Jack Stubblefield April 23, 2018
Topics in Econometrics: Big Data Final Project

Predicting the Outcome of a San Antonio Spurs Game

Economic Question: What variables better predict a Spurs win? In particular, how good of a predictor is the difference in team salaries (Spurs opponents salary- Spurs salary)? Is there a model that can produce a high classification rate in game by game predictions of Spurs games?

Data Source: Manually compiled data from the box scores from "basketball-reference.com"

Data Description: Box score statistics of each individual regular season game for the 2016-2017 and 2017-2018 seasons along with adding an additional column for salary difference. Also added a "result" column which is binary containing either a "win" or "loss" and a "win" column which contains either a "1" or "0".

Methodologies: The "result" column is the dependent variable that I will use for the regressions tested. I will use a logistic regression for this project since I am trying to predict a win or loss correctly for each Spurs game during the season. Any prediction that gives a value of less than 0.5 will predict a "loss" for that specific game. Any prediction that gives a value greater than or equal to 0.5 will predict a "win" for that specific game. I will build five regressions based on only the 2016-2017 season data. For comparison, I will look at number of correctly predicted games out of 82, the classification rate, the corresponding residual plot, and the MSE for each regression. After comparing five different models, I will use one of the regressions to see how it wells predict the game by game wins and losses for the 2017-2018 season. Lastly, using that same regression, I will look at the total number of correctly predicted games for both seasons.

Expected Results: I expect the difference in team salaries to be somewhat significant in predicting the outcome of games but not as significant as some other variables like Assists, Opponent Three Pointers Made, and Three Point Percentage. I believe those other variables are very accurate predictors for how well the Spurs are playing during that specific game and how likely they are to win that specific game. I do expect to find a regression with some of these variables including Salary Difference that very accurately predict the outcome of Spurs games during the last two seasons.

Header for the data:

```
> head(spursdata)
  Season Win Result Game.Number Opponent Location FGM FGA FGPerc ThreePtM ThreePtA ThreePtPerc
1   2017   1    Win           1      GSW     Away  47  98  0.480       12       24       0.500
2   2017   1    Win           2      SAC     Away  36  79  0.456        6       18       0.333
3   2017   1    Win           3      NOP     Home  35  83  0.422       10       24       0.417
4   2017   1    Win           4      MIA     Away  37  82  0.451       10       18       0.556
5   2017   0   Loss           5      UTA     Home  33  76  0.434        6       20       0.300
6   2017   1    Win           6      UTA     Away  37  83  0.446        6       20       0.300
  FTM FTA FTPerc REB AST BLK STL TO Fouls OFGM O.FGA OFGPerc O3PM O.3PA O3PPerc OFTM O.FTA
1  23  26  0.885  55  25   3  13 13    19   40    85   0.471    7    33   0.212   13    18
2  24  27  0.889  40  23   8  10  9    26   28    70   0.400    6    20   0.300   32    38
3  18  23  0.783  50  21   7   6  9    15   32    86   0.372    4    22   0.182   11    15
4  22  26  0.846  44  20   5   4 15    26   37    80   0.463    7    20   0.350   18    26
5  19  21  0.905  34  19   5   6  9    15   38    76   0.500   15    31   0.484   15    18
6  20  22  0.910  49  17   8   9 11    25   30    80   0.375   11    26   0.423   15    24
  OFTPerc O.REB O.AST O.BLK O.STL O.TO O.Fouls Salary.Diff
1   0.722    35    24     6    11   16      19    -6.96503
2   0.842    40    22     0     7   15      21   -12.59753
3   0.733    45    14     5     6    9      19    -6.92801
4   0.692    36    19     6     7   12      24    -6.82222
5   0.833    39    22     8     5   10      20   -28.04243
6   0.625    42    15     4     7   12      19   -28.04243
```

Logistic Regression 1:

```
> spursLR=glm(Result~FGA+REB+AST+O3PM+O.AST+Salary.Diff, data=spursdata, subset=1:82,family=binomial)
> summary(spursLR)

Call:
glm(formula = Result ~ FGA + REB + AST + O3PM + O.AST + Salary.Diff,
    family = binomial, data = spursdata, subset = 1:82)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4058  -0.1432   0.1005   0.3960   1.5425

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.04885    4.80031   0.427  0.66951
FGA         -0.22451    0.08133  -2.761  0.00577 **
REB          0.29797    0.10159   2.933  0.00336 **
AST          0.50737    0.13014   3.899 9.67e-05 ***
O3PM        -0.26518    0.13114  -2.022  0.04316 *
O.AST       -0.22610    0.10645  -2.124  0.03367 *
Salary.Diff -0.11330    0.04097  -2.766  0.00568 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 46.836  on 75  degrees of freedom
AIC: 60.836

Number of Fisher Scoring iterations: 7
```

We see that all of the variables in this regression are considered statistically significant. I chose these variables since all of them I previously thought would have a significant effect on how the Spurs are playing in that particular game. For example, the Spurs focus on trying to eliminate their opponents three point attempts so the model shows us that the more three pointers their opponent makes, the worse the Spurs chances of a win becomes.

Also, we see the higher the Spurs salary is compared to their opponent, their probability of a win increases according to this model.

```
MyPredictions1617=predict(spursLR, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
```

```
> cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
   [,1]       [,2]
1     1 0.93250591
2     1 0.94063056
3     1 0.99603594
4     1 0.82021618
5     0 0.26546038
6     1 0.97093894
7     0 0.15711715
8     0 0.43144892
9     1 0.99275787
10    1 0.30434289
11    1 0.99764364
12    1 0.91076951
13    1 0.93463506
14    1 0.97802554
15    1 0.37057750
16    1 0.67937890
17    1 0.88803567
```

Above are the first 17 of the 82 results that are printed. The first column shows the actual result of the game and the second column shows the prediction for the probability of a Spurs win. Below is the predicted outcome of each of the 82 games of the 2016-2017 season using this model.

