest relationships appear to occur between Rank and AdjEM, Rank and AdjO, and Rank and AdjD. There are other variables that also show some relationship to Rank, such as SoS_AdjEM, SoS_OppO, and SoS_OppD. Somewhat surprisingly to me, Luck does not appear to have a correlation between rank, whereas I thought there would be some factor of a team continually being lucky and succeeding. With all of this in mind, I will start my analysis of the different variables.

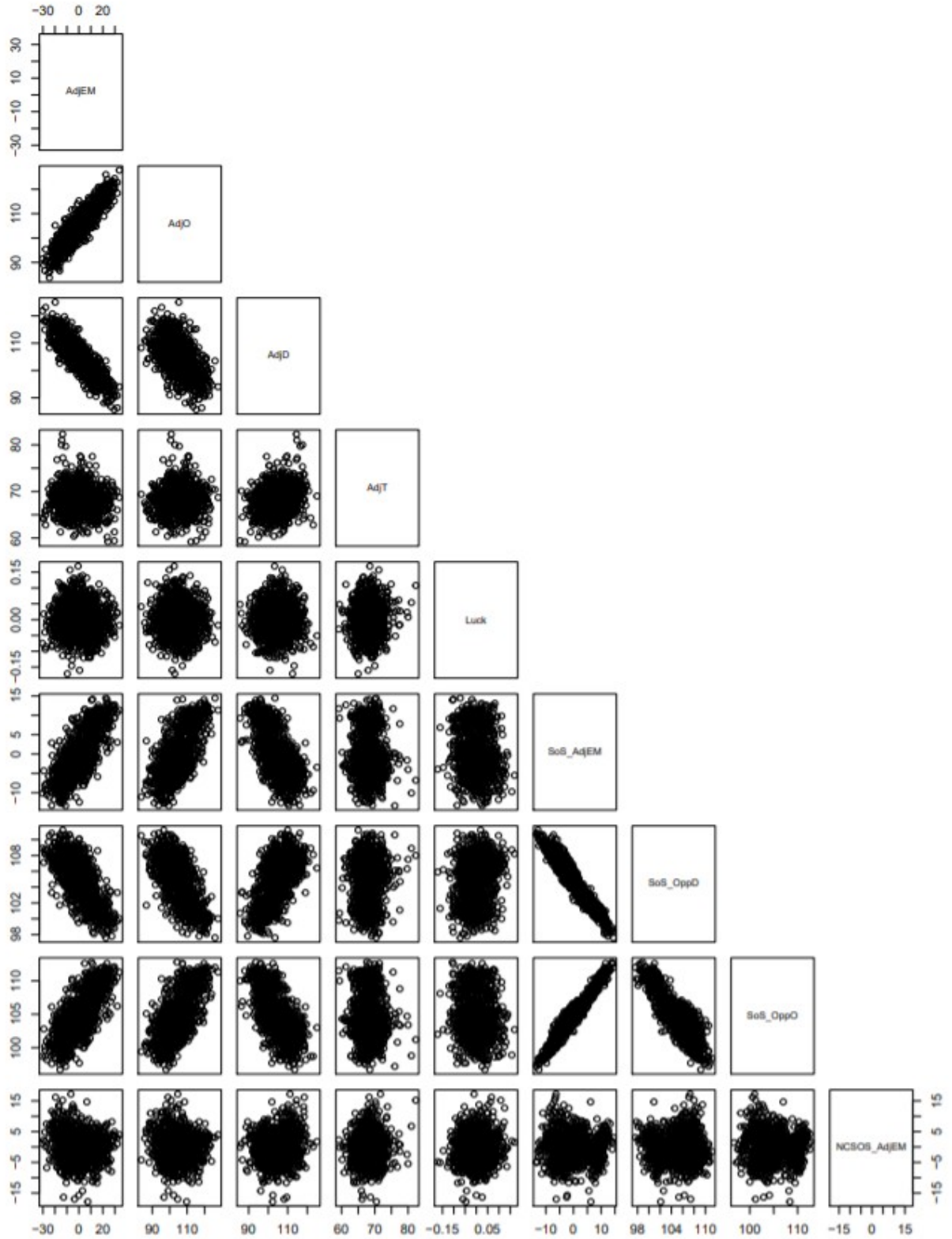

AdjEM
AdjO
AdjD
AdjT
Luck
SoS_AdjEM
SoS_OppO
SoS_OppD
NCSOS_AdjEM

Figure 1

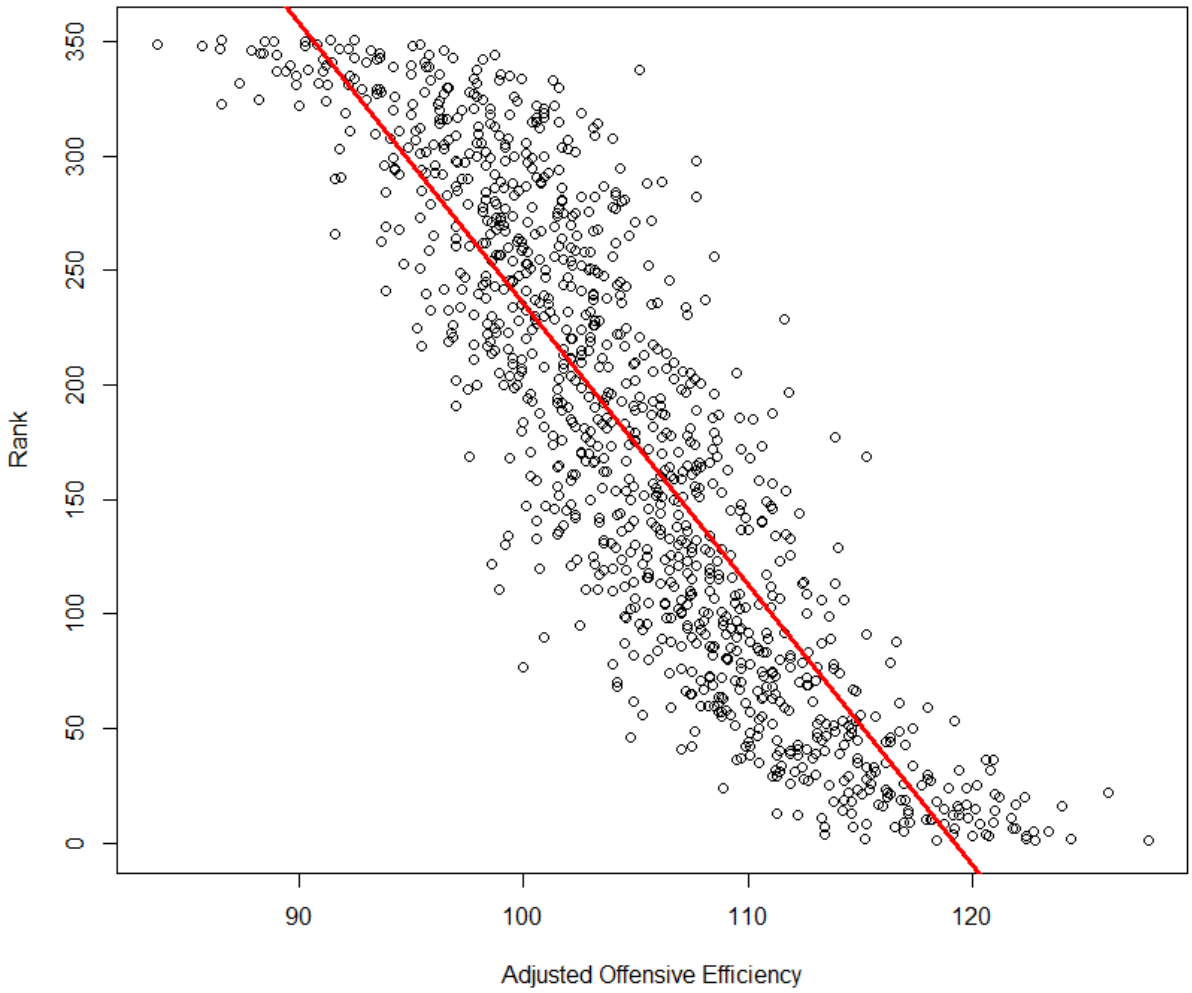
3.1 Adjusted Offensive Efficiency

