Marcus Crowder
LN 8.1
CIS 3920 Data-Ming

First off this was the hardest Learning Exercise in terms of time. I first decided to create a new data set to work on classification. I am a huge baseball fan so I picked Data from the 2013 MLB season(The last year my favorite the the Boston Red Sox won the World Series). It was also because that season was the last recorded season with stadium attendance and percentage of stadium attendance.

I initially labelled the Data set Baseball2011 by accident but it counts for the 2013 season. I tried to decide average percentage Fan attendance based on Percentage of stadium that was filled during the year. I used stats such as R.g(Runs per game),X2B(doubles),x3b(triples), hr(homeruns), BA(batting Average), Etc. If stadium attendance was above the average league attendance of 70% the team would have good attendance. I then cleaned up the Data to create a dataset of ->

```
> Baseball = read.csv("~/Downloads/Baseball2011.csv", header = TRUE)
> head(Baseball)
              TEAM  PCT  R.G X2B X3B  HR  SB  BB   SO    BA OPS.  ERA  FanAtt
1          Arizona 54.2 4.23 302  31 130  62 519 1142 0.259   96 3.92  BadAtt
2          Atlanta 63.3 4.25 247  21 181  64 542 1384 0.249   99 3.18  BadAtt
3        Baltimore 64.1 4.60 298  14 212  79 416 1125 0.260  101 4.20  BadAtt
4           Boston 94.4 5.27 363  29 178 123 581 1308 0.277  116 3.79 GoodAtt
5     Chicago Cubs 79.3 3.72 297  18 172  63 439 1230 0.238   89 4.00 GoodAtt
6 Chicago White Sox 54.4 3.69 237  19 148 105 411 1207 0.249   84 3.98  BadAtt
```