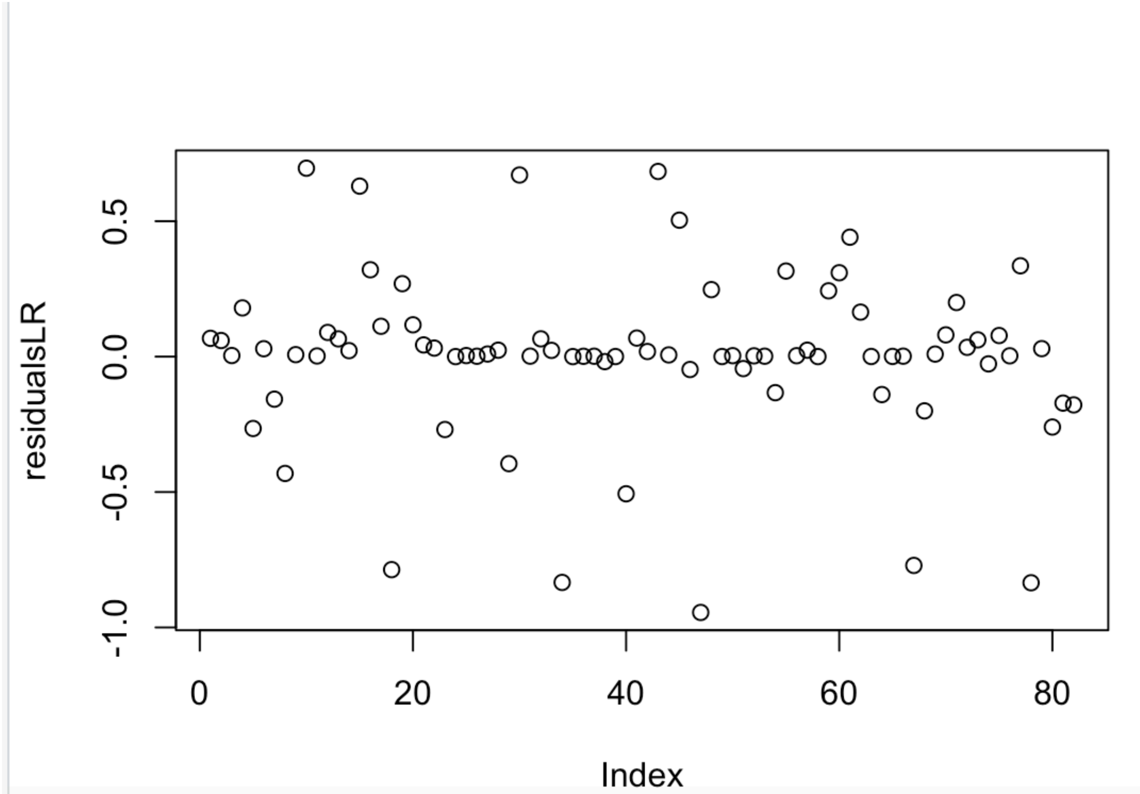
```
> PredictedResult1617
 [1] "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Loss" "Loss" "Win"  "Loss" "Win"  "Win"
[13] "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"
[25] "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[37] "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Loss" "Loss" "Win"  "Win"
[49] "Win"  "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[61] "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"
[73] "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Loss"
```

```
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 62 wins,56 of which were actually wins
```

```
> sum(PredictedResult1617==Result[1:82])
[1] 71
> #displays percentage of 82 games that are predicted correctly
> mean(PredictedResult1617==Result[1:82])
[1] 0.8658537
> #displays misclassification rate
> 1-mean(PredictedResult1617==Result[1:82])
[1] 0.1341463
>
> table(PredictedResult1617,Result[1:82])

PredictedResult1617 Loss Win
               Loss   15   5
               Win     6  56
```

The data above shows that this particular model correctly predicted 71 of the 82 games of the 2016-2017 season. The misclassification rate is low at 13.41%. The table shows that this model predicted 62 wins for the Spurs that season, 56 of which were actually wins while it predicted 20 losses for the Spurs that season, 15 of which were actually losses.



The residual plot above shows randomly distributed points with many being very accurate. Below is the MSE of this logistic regression

```
> y=Win[1:82]
> modelpredict=MyPredictions1617[1:82]
> mean((y-modelpredict)^2)
[1] 0.09152417
```

Logistic Regression 2:

```
> spursLR2=glm(Result~ThreePtPerc+O3PM,data=spursdata,subset=1:82,family=binomial)
> summary(spursLR2)

Call:
glm(formula = Result ~ ThreePtPerc + O3PM, family = binomial,
    data = spursdata, subset = 1:82)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0485  -0.4824   0.3932   0.7813   1.8315

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.70091    1.43312  -0.489  0.62478
ThreePtPerc 10.20322    3.63089   2.810  0.00495 **
O3PM        -0.24040    0.09539  -2.520  0.01173 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 77.121  on 79  degrees of freedom
AIC: 83.121

Number of Fisher Scoring iterations: 5
```

We see that the two variables in this regression are considered statistically significant. I chose these variables since the three pointer has become so powerful and heavily utilized in todays NBA. I wanted to see the affect of just the Spurs three point percentage and their opponents total three pointers made. In addition to limiting opponents three pointers made, the Spurs, who still do not take many threes, would be better estimating win probability with three point percentage.

```
MyPredictions1617=predict(spursLR2, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
```

```
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 72 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])
```

```
> cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
   [,1]      [,2]
1     1 0.9380755
2     1 0.7780561
3     1 0.9303605
4     1 0.9640597
5     0 0.2234083
6     1 0.4293945
7     0 0.7714992
8     0 0.4992222
9     1 0.9363475
10    1 0.1868841
11    1 0.9776694
12    1 0.7061355
13    1 0.6605297
14    1 0.8111031
15    1 0.7177811
16    1 0.7485304
17    1 0.8432894
18    0 0.6004899
```