Based on Figure 1, Adjusted Offensive Efficiency very clearly seems to have a linear relationship to a team's ranking, so I will start by delving into this.

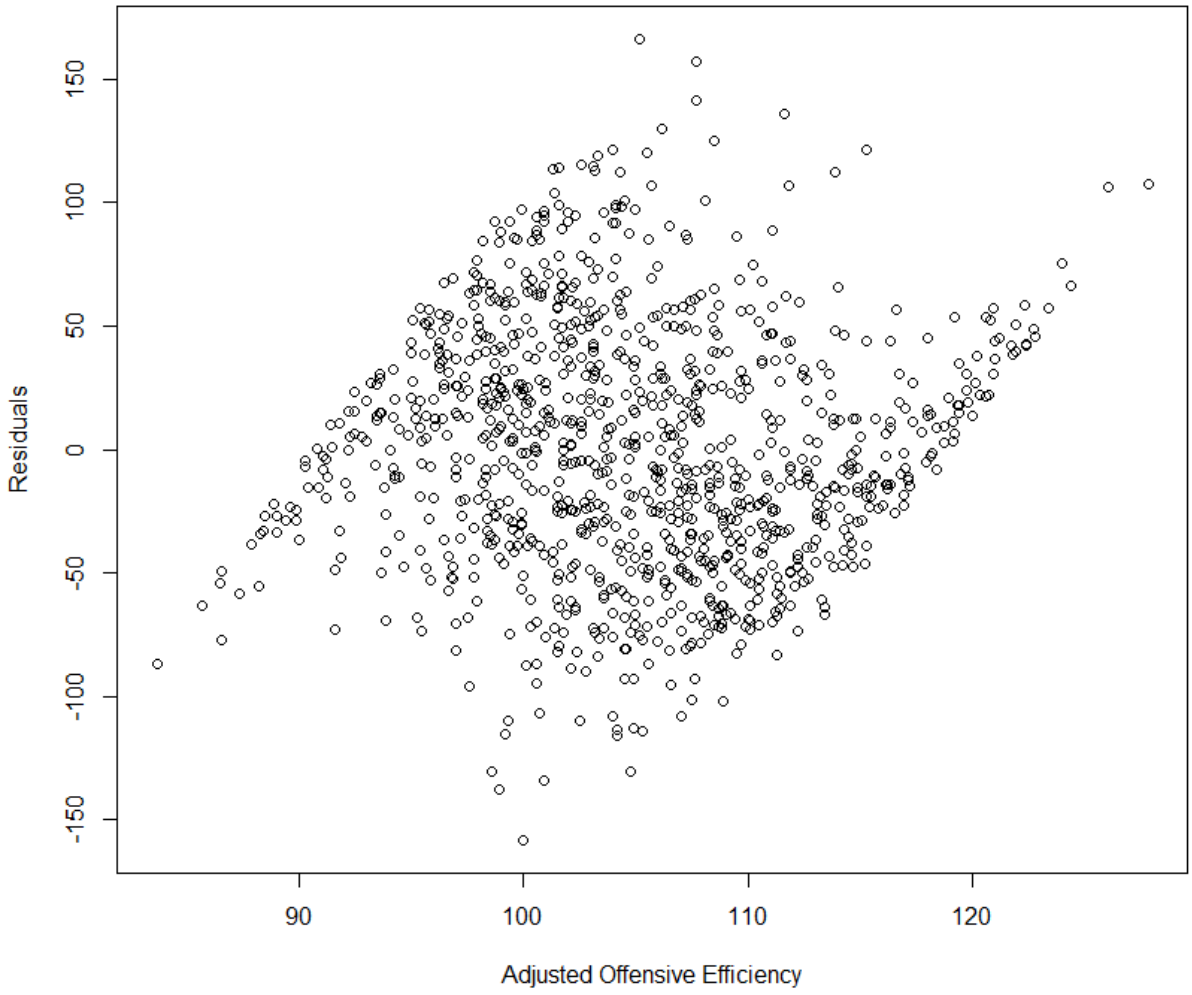
```
Call:
lm(formula = Rank ~ AdjO)

Residuals:
     Min       1Q   Median       3Q      Max
-158.441  -35.753   -2.346   34.884  166.492

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1464.9202    22.2842   65.74   <2e-16 ***
AdjO         -12.2948     0.2121  -57.98   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 49.5 on 1051 degrees of freedom
Multiple R-squared:  0.7618,    Adjusted R-squared:  0.7616
F-statistic:  3361 on 1 and 1051 DF,  p-value: < 2.2e-16
```