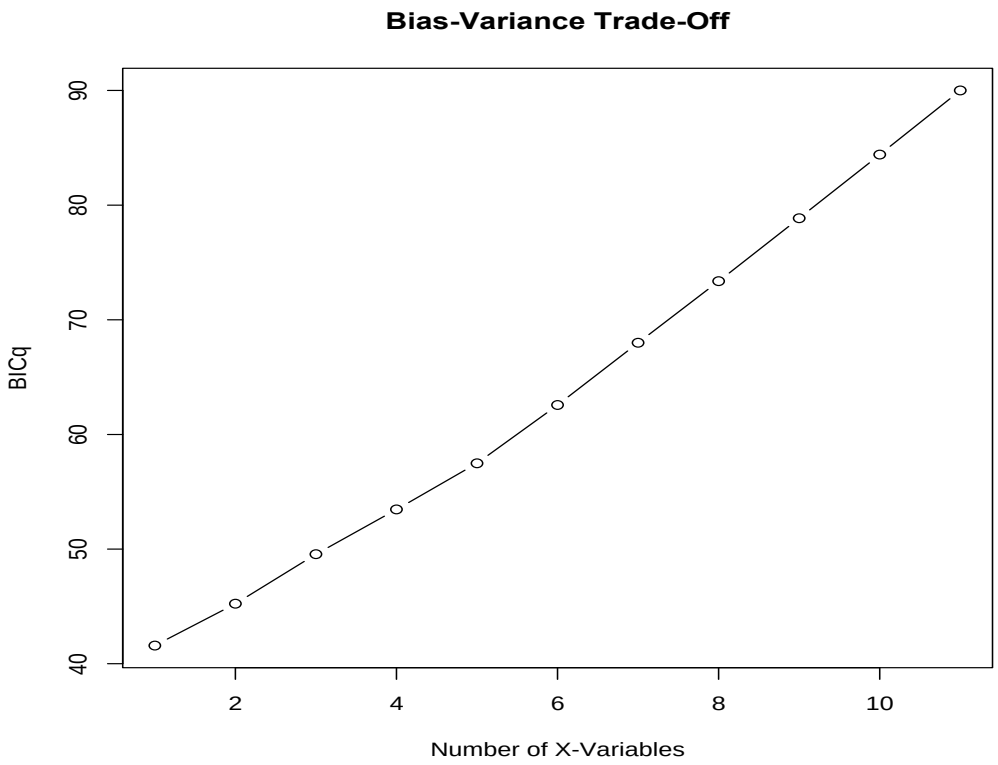
Which in theory sounds fun to try and predict attendance based on hitting with one pitching stat ERA but it didn't pan out as expected.

```
> Baseball2
    R.G X2B X3B  HR  SB  BB   SO    BA OPS.  ERA  FanAtt
1  4.23 302  31 130  62 519 1142 0.259   96 3.92  BadAtt
2  4.25 247  21 181  64 542 1384 0.249   99 3.18  BadAtt
3  4.60 298  14 212  79 416 1125 0.260  101 4.20  BadAtt
4  5.27 363  29 178 123 581 1308 0.277  116 3.79 GoodAtt
5  3.72 297  18 172  63 439 1230 0.238   89 4.00 GoodAtt
6  3.69 237  19 148 105 411 1207 0.249   84 3.98  BadAtt
7  4.31 274  20 155  67 585 1245 0.249   98 3.38 GoodAtt
8  4.60 290  23 171 117 562 1283 0.255  107 3.82  BadAtt
9  4.36 283  36 159 112 427 1204 0.270   91 4.44  BadAtt
10 4.91 292  23 176  35 531 1073 0.283  111 3.61 GoodAtt
11 3.77 266  16 148 110 426 1535 0.240   86 4.79  BadAtt
12 4.00 254  34 112 153 422 1048 0.260   89 3.45  BadAtt
13 4.52 270  39 164  82 523 1221 0.264  110 4.23 GoodAtt
14 4.01 281  17 138  78 476 1146 0.264  103 3.25 GoodAtt
15 3.17 219  31  95  78 432 1232 0.231   73 3.71  BadAtt
16 3.95 238  43 157 142 407 1183 0.252   93 3.84 GoodAtt
17 3.79 285  15 151  52 533 1430 0.242   90 4.55 GoodAtt
18 3.82 263  32 130 114 512 1384 0.237   91 3.77  BadAtt
19 4.01 247  24 144 115 466 1214 0.242   88 3.94 GoodAtt
20 4.73 301  25 186  74 573 1178 0.254  107 3.56  BadAtt
21 3.77 255  32 140  73 417 1205 0.248   91 4.32 GoodAtt
22 3.91 273  35 161  94 469 1330 0.245  100 3.26 GoodAtt
23 3.81 246  26 146 118 467 1309 0.245   97 3.98  BadAtt
24 3.85 249  17 188  49 529 1353 0.237   98 4.31  BadAtt
25 3.88 280  35 107  67 469 1078 0.260  100 4.00 GoodAtt
26 4.83 322  20 125  45 481 1110 0.269  102 3.42 GoodAtt
27 4.29 296  23 165  73 589 1171 0.257  106 3.74  BadAtt
28 4.48 262  23 176 149 462 1067 0.262  100 3.62 GoodAtt
29 4.40 273  24 185 112 510 1123 0.252   99 4.25  BadAtt
30 4.05 259  27 161  88 464 1192 0.251   94 3.59 GoodAtt
```

The Bestglm algorithm only worked for BA when with the baseball data without t=1 when using t=1 everything was selected as an Xvariable
And the graph of the bias Variance trade off ended up just increasing with the best fit being only 1.

```
> out=bestglm(Baseball.new,IC="BICq",family=binomial)
Morgan-Tatar search since family is non-gaussian.
> out
BICq(q = 0.25)
BICq equivalent for q in (0, 0.674109237390973)
Best Model:
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -11.18164   8.383732 -1.33373 0.1822922
BA           44.14984  33.092108  1.33415 0.1821546
> out=bestglm(Baseball.new,IC="BICq",t=1,family=binomial)
Note: in this special case with BICq with t = 1 only fitted model is returned.
With t=1, full model is fitted.
> out
BICq(q = 1)
Best Model:
               Estimate   Std. Error     z value  Pr(>|z|)
(Intercept)   4.413951951 2.129878e+01  0.20723961 0.8358227
R.G          -0.364612185 3.877456e+00 -0.09403388 0.9250822
X2B           0.014790764 2.562766e-02  0.57714076 0.5638444
X3B           0.025728774 7.618473e-02  0.33771562 0.7355775
HR           -0.020296765 3.528986e-02 -0.57514442 0.5651936
SB           -0.019185140 1.809753e-02 -1.06009733 0.2891003
BB           -0.021125121 1.583987e-02 -1.33366750 0.1823128
SO            0.001118331 5.356235e-03  0.20879044 0.8346118
BA          -18.380932742 1.029760e+02 -0.17849731 0.8583324
OPS.          0.159816301 1.286057e-01  1.24268488 0.2139840
ERA          -1.182666857 1.394725e+00 -0.84795694 0.3964620
> out$Subsets$BICq
Error in out$Subsets$BICq : $ operator is invalid for atomic vectors
> out=bestglm(Baseball.new,IC="BICq",family=binomial, TopModels=1)
Morgan-Tatar search since family is non-gaussian.
> out$Subsets$BICq
 [1] 41.58883 45.23972 49.55436 53.46256 57.48088 62.56891 67.99867 73.36569 78.85798 84.41140 90.00103
> out
BICq(q = 0.25)
BICq equivalent for q in (0, 0.674109237390973)
Best Model:
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -11.18164   8.383732 -1.33373 0.1822922
BA           44.14984  33.092108  1.33415 0.1821546
```