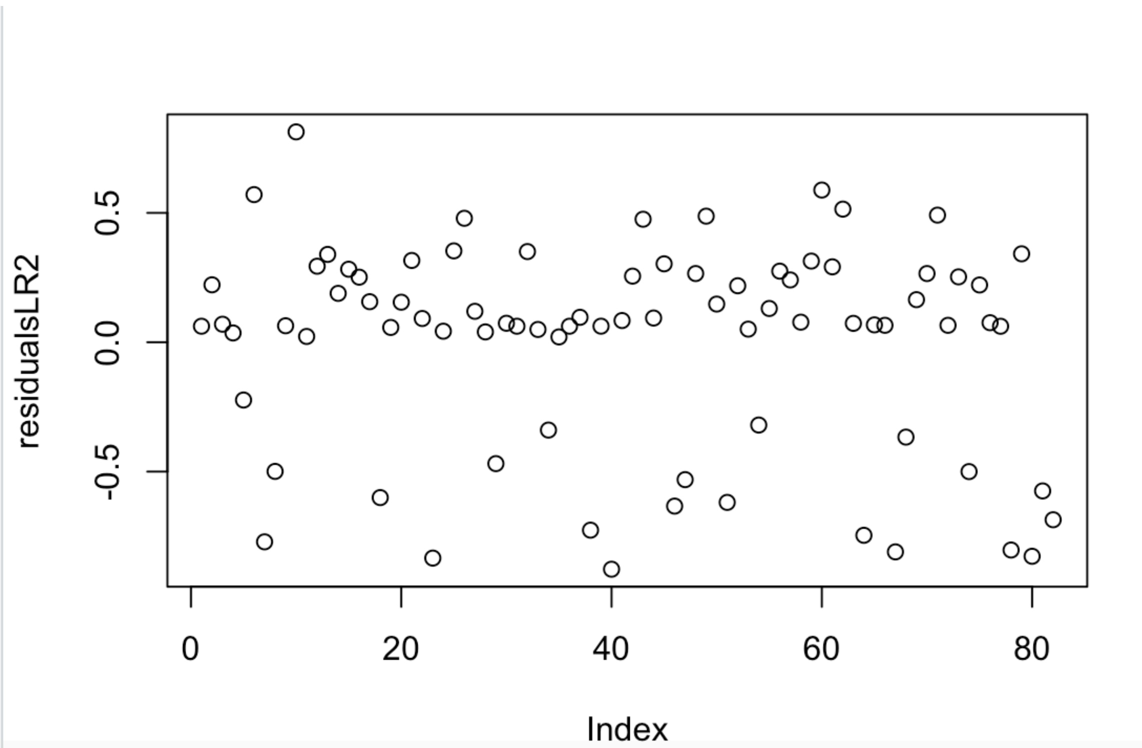
Above are the first 18 of the 82 results that are printed. The first column shows the actual result of the game and the second column shows the prediction for the probability of a Spurs win. Below is the predicted outcome of each of the 82 games of the 2016-2017 season using this model.

```
> PredictedResult1617
 [1] "Win"  "Win"  "Win"  "Wi predict(object, ...) s" "Win"  "Loss" "Win"  "Loss" "Win"  "Win"
[13] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[25] "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"
[37] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[49] "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss"
[61] "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"
[73] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
```

```
> #displays the count of games on which PredictedResult==Result
> sum(PredictedResult1617==Result[1:82])
[1] 63
> #displays percentage of 82 games that are predicted correctly
> mean(PredictedResult1617==Result[1:82])
[1] 0.7682927
> #displays misclassification rate
> 1-mean(PredictedResult1617==Result[1:82])
[1] 0.2317073
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 62 wins,56 of which were actually wins
> table(PredictedResult1617,Result[1:82])

PredictedResult1617 Loss Win
               Loss    6   4
               Win    15  57
```

The data above shows that this particular model correctly predicted 63 of the 82 games of the 2016-2017 season. The misclassification rate is pretty low at 23.17%. The table shows that this model predicted 72 wins for the Spurs that season, 57 of which were actually wins while it predicted 10 losses for the Spurs that season, 6 of which were actually losses.



The residual plot above shows a gap in residual values. This means that some of the predictions are way off. Since the gap is for points that are at -0.5 or below, it means the model is predicting some high probabilities for Spurs wins when they actually lost the game in those cases. The MSE for this model is higher than the previous model.

```
> y=Win[1:82]
> modelpredict=MyPredictions1617[1:82]
> mean((y-modelpredict)^2)
[1] 0.1575897
```

Logistic Regression 3:

```
> spursLR3=glm(Result~AST,data=spursdata,subset=1:82,family=binomial)
> summary(spursLR3)

Call:
glm(formula = Result ~ AST, family = binomial, data = spursdata,
    subset = 1:82)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2344  -0.4149   0.4645   0.7782   1.3722

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.25706    1.50932  -2.821 0.004795 **
AST          0.23812    0.06976   3.413 0.000642 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 76.529  on 80  degrees of freedom
AIC: 80.529

Number of Fisher Scoring iterations: 5
```

We see that the variable Assists is statistically significant. I chose this variable since teams like the Spurs and Warriors who have been so good over the last few years have been at the top of the league in Assists per game.

```
MyPredictions1617=predict(spursLR3, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
```

```
> cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
   [,1]      [,2]
1     1 0.8450023
2     1 0.7720089
3     1 0.6777499
4     1 0.6237113
5     0 0.5664081
6     1 0.4479324
7     0 0.5664081
8     0 0.7274196
9     1 0.8111974
10    1 0.6777499
11    1 0.8111974
12    1 0.9339128
13    1 0.9471764
14    1 0.8111974
15    1 0.9176085
16    1 0.8111974
17    1 0.7274196
```

Above are the first 17 of the 82 results that are printed. The first column shows the actual result of the game and the second column shows the prediction for the probability of a Spurs win. Below is the predicted outcome of each of the 82 games of the 2016-2017 season using this model.

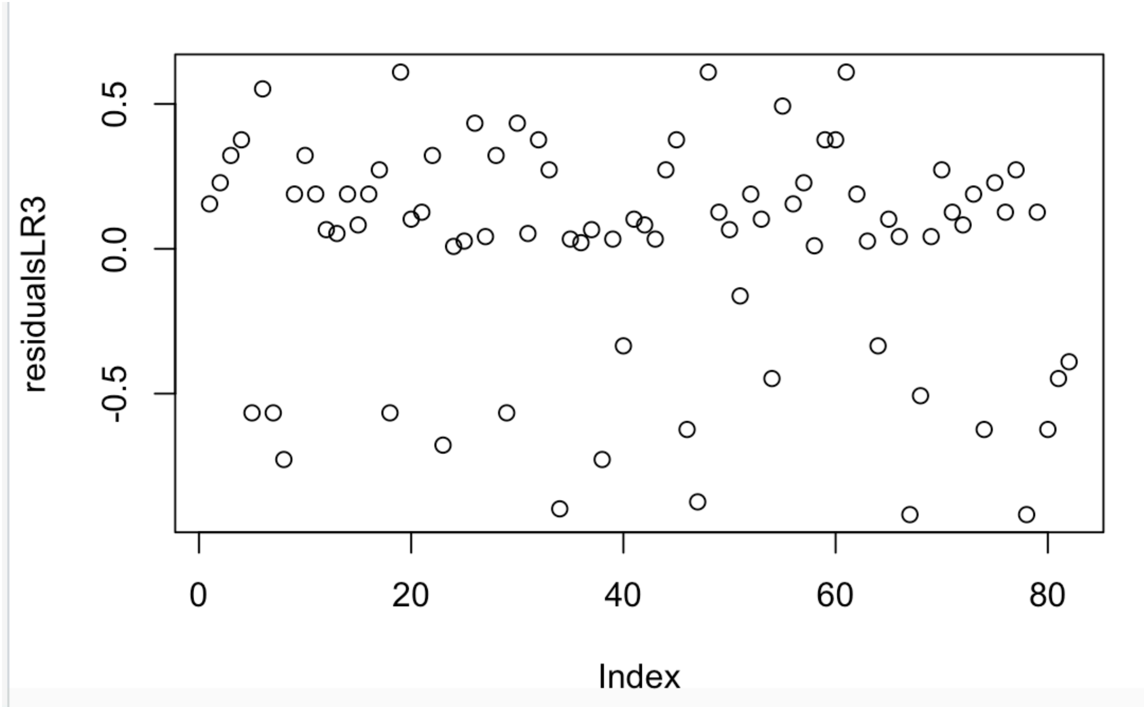
```
> PredictedResult1617
 [1] "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[13] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"
[25] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[37] "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss"
[49] "Win"  "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[61] "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[73] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Loss"
```

```
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 72 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])
```

```
> #displays the count of games on which PredictedResult==Result
> sum(PredictedResult1617==Result[1:82])
[1] 63
> #displays percentage of 82 games that are predicted correctly
> mean(PredictedResult1617==Result[1:82])
[1] 0.7682927
> #displays misclassification rate
> 1-mean(PredictedResult1617==Result[1:82])
[1] 0.2317073
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 72 wins,57 of which were actually wins
> table(PredictedResult1617,Result[1:82])

PredictedResult1617 Loss Win
               Loss    6   4
               Win    15  57
```