As seen in the above analysis, the value for the multiple R-squared and Adjusted R-squared are 0.7618 and 0.7616 respectively, both of which are better predictors the higher the value is. This is better than I expected a single factor could predict the ranking because there are so many different variables that factor into how well a team plays. However, the Residual standard error (RSE) is 49.5, which is high, while this is a value that is a better predictor when lower. Additionally, below can be seen the plot of Rank

vs AdjO with its line of best fit going through it.



The line does appear to follow the general trend of the data, but there are very large residuals, or differences between the expected and predicted position. The plot of these residuals can be seen below:



The above plot of residuals makes it clear that while using a single variable can generally predict some of the outcomes correctly, a linear fit with this variable is not the right approach. This plot can also help to explain why the RSE had such a high value.

To test the accuracy of this model for comparison's sake, I ran code which can be seen in Figure 2. I then created a second vector named rank ordered and sorted this vector from least to greatest. Finally, I created a third vector named correct which was the difference between rank and rank

order. All zero entries in the correct vector will indicate that a team's rank was predicted properly, while any non-zero entries indicate that the ranking was predicted incorrectly.

```
1  x=AdjO
2  y=AdjD
3  b0=1464.9702
4  b1=-12.2948
5  b2=0
6  rank=c()
7  for (i in 1:351){
8    rank[i]=b0+b1*x[i]+b2*y[i]
9  }
```

```
> sum(correct==0)
[1] 6
> sum(correct!=0)
[1] 345
> 6+345
[1] 351
> 6/351
[1] 0.01709402
```

Figure 2

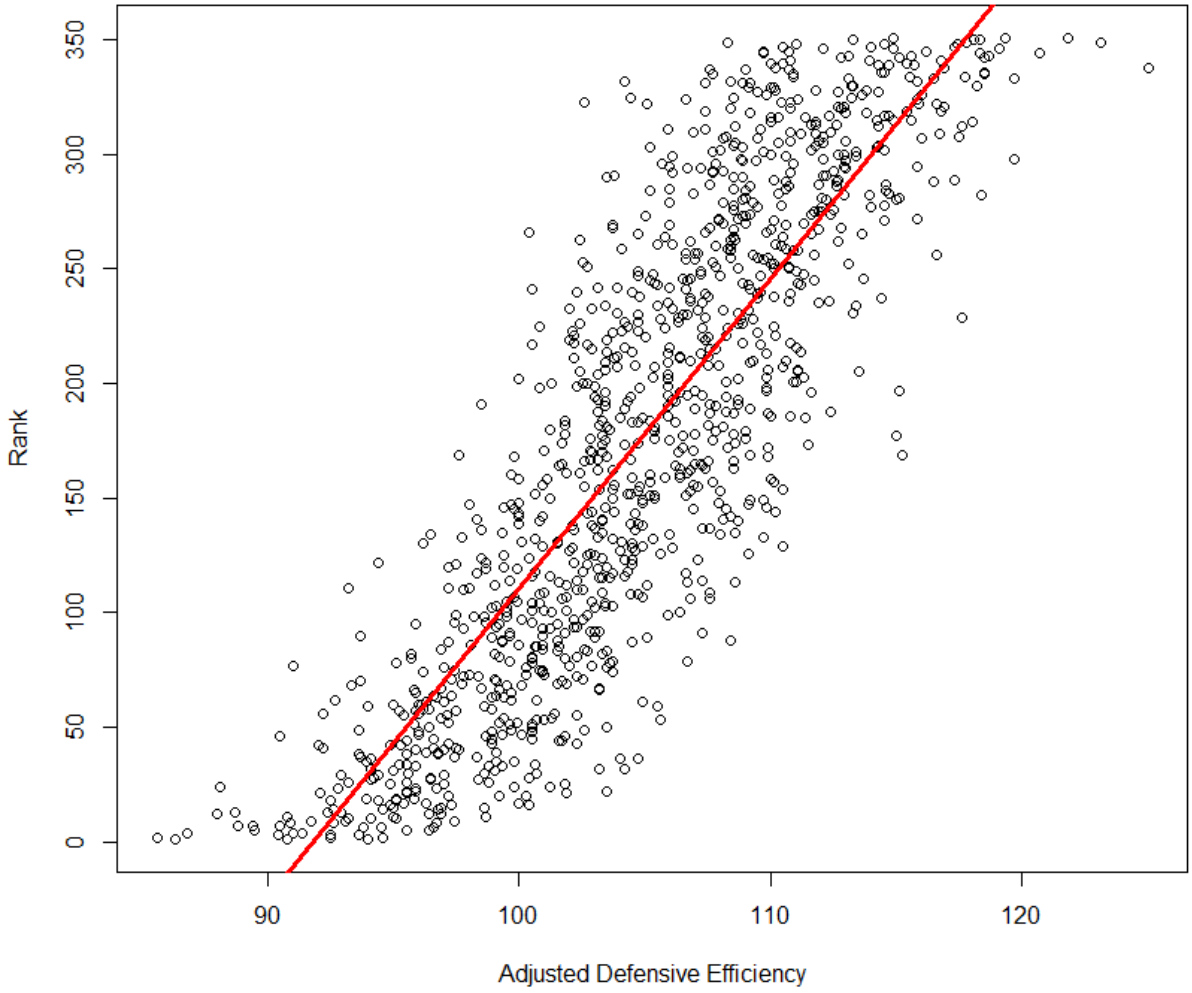
As seen in Figure 3, this model only correctly predicted 6 of the team's ranking for 2018, or 1.7%. This poor accuracy is not surprising though so to the large residuals and poor results of both the RSE and R-squared values.