Bias-Variance Trade-Off
Of Baseball data to predict average attendance. This made me Decide to alter my dataset.



For my new Dataset I decided to actually name the data by its Name with Baseball2013. I decided to base the classifier on if the team finished with a record of .500 or above. And I decided to use more pitching stats with a stat for Errors(E). These stats are for all 30 baseball teams.

```
> Baseball2013 = read.csv("~/Downloads/Baseball2013.csv", header = TRUE)
> head(Baseball2013)
    R  HR RBI   SO    BA   OBP   SLG OPS.  ERA  WHIP  ER   E Season
1 685 130 647 1142 0.259 0.323 0.391   96 3.92 1.301 651  75   Good
2 688 181 656 1384 0.249 0.321 0.402   99 3.18 1.196 512  85   Good
3 745 212 719 1125 0.260 0.313 0.431  101 4.20 1.315 678  54   Good
4 853 178 819 1308 0.277 0.349 0.446  116 3.79 1.300 613  80   Good
5 602 172 576 1230 0.238 0.300 0.392   89 4.00 1.293 643 100    Bad
6 598 148 574 1207 0.249 0.302 0.378   84 3.98 1.329 643 121    Bad
> Baseball = Baseball2013
> Baseball$Season = c(Baseball2013$Season)
> head(Baseball$Season)
[1] 2 2 2 2 1 1
> Baseball$Season[Baseball$Season==1]<-0
> Baseball$Season[Baseball$Season==2]<-1
> head(Baseball$Season)
[1] 1 1 1 1 0 0
> head(Baseball)
    R  HR RBI   SO    BA   OBP   SLG OPS.  ERA  WHIP  ER   E Season
1 685 130 647 1142 0.259 0.323 0.391   96 3.92 1.301 651  75      1
2 688 181 656 1384 0.249 0.321 0.402   99 3.18 1.196 512  85      1
3 745 212 719 1125 0.260 0.313 0.431  101 4.20 1.315 678  54      1
4 853 178 819 1308 0.277 0.349 0.446  116 3.79 1.300 613  80      1
5 602 172 576 1230 0.238 0.300 0.392   89 4.00 1.293 643 100      0
6 598 148 574 1207 0.249 0.302 0.378   84 3.98 1.329 643 121      0
```

R(Runs), HR(Homeruns), RBI(Runs batted In), SO(Strikeouts by team batters), BA(Batting Average), OBP(On Base Percentage), SLG(Slugging %), ERA(Earned run average by team pitchers), WHIP(Walks and Hits per innings pitched Pitcher), ER(Earned runs by other team) , E(Error).

I created a duplicate of my baseball data then Changed Season from Good and Bad as a classifier to 1s and 0s with the replace method in R as shown:

```
> Baseball = Baseball2013
> Baseball$Season = c(Baseball2013$Season)
> head(Baseball$Season)
[1] 2 2 2 2 1 1
> Baseball$Season[Baseball$Season==1]<-0
> Baseball$Season[Baseball$Season==2]<-1
> head(Baseball$Season)
[1] 1 1 1 1 0 0
```