The data above shows that this particular model correctly predicted 63 of the 82 games of the 2016-2017 season. The misclassification rate is pretty low at 23.17%. The table shows that this model predicted 72 wins for the Spurs that season, 57 of which were actually wins while it predicted 10 losses for the Spurs that season, 6 of which were actually losses.



Similar to logistic regression 2, this residual plot shows a gap between values on the y axis which means some of the predictions in this model are way off. Similar to model 2, with so many predictions with a residual of an less than -0.5, it means the model gave the Spurs a high probability of a win when they lost that specific game. The MSE is pretty comparable to the previous model.

```
> y=Win[1:82]
> modelpredict=MyPredictions1617[1:82]
> mean((y-modelpredict)^2)
[1] 0.1517373
```

Logistic Regression 4:

```
> spursLR4=glm(Result~AST+Salary.Diff, data=spursdata,subset=1:82,family=binomial)
> summary(spursLR4)

Call:
glm(formula = Result ~ AST + Salary.Diff, family = binomial,
    data = spursdata, subset = 1:82)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3237  -0.3053   0.4042   0.7411   1.5041

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.00645    1.66363  -3.009 0.002618 **
AST          0.25413    0.07441   3.415 0.000637 ***
Salary.Diff -0.04279    0.02525  -1.695 0.090163 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 73.546  on 79  degrees of freedom
AIC: 79.546
```

We see that the Assist variable in this model is statistically significant while salary difference is not quite significant in this model. I chose these variables to see the effect of adding the variable salary difference onto the already known important variable assists. I wanted to see how adding salary difference would affect the total number of correctly predicted games.

```
MyPredictions1617=predict(spursLR4, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
```

```
> cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
   [,1]       [,2]
1     1 0.83819873
2     1 0.79860766
3     1 0.65175686
4     1 0.59100253
5     0 0.73534538
6     1 0.62566144
7     0 0.37684434
8     0 0.79587725
9     1 0.74625102
10    1 0.75149293
11    1 0.79973567
12    1 0.94796557
13    1 0.95845569
14    1 0.78441011
15    1 0.94781254
16    1 0.85094610
17    1 0.70146845
```

Above are the first 17 of the 82 results that are printed. The first column shows the actual result of the game and the second column shows the prediction for the probability of a Spurs win. Below is the predicted outcome of each of the 82 games of the 2016-2017 season using this model.

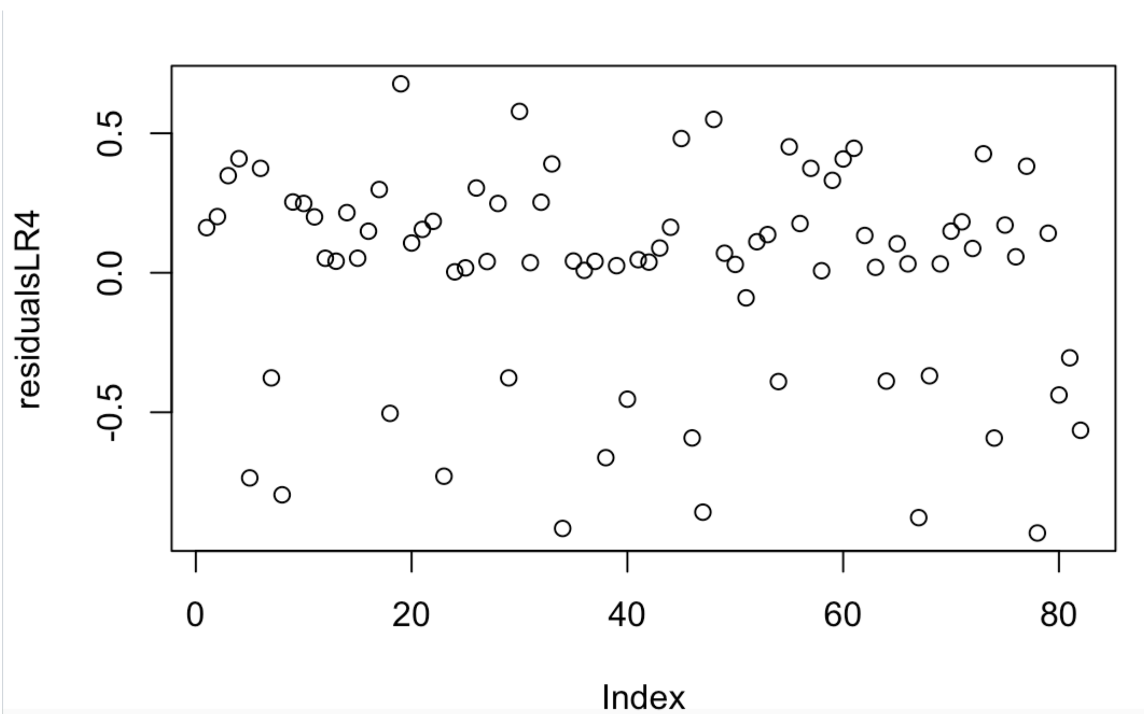
```
> PredictedResult1617
 [1] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"
[13] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"
[25] "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[37] "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss"
[49] "Win"  "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Win"
[61] "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"
[73] "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Win"
```

```
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 70 wins,58 of which were actually wins
table(PredictedResult1617,Result[1:82])
```

```
> #displays the count of games on which PredictedResult==Result
> sum(PredictedResult1617==Result[1:82])
[1] 67
> #displays percentage of 82 games that are predicted correctly
> mean(PredictedResult1617==Result[1:82])
[1] 0.8170732
> #displays misclassification rate
> 1-mean(PredictedResult1617==Result[1:82])
[1] 0.1829268
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 64 wins,54 of which were actually wins
> table(PredictedResult1617,Result[1:82])

PredictedResult1617 Loss Win
               Loss    9   3
               Win    12  58
```

The data above shows that this particular model correctly predicted 67 of the 82 games of the 2016-2017 season. The misclassification rate is pretty low at 18.29%. The table shows that this model predicted 70 wins for the Spurs that season, 58 of which were actually wins while it predicted 12 losses for the Spurs that season, 9 of which were actually losses.