3.2Adjusted Defensive Efficiency

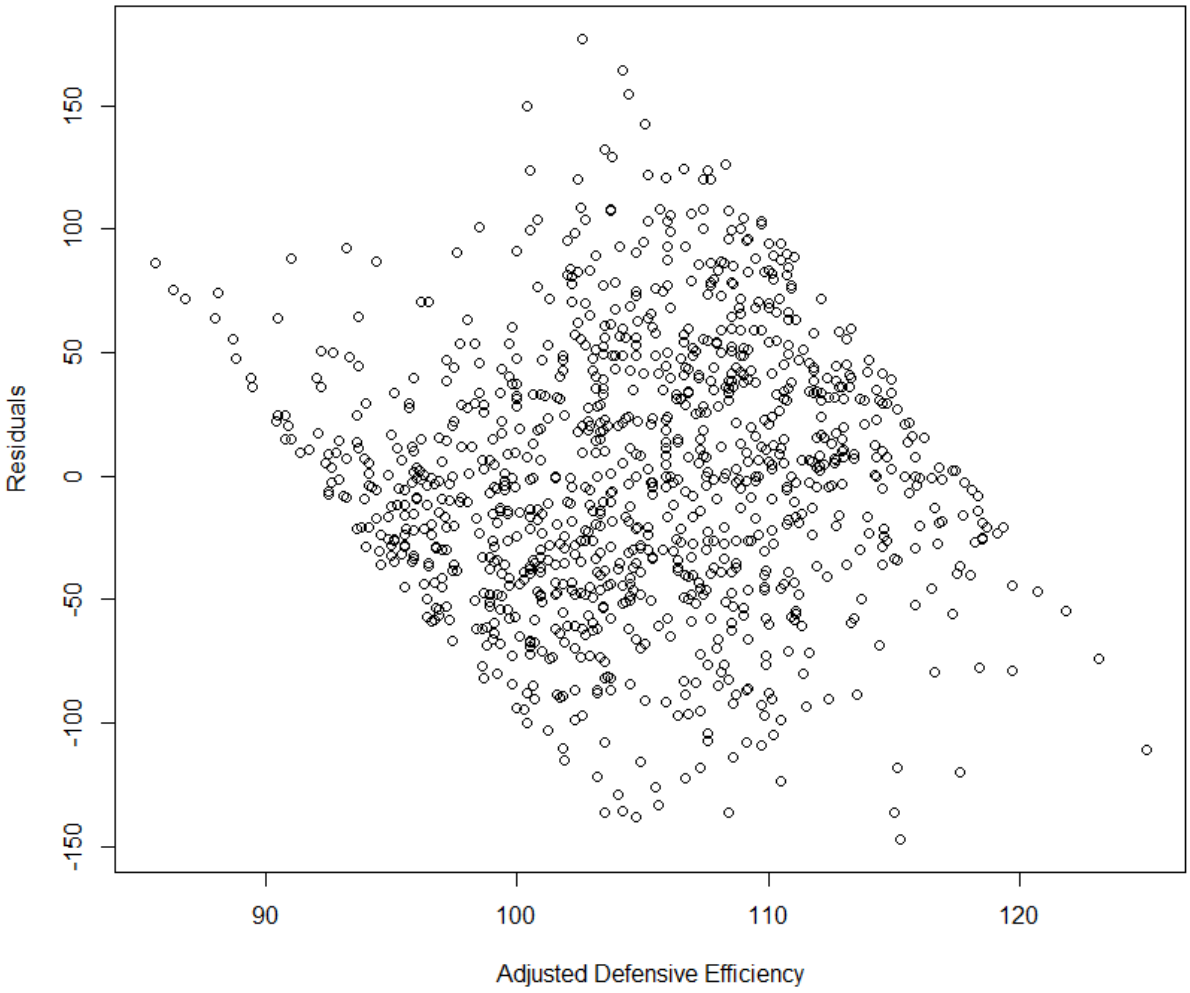
Similarly to Adjusted Offensive Efficiency, Adjusted Defensive Efficiency seemed as it there could be some correlation between the data, so I will now run the same process as in section 3.1 to see if this is a better predictor.

```
Call:
lm(formula = Rank ~ AdjD)

Residuals:
     Min       1Q   Median       3Q      Max
-147.148  -36.763   -1.138   37.113  177.247

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1241.7547    27.1237  -45.78   <2e-16
AdjD           13.5235     0.2582   52.37   <2e-16

(Intercept) ***
AdjD        ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 53.39 on 1051 degrees of freedom
Multiple R-squared:  0.7229,    Adjusted R-squared:  0.7227
F-statistic:  2742 on 1 and 1051 DF,  p-value: < 2.2e-16
```

Here it can be seen that the values for multiple R-squared and Adjusted R-squared are 0.7229 and 0.7227 respectively. This is a bit worse than the R-squared values of the Offense. Additionally the RSE rose to 53.39. This leads me to believe that a team’s defense is not as good of a linear predictor of its ranking, as compared to its offense. This makes sense in that fact that while a team does need a good defense in order to make it so the opposing team cannot score on them, they cannot win without having a strong offense, making it so this quality is often seen more in teams that succeed. Additionally, if a team's offense is strong enough, then the team will likely hold possession of the ball for the majority of the game making it so their defense isn't needed as much in the game.