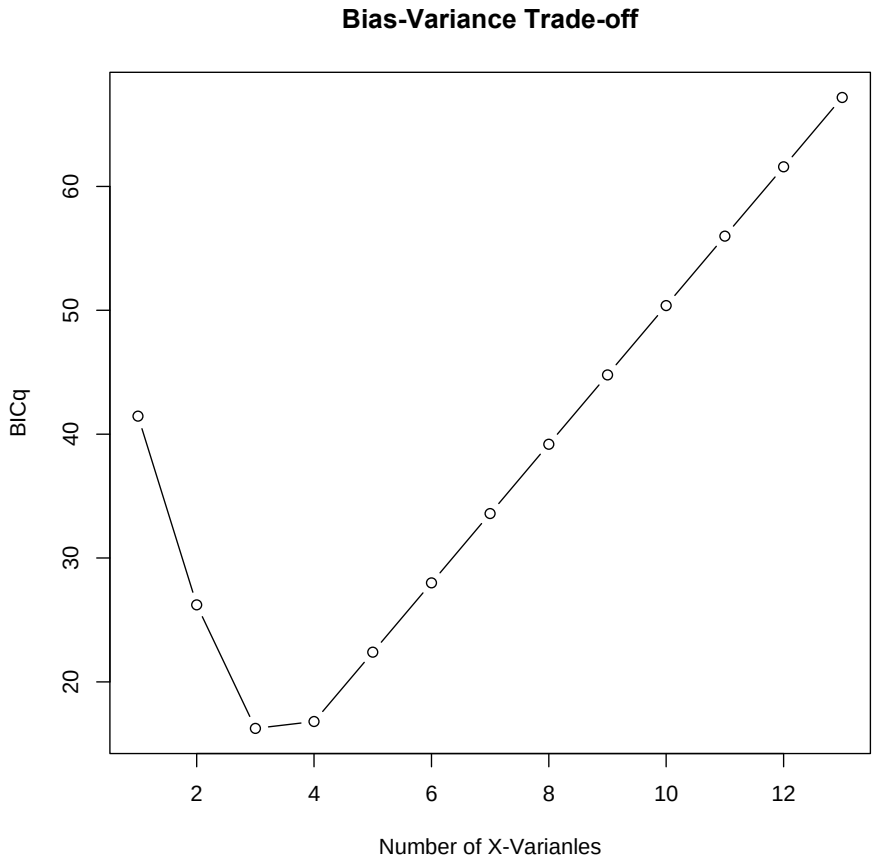
I then used the bestglm function on my new dataset Baseball.

```
> out=bestglm(Baseball,IC="BICq",family=binomial, TopModels=1)
Morgan-Tatar search since family is non-gaussian.
There were 50 or more warnings (use warnings() to see the first 50)
> out
BICq(q = 0.25)
BICq equivalent for q in (0.00226905182845594, 0.304836063521138)
Best Model:
               Estimate   Std. Error   z value  Pr(>|z|)
(Intercept)  164.7210503 140.02282905  1.176387 0.2394402
R              0.1122553   0.09379856  1.196770 0.2313961
WHIP        -183.0948870 149.86962224 -1.221694 0.2218232
> out$Subsets$BICq
 [1] 41.45540 26.22181 16.24679 16.79527 22.39369 27.99211 33.59053 39.18895 44.78738 50.38580 55.98422 61.58264 67.18106
```

Out=bestglm(Baseball,IC="BICq",family=binomial, TopModels=1)

The Xvariables chosen was R(runs scored by team) and WHIP(Walks and Runs per inning) this makes a lot of sense but I would of liked a lower PR but this will do for now. There were also some warnings I ignored but because it worked it couldn't be too bad (yet).

I then used the plot function to create a plot of the Bias-Variance Tradeoff according to number of xvariables the best model appears to be at 3 Xvariables (I guess if intercept was counted). The second was at 4 variables followed by an inclining trend.



plot(out$Subsets$BICq,type="b",xlab="Number of X-Varianles",ylab="BICq", main="Bias-Variance Trade-off")

The last thing left to do was fitting this model on a plot with TopModels = 4096 or 2^12 for the number of Xvariables used.

```
> GLM.fit = bestglm(Baseball, IC="BICq",family=binomial,TopModels=4096)
Morgan-Tatar search since family is non-gaussian.
There were 50 or more warnings (use warnings() to see the first 50)
> y=GLM.fit$BestModels
> x=apply(y[1:12],1,sum)
> plot(x,y[,13],xlab="# of Regressors",ylab="BICq")
```

> GLM.fit = bestglm(Baseball, IC="BICq",family=binomial,TopModels=4096)
this function was used to select all the models
then y was sekected ti be the list of values in the best models.
X was created with the apply function since y just held TRUE and False Values the goal was to sum the True and Falses statements in order to fit them on a plot.

The BICq appears to be lowest at 3 as mentioned before with a lot of points collapsing on top of each other. Which could mean that some predictors had no actual impact on other predictors but again this is 4096 points so this was expected. It's beautiful to see baseball data plotted like this.