Similar to the two previous models, there is a gap in the residuals show some of the predicitons as being way off. Many of these points at the bottom are games in which the model predicts a high probability for a Spurs win and they lose the game. The MSE for this is slightly lower than in logistic regression 2 and 3.

```
> y=Win[1:82]
> modelpredict=MyPredictions1617[1:82]
> mean((y-modelpredict)^2)
[1] 0.1446324
```

Logistic Regression 5:

```
> spursLR5=glm(Result~REB+AST+ThreePtPerc+STL+TO+OFGPerc+O3PM+OFTPerc+O.AST-
spursdata,subset=1:82,family=binomial)
> summary(spursLR5)

Call:
glm(formula = Result ~ REB + AST + ThreePtPerc + STL + TO + OFGPerc +
    O3PM + OFTPerc + O.AST + Salary.Diff, family = binomial,
    data = spursdata, subset = 1:82)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.20248  -0.02399   0.06172   0.24040   1.32882

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  18.67321   10.82811   1.725  0.08462 .
REB           0.03316    0.07945   0.417  0.67636
AST           0.40854    0.14682   2.783  0.00539 **
ThreePtPerc  17.09797    8.45393   2.022  0.04313 *
STL           0.14758    0.20736   0.712  0.47664
TO           -0.02789    0.16887  -0.165  0.86881
OFGPerc     -48.02905   19.24481  -2.496  0.01257 *
O3PM         -0.24550    0.18512  -1.326  0.18478
OFTPerc      -8.93942    5.36891  -1.665  0.09591 .
O.AST        -0.23514    0.14288  -1.646  0.09981 .
Salary.Diff  -0.08504    0.05562  -1.529  0.12625
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 36.187  on 71  degrees of freedom
AIC: 58.187
```

We see that some of the variables in this regression like Assists, Three Point Percentage, and Opponent Field Goal Percentage are considered statistically significant. I chose these variables since I thought these variables would have a significant effect on how the Spurs are playing in that particular game. In this model, salary difference is not considered statistically significant.

```
MyPredictions1617=predict(spursLR5, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
```

```
> cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
   [,1]         [,2]
1     1 0.9931043739
2     1 0.9820317678
3     1 0.9998104592
4     1 0.9811452309
5     0 0.0099011671
6     1 0.9974092864
7     0 0.0012717448
8     0 0.3711392736
9     1 0.9986596276
10    1 0.7051905435
11    1 0.9994991918
12    1 0.9730374992
13    1 0.9760166243
14    1 0.9762940730
15    1 0.6369022366
16    1 0.4135884197
17    1 0.6883776466
```

Above are the first 17 of the 82 results that are printed. The first column shows the actual result of the game and the second column shows the prediction for the probability of a Spurs win. Below is the predicted outcome of each of the 82 games of the 2016-2017 season using this model.

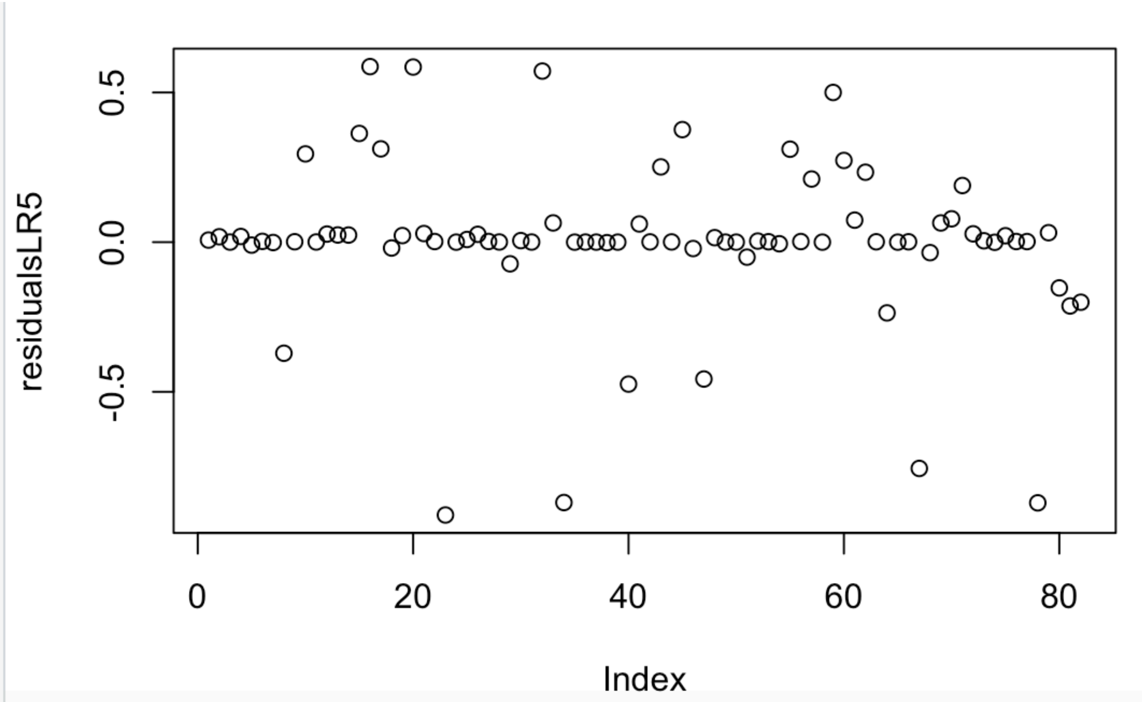
```
> PredictedResult1617
 [1] "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Loss" "Loss" "Win"  "Win"  "Win"  "Win" 
[13] "Win"  "Win"  "Win"  "Loss" "Win"  "Loss" "Win"  "Loss" "Win"  "Win"  "Win"  "Win" 
[25] "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win" 
[37] "Win"  "Loss" "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Win" 
[49] "Win"  "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Loss" "Win" 
[61] "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"  "Win" 
[73] "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Loss"
```

```
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 61 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])
```

```
> #displays the count of games on which PredictedResult==Result
> sum(PredictedResult1617==Result[1:82])
[1] 74
> #displays percentage of 82 games that are predicted correctly
> mean(PredictedResult1617==Result[1:82])
[1] 0.902439
> #displays misclassification rate
> 1-mean(PredictedResult1617==Result[1:82])
[1] 0.09756098
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 63 wins,58 of which were actually wins
> table(PredictedResult1617,Result[1:82])

PredictedResult1617 Loss Win
               Loss   17   4
               Win     4  57
```

The data above shows that this particular model correctly predicted 74 of the 82 games of the 2016-2017 season. The misclassification rate is very low at 9.76%. The table shows that this model predicted 61 wins for the Spurs that season, 57 of which were actually wins while it predicted 21 losses for the Spurs that season, 17 of which were actually losses.