As done in section 3.1, I plot the line of best fit, seen above. Again the line follows the general trend of the data, but there are still very large residuals. Due to this fact, the residuals plot can be seen below.



This once again leads me to believe that a simple linear approach is not the right method to predict the final ranking of teams. This can be further proved by running the code from Figure 2 to test how accurate it is on our known results from past years:

```
> sum(correct==0)
[1] 4
> sum(correct!=0)
[1] 347
> 4+347
[1] 351
> 4/351
[1] 0.01139601
```

Running this shows that the accuracy of this model is slightly worse than that of the offensive, predicting only 1.1% of outcomes correctly.

3.3Luck

Out of curiosity I wanted to see if there was any correlation at all between luck and rank, and if luck could reliably be used to measure how well a team performs. Based on the original graph, it could be predicted that luck will not be an accurate predictor of outcome.

```
Call:
lm(formula = Rank ~ Luck)

Residuals:
     Min       1Q   Median       3Q      Max
-176.441  -86.529   -1.221   88.403  179.049

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  175.909      3.126  56.269   <2e-16 ***
Luck          52.748     61.924   0.852    0.395
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 101.4 on 1051 degrees of freedom
Multiple R-squared:  0.0006899, Adjusted R-squared:  -0.0002609
F-statistic: 0.7256 on 1 and 1051 DF,  p-value: 0.3945
```

Unsurprisingly, Luck was found to be an insignificant value, with a very large RSE and R-squared values of almost 0. Based on this result, I will not run any further analysis on Luck as a predictor and will instead move onto other variables and methods.

3.4Other Variables

The adjusted offensive and defensive efficiencies are both clearly seen to be the most linear predictors, and as seen in sections 3.1 and 3.2, they are not able to explain the variance in the data and therefore will be poor predictors of a team's ranking at the end of the season. Knowing this, I feel

like it is safe to move on without testing the other variables, as I suspect they will only prove to show weaker correlations than the ones already seen.

4 Simple Linear Analysis

Since a simple linear model proved not to be the correct approach path to model my data, I will now attempt a multiple linear analysis. Starting off, I wanted to see which variables are most important to be used, and how many I should be using to optimize my model. To do this I performed a Best Subset Selection. This method takes into account a variety of predictors and tells the user which ones are best to predict the outcome. The results of this can be seen below:

```
Subset selection object
Call: regsubsets.formula(Rank ~ AdjO + AdjD + AdjT + Luck + SoS_OppD +
    SoS_OppO + NCSOS_AdjEM, data = KP3)
7 Variables  (and intercept)
            Forced in Forced out
AdjO            FALSE      FALSE
AdjD            FALSE      FALSE
AdjT            FALSE      FALSE
Luck            FALSE      FALSE
SoS_OppD        FALSE      FALSE
SoS_OppO        FALSE      FALSE
NCSOS_AdjEM     FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: exhaustive
         AdjO AdjD AdjT Luck SoS_OppD SoS_OppO NCSOS_AdjEM
1  ( 1 ) "*"  " "  " "  " "  " "      " "      " "
2  ( 1 ) "*"  "*"  " "  " "  " "      " "      " "
3  ( 1 ) "*"  "*"  " "  "*"  " "      " "      " "
4  ( 1 ) "*"  "*"  " "  "*"  " "      " "      "*"
5  ( 1 ) "*"  "*"  " "  "*"  " "      "*"      "*"
6  ( 1 ) "*"  "*"  " "  "*"  "*"      "*"      "*"
7  ( 1 ) "*"  "*"  "*"  "*"  "*"      "*"      "*"
```

Here it can be seen that when only using one predictor, it is best to use Adjusted Offense, and with two it is best to use Adjusted Offense and Adjusted Defense. Neither of these surprise me as they both seemed to be the most linear data and the strongest correlated. However what did surprise me though is the inclusion of Luck when using three predictors. When

analyzing Luck as a single linear predictor, is seemed completely unrelated to a team’s ranking, therefore it's inclusion either means one of two things. The first is that when combining all of these terms Luck somehow does play a role that I am unable to explain. The second is that there is a sharp drop off in the RSE and R squared values after the second term. To figure out which of these it is, I will need to compare the Residual Sum of Squares, Adjusted R-Squared, Cp/AIC, and BIC to see how many variables should be used, all of which can be seen below in Figure 4. For these tests, the best number of predictors can be seen by finding the minimum of the Residual Sum of Squares, Cp/AIC, and BIC, or the maximum of the Adjusted R-Squared.

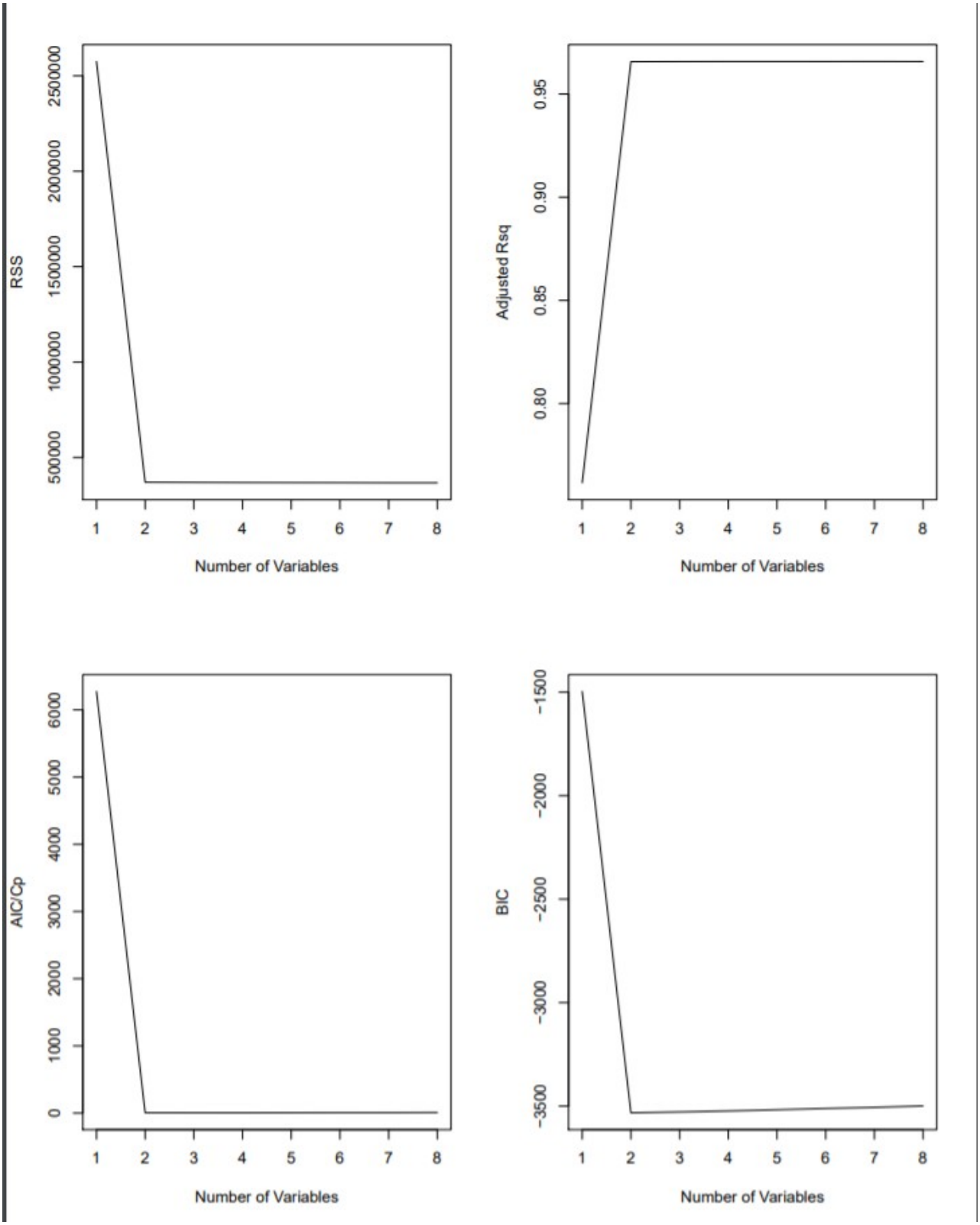

RSS
500000
1000000
1500000
2000000
2500000
1 2 3 4 5 6 7 8
Number of Variables
Adjusted Rsq
0.80
0.85
0.90
0.95
1 2 3 4 5 6 7 8
Number of Variables
AIC/Cp
0
1000
2000
3000
4000
5000
6000
1 2 3 4 5 6 7 8
Number of Variables
BIC
−3500
−3000
−2500
−2000
−1500
1 2 3 4 5 6 7 8
Number of Variables

Figure 4

Based on the above analysis, the optimal number of predictors is 2. Therefore the predictors that will be used for the model for the remainder of this section will be the Adjusted Offense and the Adjusted Defense. Running a simple analysis of these data sets provides the following information:

```
Call:
lm(formula = Rank ~ AdjO + AdjD)

Residuals:
    Min      1Q  Median      3Q     Max
-80.181 -13.881  -2.793  13.384 106.455

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 145.93255   18.69831   7.805 1.44e-14 ***
AdjO         -8.23469    0.09543 -86.290  < 2e-16 ***
AdjD          8.52134    0.10775  79.083  < 2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.78 on 1050 degrees of freedom
Multiple R-squared:  0.9658,    Adjusted R-squared:  0.9657
F-statistic: 1.481e+04 on 2 and 1050 DF,  p-value: < 2.2e-16
```

By using these two predictors rather than just one, the model has greatly increased. Now the RSE is only 18.78, which is still higher than I want to make a perfect bracket but is much improved from the models produced earlier. Similarly the Multiple R-squared and Adjusted R-squared values are 0.9658 and 0.9657 which are immensely improved upon from before.

4.1Prediction Accuracy

Knowing the optimal number of predictors, and what these predictors are, I will begin testing and implementing this model to see when used on one specific year, how many of the team's rankings it can properly predict.