Similar to the first logistic regression the residuals above are more randomly distributed with many being very close to zero. The MSE of this model is the smallest out of all of the models.

```
> y=Win[1:82]
> modelpredict=MyPredictions1617[1:82]
> mean((y-modelpredict)^2)
[1] 0.07053186
```

Using Logistic Regression 1 for 2017-2018 Season Predictions:

Decided to use logistic regression 1 since it had the second lowest misclassification rate and second lowest MSE while also having all six of the explanatory variables as statistically significant.

```
MyPredictions1718=predict(spursLR, type="response",spursdata)
#compare actual wins with MyPredictions
cbind(Result[83:164]=="Win",MyPredictions1718[83:164])
#makes an array of "Loss" for all of the games from the 2017-2018 season
PredictedResult1718=array("Loss",dim(spursdata)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1718[MyPredictions1718 >=0.5]="Win"
PredictedResult1718
```

```
> cbind(Result[83:164]=="Win",MyPredictions1718[83:164])
     [,1]         [,2]
83      1 0.6586545171
84      1 0.9979441608
85      1 0.7905132410
86      1 0.2385936966
87      0 0.6085516133
88      0 0.4145697374
89      0 0.1665729344
90      0 0.0008663717
91      1 0.9987592579
92      1 0.7015779652
93      1 0.9781841216
94      0 0.6485284622
95      1 0.9999069486
96      1 0.9845596961
97      0 0.3160968666
98      1 0.4190813408
99      1 0.9992459025
100     0 0.0835725275
```

Above is the results from the first 17 games of the 2017-2018 season which the left column shows the actual result and the right column showing this model's prediction for probability of a win in that game. Below is the predicted results for all 82 games of the 2017-2018 season using the logistic regression 1 created with 2016-2017 season data.

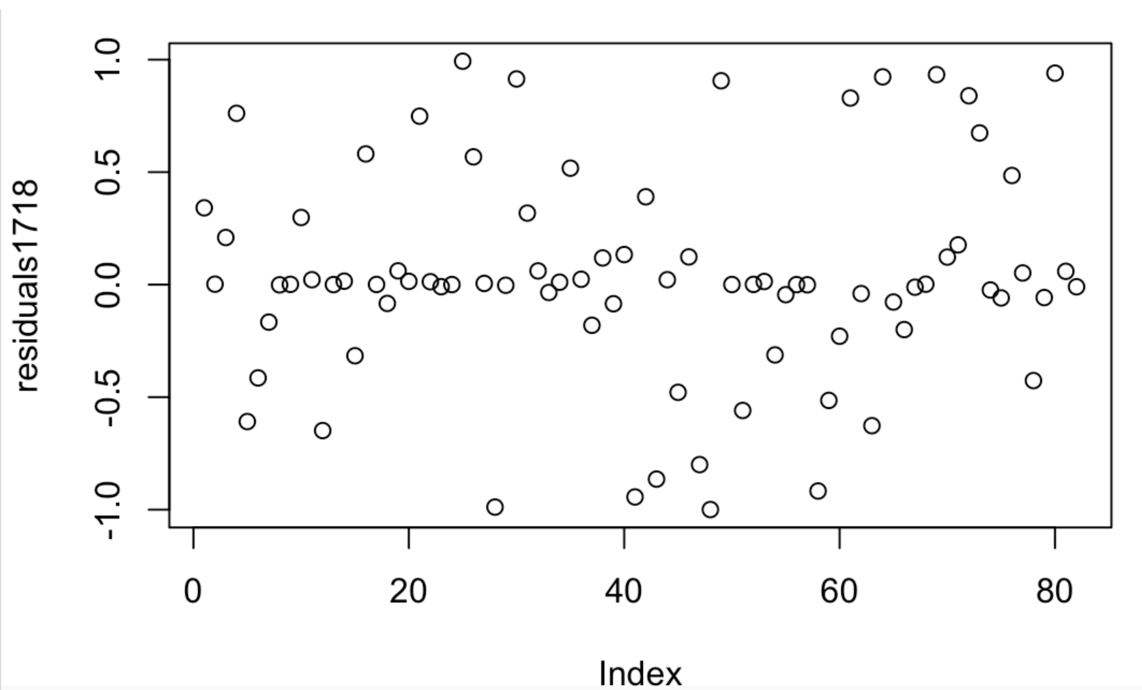
```
> PredictedResult1718[83:164]
 [1] "Win"  "Win"  "Win"  "Loss" "Win"  "Loss" "Loss" "Loss" "Win"  "Win"  "Win"  "Win"
[13] "Win"  "Win"  "Loss" "Loss" "Win"  "Loss" "Win"  "Win"  "Loss" "Win"  "Loss" "Win"
[25] "Loss" "Loss" "Win"  "Win"  "Loss" "Loss" "Win"  "Win"  "Loss" "Win"  "Loss" "Win"
[37] "Loss" "Win"  "Loss" "Win"  "Win"  "Win"  "Win"  "Win"  "Loss" "Win"  "Win"  "Win"
[49] "Loss" "Win"  "Win"  "Win"  "Win"  "Loss" "Loss" "Win"  "Loss" "Win"  "Win"  "Loss"
[61] "Loss" "Loss" "Win"  "Loss" "Loss" "Loss" "Loss" "Win"  "Loss" "Win"  "Win"  "Loss"
[73] "Loss" "Loss" "Loss" "Win"  "Win"  "Loss" "Loss" "Loss" "Win"  "Loss"
```

```
#displays the count of games which our prediction
#equals the actual outcome for the game for 2018 using 2017 regression
sum(PredictedResult1718[83:164]==Result[83:164])
#displays percentage of 82 games from 2017-2018 season that are predicted correctly
mean(PredictedResult1718[83:164]==Result[83:164])
#displays misclassification rate
1-mean(PredictedResult1718[83:164]==Result[83:164])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 44 wins,33 of which were actually wins
table(PredictedResult1718[83:164],Result[83:164])
```

```
> #displays the count of games which our prediction
> #equals the actual outcome for the game for 2018 using 2017 regression
> sum(PredictedResult1718[83:164]==Result[83:164])
[1] 57
> #displays percentage of 82 games from 2017-2018 season that are predicted correctly
> mean(PredictedResult1718[83:164]==Result[83:164])
[1] 0.695122
> #displays misclassification rate
> 1-mean(PredictedResult1718[83:164]==Result[83:164])
[1] 0.304878
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 44 wins,33 of which were actually wins
> table(PredictedResult1718[83:164],Result[83:164])

       Loss Win
  Loss   24  14
  Win    11  33
```

The data above shows that this particular model correctly predicted 57 of the 82 games of the 2016-2017 season. The misclassification rate is much higher at 30.49% but not bad considering the regression is based off of last season's data. The table shows that this model predicted 44 wins for the Spurs that season, 33 of which were actually wins while it predicted 38 losses for the Spurs that season, 24 of which were actually losses.