To find how many of the 351 were predicted accurately for the year 2016, I used the following code which input the predicted values into a vector named rank.

```
x=AdjO
y=AdjD
b0=145.93255
b1=-8.23469
b2=8.52134
rank=c()
for (i in 1:351){
  rank[i]=b0+b1*x[i]+b2*y[i]
}
```

This code follows the same method as was run in the simple linear analysis. This leads to the following results which are produced.

```
> sum(correct==0)
[1] 149
> sum(correct!=0)
[1] 202
> 149+202
[1] 351
> 149/351
[1] 0.4245014
```

From this I was able to see that 149 of the 351 teams, or 42%, were correctly predicted for the year 2016. Considering the different variables that go into a team's rank, this is better than I expected this would be, but it still doesn't have very high odds and seems like there should be a better method. This is also a prediction for how many teams ranking it predicted out of all 351 teams, whereas March Madness only looks at the top 68 teams. If we take this into account, we now get the following:

```
> correctmm=correct[-(69:351)]
> sum(correctmm==0)
[1] 42
> sum(correctmm!=0)
[1] 26
> 42/68
[1] 0.6176471
```

This tells a few things. First, I find it important to note that the accuracy greatly increases to predicting 61% of the games correctly. This means that the model is getting worse as a team is ranked lower. I

hypothesize that this is since as teams get worse, the numerical differences between each team are much smaller and that this data set does not do a good enough job capturing the differences between these teams. This is something that I would like to further explore to see if different data sets will improve this prediction accuracy, especially regarding worse teams. To make sure this isn't only true of 2016, I will now run this with data from 2017 and 2018 to make sure it is predicting at a similar rate.

For 2017, the calculations can be seen below:

```
> sum(correct==0)
[1] 156
> sum(correct!=0)
[1] 195
> 156/351
[1] 0.4444444
> sum(correctmm==0)
[1] 43
> sum(correctmm!=0)
[1] 25
> 43/68
[1] 0.6323529
```

For 2017, it is found that the accuracy for the entire league was 44% but for the top 64 teams it was about 63%.

For 2018, the calculations can be seen below:

```
> sum(correct==0)
[1] 152
> sum(correct!=0)
[1] 199
> 152/351
[1] 0.4330484
> correctmm=correct[-(69:351)]
> sum(correctmm==0)
[1] 43
> sum(correctmm!=0)
[1] 25
> 43/68
[1] 0.6323529
```

For 2018 it is found that the accuracy for the entire league was 43% but for the top 64 teams it was about 63%.

Therefore the multiple linear regression can predict the correct outcome of NCAA basketball approximately 43% of the time with regards to the whole league, and more importantly, approximately 62.33% of the time for the top 64 teams which is the number of teams participating in March Madness.

5 Polynomial Regression

In hopes of bettering the prediction rate of the linear model, I will now implement polynomial regression.

5.1Adjusted Offensive Efficiency

I began by creating five different polynomial regression models using Rank and the outcome and Adjusted Offense as the predictor. To see which of these is the best, I looked at the analysis of variance, or ANOVA. This allows the null hypothesis to be tested and helps to suggest which polynomial regression is the most accurate.

```
> anova(fit1,fit2,fit3,fit4,fit5,fit6)
Analysis of Variance Table

Model 1: Rank ~ poly(AdjO, 1, raw = T)
Model 2: Rank ~ poly(AdjO, 2, raw = T)
Model 3: Rank ~ poly(AdjO, 3, raw = T)
Model 4: Rank ~ poly(AdjO, 4, raw = T)
Model 5: Rank ~ poly(AdjO, 5, raw = T)
Model 6: Rank ~ poly(AdjO, 6, raw = T)
  Res.Df     RSS Df Sum of Sq       F  Pr(>F)
1   1051 2575160
2   1050 2568213  1      6947  3.0849 0.07931 .
3   1049 2363549  1    204663 90.8838 < 2e-16 ***
4   1048 2363066  1       483  0.2144 0.64341
5   1047 2355520  1      7546  3.3510 0.06745 .
6   1046 2355511  1         9  0.0042 0.94831
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Looking at this, only the linear, quadratic, and quintic models have values that suggest sufficiency. I will test the cubic and quintic models using 2018 data here as I have already tested the linear model.

Starting by running analysis on the quadratic model, the following is outputted:

```
Call:
lm(formula = Rank ~ poly(AdjO, 2, raw = T), data = KP3)

Residuals:
     Min       1Q   Median       3Q      Max
-157.631  -35.938   -2.902   34.212  168.387

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           1868.00003  240.20675   7.777 1.77e-14 ***
poly(AdjO, 2, raw = T)1  -19.97451    4.56179  -4.379 1.31e-05 ***
poly(AdjO, 2, raw = T)2    0.03641    0.02160   1.685   0.0922 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49.46 on 1050 degrees of freedom
Multiple R-squared:  0.7624,    Adjusted R-squared:  0.762
F-statistic:  1685 on 2 and 1050 DF,  p-value: < 2.2e-16
```

```
> sum(correct==0)
[1] 6
> sum(correct!=0)
[1] 345
> 6/351
[1] 0.01709402
```

The quadratic model has Residual Standard Error and R-squared values that are comparable to that of the linear model, so I did not expect majorly improved performance. This was a correct assumption as it only correctly predicted six of the rankings correctly.

Now running the same analysis on the quintic model, the following is outputted:

```
Call:
lm(formula = Rank ~ poly(AdjO, 4, raw = T), data = X16_18_KP_Data)

Residuals:
    Min      1Q  Median      3Q     Max
-168.16  -31.83   -1.09   29.85  169.37

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -2.826e+04  2.038e+04  -1.387    0.166
poly(AdjO, 4, raw = T)1  9.360e+02  7.805e+02   1.199    0.231
poly(AdjO, 4, raw = T)2 -1.081e+01  1.117e+01  -0.967    0.334
poly(AdjO, 4, raw = T)3  5.073e-02  7.086e-02   0.716    0.474
poly(AdjO, 4, raw = T)4 -7.773e-05  1.680e-04  -0.463    0.644

Residual standard error: 47.49 on 1048 degrees of freedom
Multiple R-squared:  0.7814,    Adjusted R-squared:  0.7806
F-statistic: 936.6 on 4 and 1048 DF,  p-value: < 2.2e-16
```

```
> sum(correct==0)
[1] 6
> sum(correct!=0)
[1] 345
> 6+345
[1] 351
> 6/351
[1] 0.01709402
```

This quintic model did just as bad as the quadratic model, also it only predicted six of the rankings correctly.

5.2Adjusted Defensive Efficiency

After how inaccurate the polynomial regression using adjusted offense was, I wanted to test defense to see if it was just as bad. I ran the ANOVA again as seen below, and it once again suggests using the quadratic and quintic models.