Residual plot above shows random distribution of points and much more wider range of residuals. The MSE for this regression on 2017-2018 data is also much greater.

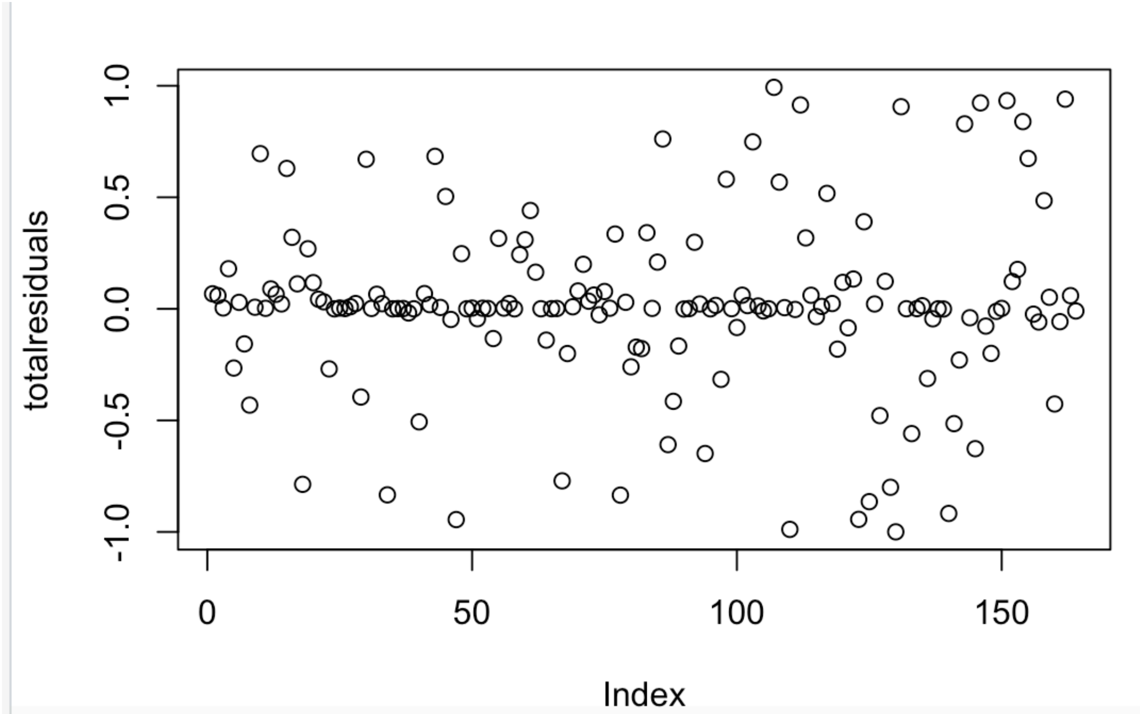
```
> y2=Win[83:164]
> modelpredict2=MyPredictions1718[83:164]
> mean((y2-modelpredict2)^2)
[1] 0.2174843
```

Combining Results for Both Seasons:

```
> #displays total count of games for both seasons
> #using regression from 2017 season data
> sum(PredictedResult1718==Result[1:164])
[1] 128
> #displays percentage of 164 games from both seasons that are predicted correctly
> mean(PredictedResult1718==Result[1:164])
[1] 0.7804878
> #displays misclassification rate for both seasons
> 1-mean(PredictedResult1718[1:164]==Result[1:164])
[1] 0.2195122
> #shows table of number of correctly and incorrectly predicted
> #wins and losses; model predicted 106 wins,89 of which were actually wins
> table(PredictedResult1718,Result[1:164])

PredictedResult1718 Loss Win
               Loss   39  19
               Win    17  89
```

This shows out of 164 games for the two seasons, logistic regression 1 correctly predicted 128 of them. The misclassification rate is 21.95%. The model predicted a total of 106 wins in both seasons combined in which 89 were actually wins and predicted a total of 58 losses in which 39 of them were actually losses.



Residual plot above for all 164 observations shows a random distribution of values with many of the residuals being close to zero.

Conclusion:

```
> spursLR5=glm(Result~REB+AST+ThreePtPerc+STL+TO+OFGPerc+O3PM+OFTPerc+O.AST-
spursdata,subset=1:82,family=binomial)
> summary(spursLR5)

Call:
glm(formula = Result ~ REB + AST + ThreePtPerc + STL + TO + OFGPerc +
    O3PM + OFTPerc + O.AST + Salary.Diff, family = binomial,
    data = spursdata, subset = 1:82)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.20248  -0.02399   0.06172   0.24040   1.32882

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  18.67321   10.82811   1.725  0.08462 .
REB           0.03316    0.07945   0.417  0.67636
AST           0.40854    0.14682   2.783  0.00539 **
ThreePtPerc  17.09797    8.45393   2.022  0.04313 *
STL           0.14758    0.20736   0.712  0.47664
TO           -0.02789    0.16887  -0.165  0.86881
OFGPerc     -48.02905   19.24481  -2.496  0.01257 *
O3PM         -0.24550    0.18512  -1.326  0.18478
OFTPerc      -8.93942    5.36891  -1.665  0.09591 .
O.AST        -0.23514    0.14288  -1.646  0.09981 .
Salary.Diff  -0.08504    0.05562  -1.529  0.12625
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 36.187  on 71  degrees of freedom
AIC: 58.187
```

```
> spursLR=glm(Result~FGA+REB+AST+O3PM+O.AST+Salary.Diff, data=spursdata, subset=1:82,family=binomial)
> summary(spursLR)

Call:
glm(formula = Result ~ FGA + REB + AST + O3PM + O.AST + Salary.Diff,
    family = binomial, data = spursdata, subset = 1:82)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4058  -0.1432   0.1005   0.3960   1.5425

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.04885    4.80031   0.427  0.66951
FGA         -0.22451    0.08133  -2.761  0.00577 **
REB          0.29797    0.10159   2.933  0.00336 **
AST          0.50737    0.13014   3.899 9.67e-05 ***
O3PM        -0.26518    0.13114  -2.022  0.04316 *
O.AST       -0.22610    0.10645  -2.124  0.03367 *
Salary.Diff -0.11330    0.04097  -2.766  0.00568 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 93.305  on 81  degrees of freedom
Residual deviance: 46.836  on 75  degrees of freedom
AIC: 60.836

Number of Fisher Scoring iterations: 7
```