```
> anova(fit1,fit2,fit3,fit4,fit5,fit6)
Analysis of Variance Table

Model 1: Rank ~ poly(AdjD, 1, raw = T)
Model 2: Rank ~ poly(AdjD, 2, raw = T)
Model 3: Rank ~ poly(AdjD, 3, raw = T)
Model 4: Rank ~ poly(AdjD, 4, raw = T)
Model 5: Rank ~ poly(AdjD, 5, raw = T)
Model 6: Rank ~ poly(AdjD, 6, raw = T)
  Res.Df     RSS Df Sum of Sq       F   Pr(>F)
1   1051 2995377
2   1050 2994153  1      1224  0.4563  0.49949
3   1049 2821178  1    172975 64.4956 2.59e-15 ***
4   1048 2818168  1      3010  1.1224  0.28965
5   1047 2808192  1      9976  3.7196  0.05405 .
6   1046 2805339  1      2852  1.0636  0.30264
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Now I will use this information to test the models using 2018 data and see how accurate they are.

Starting with the quadratic model, the following is outputted:

```
Call:
lm(formula = Rank ~ poly(AdjD, 2, raw = T), data = KP3)

Residuals:
     Min       1Q   Median       3Q      Max
-145.691  -36.838   -1.207   37.270  176.505

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1.462e+03  3.373e+02  -4.334  1.6e-05 ***
poly(AdjD, 2, raw = T)1     1.775e+01  6.459e+00   2.748  0.00609 **
poly(AdjD, 2, raw = T)2 -2.021e-02  3.085e-02  -0.655  0.51253
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.4 on 1050 degrees of freedom
Multiple R-squared:  0.723,     Adjusted R-squared:  0.7225
F-statistic:  1371 on 2 and 1050 DF,  p-value: < 2.2e-16
```

```
> sum(correct==0)
[1] 0
> sum(correct!=0)
[1] 351
> 0/351
[1] 0
```

The quadratic model is inaccurate, as it correctly predicts 0 of the rankings. Moving onto the quantum model:

```
> fit4=lm(Rank~poly(AdjD,4,raw=T),data=X16_18_KP_Data)
> summary(fit4)

Call:
lm(formula = Rank ~ poly(AdjD, 4, raw = T), data = X16_18_KP_Data)

Residuals:
     Min       1Q   Median       3Q      Max
-145.419  -34.032   -0.323   33.554  181.300

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.063e+04  3.302e+04   1.836   0.0666 .
poly(AdjD, 4, raw = T)1 -2.112e+03  1.271e+03  -1.662   0.0968 .
poly(AdjD, 4, raw = T)2  2.687e+01  1.829e+01   1.469   0.1420
poly(AdjD, 4, raw = T)3 -1.475e-01  1.167e-01  -1.264   0.2064
poly(AdjD, 4, raw = T)4  2.947e-04  2.785e-04   1.058   0.2903
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 51.86 on 1048 degrees of freedom
Multiple R-squared:  0.7393,    Adjusted R-squared:  0.7383
F-statistic: 743.1 on 4 and 1048 DF,  p-value: < 2.2e-16
```

```
> sum(correct==0)
[1] 0
> sum(correct!=0)
[1] 351
> 0/351
[1] 0
```

These models ran worse than the polynomial regression using adjusted offense, as both were unable to predict a single team's ranking correctly.

6 Conclusion

The implications of this study are that I will now be able to better predict who will win in March Madness. Additionally, it may mean that others are able to use this or similar techniques to work toward creating the perfect bracket. While this model is not perfect, I believe that using this along with my own knowledge of basketball and my judgement about how teams will play, can make it so my bracket will be better than ever. The biggest difficulty I faced with regards to this study was the sheer amount of data and parsing through it and deciding what to and not to use initially. Before

settling on this data set, there were many other data sets that offered similar statistics but used different measures. Even once I settled on this set, the fact that there were twenty different predictors to have to go through and analyze was intimidating. If I were to continue this investigation, I would like to further explore other predictors, and get more information from Ken Pomeroy about how he creates the numbers that go with each predictor. By further understanding how the predictors are derived it would help me to see if there is something in particular that drives the correlation, and if that can be manipulated to make the model even more accurate.

At the end of the day I am very happy with my study. It performed much better than I imagined. Going into the project, I picked this topic because it interested me, but I expected to find nothing, making it so the accuracy that I was able to find seems amazing. This is a topic that has taken the attention of statisticians and sports fans alike and being able to say that I have some inside information on how it works is something I am very proud of.

7 Acknowledgements

I would like to acknowledge the following people for helping somewhere along the process of this project. Ken Pomeroy, for parsing the raw data into a usable fashion and making his data freely available online. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani for writing the textbook *An Introduction to Statistical Learning*, which I heavily relied on over the course of this project. Dr. Suzanne Sindi, for teaching many of these

principles and skills in the course Math 180 and leaving a good record of examples to follow. Zachary Malone, for assisting me with the editing of this paper and for helping me come up with the initial formulation of my topic. Barack Obama, for sparking my interest in college basketball.

References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning: With applications in R*. New York: Springer.

Pomeroy, K. (n.d.). Pomeroy College Basketball Ratings. Retrieved November 19, 2018, from http://kenpom.com/