Based off of the regressions used, regression 1 and 5 are the best at predicting games while also not having any issues with their residual plots. While logistic regression 1 did not predict the 2017-2018 season results as well as it did the 2016-2017 season results, it still predicted 57 out of 82 correctly which is pretty good considering the regression was modeled off of 2016-2017 season data. Additionally, there is a variation from season to season, in particular the 2017-2018 season for the Spurs when their best player Kawhi Leonard was injured for all but 9 games of the season. As a result, the impact of his absence would most likely affect a number of the predictors. It would also be very interesting to use these regressions to regress on future seasons while also seeing how the model for the Spurs would work for other NBA teams. Over time, other variables may become more explanatory than others. Perhaps, other teams are affected more by other variables so the model would need some changes.

R Code:

```
#Final Project
library(ISLR)
library(glmnet)
spursdata=read.csv("SpursStatsPerGame.csv")
dim(spursdata)
head(spursdata)

spurs1617data=spursdata[1:82,]

spurs1718data=spursdata[83:164,]

spursLR=glm(Result~FGA+REB+AST+O3PM+O.AST+Salary.Diff, data=spursdata,
subset=1:82,family=binomial)
summary(spursLR)
contrasts(Result)
#first seeing regression fit based off of 2016-2017 season on 2016-2017 season
games
MyPredictions1617=predict(spursLR, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 62 wins,56 of which were actually wins
table(PredictedResult1617,Result[1:82])

residualsLR=Win[1:82]-MyPredictions1617
```

```
plot(residualsLR)

y=Win[1:82]
modelpredict=MyPredictions1617[1:82]
mean((y-modelpredict)^2)

spursLR2=glm(Result~ThreePtPerc+O3PM,data=spursdata,subset=1:82,family=bin
omial)
summary(spursLR2)

#first seeing regression fit based off of 2016-2017 season on 2016-2017 season
games
MyPredictions1617=predict(spursLR2, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 72 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])
plot(residuals(spursLR2))
par(mfrow=c(2,2))

residualsLR2=Win[1:82]-MyPredictions1617
plot(residualsLR2)

y=Win[1:82]
modelpredict=MyPredictions1617[1:82]
mean((y-modelpredict)^2)


spursLR3=glm(Result~AST,data=spursdata,subset=1:82,family=binomial)
summary(spursLR3)

#first seeing regression fit based off of 2016-2017 season on 2016-2017 season
games
MyPredictions1617=predict(spursLR3, type="response")
```

```
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 72 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])

residualsLR3=Win[1:82]-MyPredictions1617
plot(residualsLR3)

y=Win[1:82]
modelpredict=MyPredictions1617[1:82]
mean((y-modelpredict)^2)

spursLR4=glm(Result~AST+Salary.Diff,
data=spursdata,subset=1:82,family=binomial)
summary(spursLR4)

#first seeing regression fit based off of 2016-2017 season on 2016-2017 season
games
MyPredictions1617=predict(spursLR4, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 70 wins,58 of which were actually wins
```

```
table(PredictedResult1617,Result[1:82])

residualsLR4=Win[1:82]-MyPredictions1617
plot(residualsLR4)

y=Win[1:82]
modelpredict=MyPredictions1617[1:82]
mean((y-modelpredict)^2)

spursLR5=glm(Result~REB+AST+ThreePtPerc+STL+TO+OFGPerc+O3PM+OFTPerc
+O.AST+Salary.Diff,data=spursdata,subset=1:82,family=binomial)
summary(spursLR5)

#first seeing regression fit based off of 2016-2017 season on 2016-2017 season
games
MyPredictions1617=predict(spursLR5, type="response")
#compare actual wins with MyPredictions
cbind(Result[1:82]=="Win",MyPredictions1617[1:82])
#makes an array of "Loss" for all 82 games
PredictedResult1617=array("Loss",dim(spurs1617data)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
PredictedResult1617[MyPredictions1617 >=0.5]="Win"
PredictedResult1617
#displays the count of games on which PredictedResult==Result
sum(PredictedResult1617==Result[1:82])
#displays percentage of 82 games that are predicted correctly
mean(PredictedResult1617==Result[1:82])
#displays misclassification rate
1-mean(PredictedResult1617==Result[1:82])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 61 wins,57 of which were actually wins
table(PredictedResult1617,Result[1:82])

residualsLR5=Win[1:82]-MyPredictions1617
plot(residualsLR5)

y=Win[1:82]
modelpredict=MyPredictions1617[1:82]
mean((y-modelpredict)^2)
#finding 2017-2018 predictions
MyPredictions1718=predict(spursLR, type="response",spursdata)
#compare actual wins with MyPredictions
cbind(Result[83:164]=="Win",MyPredictions1718[83:164])
#makes an array of "Loss" for all of the games from the 2017-2018 season
PredictedResult1718=array("Loss",dim(spursdata)[1])
#replaces Loss by Win for games which MyPredictions >= 0.5
```

```
PredictedResult1718[MyPredictions1718 >=0.5]="Win"
PredictedResult1718[83:164]
#displays the count of games which our prediction
#equals the actual outcome for the game for 2018 using 2017 regression
sum(PredictedResult1718[83:164]==Result[83:164])
#displays percentage of 82 games from 2017-2018 season that are predicted
correctly
mean(PredictedResult1718[83:164]==Result[83:164])
#displays misclassification rate
1-mean(PredictedResult1718[83:164]==Result[83:164])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 44 wins,33 of which were actually wins
table(PredictedResult1718[83:164],Result[83:164])

y2=Win[83:164]
modelpredict2=MyPredictions1718[83:164]
mean((y2-modelpredict2)^2)

residuals1718=Win[83:164]-MyPredictions1718[83:164]
plot(residuals1718)

#displays total count of games for both seasons
#using regression from 2017 season data
sum(PredictedResult1718==Result[1:164])
#displays percentage of 164 games from both seasons that are predicted correctly
mean(PredictedResult1718==Result[1:164])
#displays misclassification rate for both seasons
1-mean(PredictedResult1718[1:164]==Result[1:164])
#shows table of number of correctly and incorrectly predicted
#wins and losses; model predicted 106 wins,89 of which were actually wins
table(PredictedResult1718,Result[1:164])

totalresiduals=Win[1:164]-MyPredictions1718
plot(totalresiduals)
```