Chapter 4

Tong Sun

1/27/2022

##4.6 Suppose we collect data for a group of students in a statistics class with variables X1 = hours studied, X2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, β_0 = -6, β_1 = 0.05, β_2 = 1.

###(a) Estimate the probability that a student who studies for 40h and has an undergrad GPA of 3.5 gets an A in the class.

Here we have $X_1 = 40$, $X_2 = 3.5$ and we have the equation for predicted probability is $Y = -6+0.05*X_1+X_2$. Plugging the predictors' values in the equation we get:

$P(X) = \frac{e^{-6+0.05*X_1+X_2}}{1+e^{-6+0.05*X_1+X_2}} = 0.3775$

So the probability that a student who studies for 40h and has an undergrad GPA of 3.5 gets an A in the class was 0.3775.

###(b) How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class?

The equation for predicted probability tells us that:

$$\frac{e^{-6+0.05*X_1+3.5}}{1+e^{-6+0.05*X_1+3.5}} = 0.5$$

which means:

$$e^{-6+0.05*X_1+3.5} = 1$$

After taking the logarithm of both sides, we have:

$$X_1 = \frac{2.5}{0.05} = 50$$

We can make the conclusion that the student in part (a) need to study 50 hours in order to have a 50% chance of getting an A in the class.

##4.8 Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures.First we use logistic regression and get an error rate of 20% on the training data and 30% on the test data. Next we use 1-nearest neighbors and get an average error rate (averaged over both test and training data sets) of 18%. Based on these results, which method should we prefer to use for classification of new observations? Why ?

We have: Logistic Regression: 20% training error rate, 30% test error rate KNN(K=1): 18% average error rate

The nearest neighbor of any training observation should be the observation itself, so for KNN with K=1, the training error rate is zero.In this situation, the test error rate will be 36% in order to make the average error rate is 18%. As a result, I will choose logistic regression so that the test error rate will be lower.

##4.9 This problem has to do with odds. ###(a) On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

We have the equation:

$$\frac{P(X)}{1-P(X)} = 0.37P(X) = \frac{0.37}{1+0.37} = 0.27$$

We can make the conclusion: on average, a fraction of 27% of people defaulting on their credit card payment.

###(b) Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default?

We have $P(X) = 0.16$, so:

$$\frac{P(X)}{1-P(X)} = \frac{0.16}{1-0.16} = 0.19$$

The odds that she will default is 19% if she has a 16% chance of defaulting on her credit card payment.

##4.13

This question should be answered using the Weekly data set, which is part of the ISLR2 package.This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

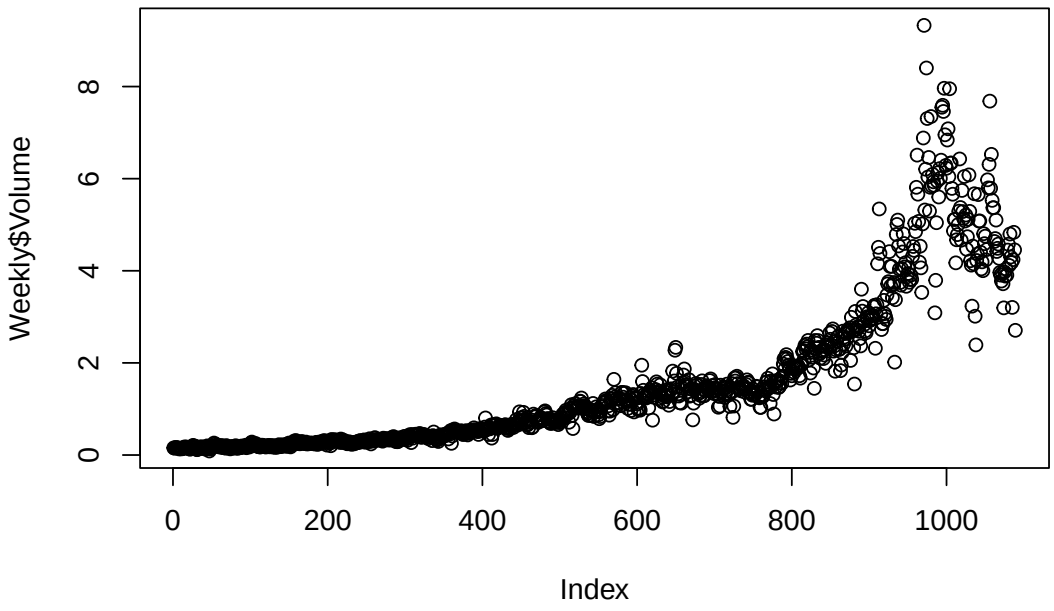
###(a) Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
## Warning: package 'ISLR2' was built under R version 4.1.2
```

```
##       Year           Lag1                Lag2                Lag3          
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
##       Lag4                Lag5                Volume            Today          
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
##  Direction 
##  Down:484  
##  Up  :605  
##            
##            
##            
##
```

```
##                Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
```

```
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```



Looking at the correlations table, I find the correlations between "lag" variables and "today" are almost zero, which means there are not too much significant relationship between these variables. But the correlation between "Volume" and "Year" is obvious, 0.84.When I plot "Volume", I find it is the fact that "Volume" increase over time.

###(b) Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

From the results showed above, I find that only the p-value of predictor "Lag2" is 0.0296, less than 0.05. So "Lag2" is the only predictor statistically significant.

###(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
##
## pred.fit1 Down  Up
##      Down   54  48
##      Up    430 557
```

Given the predictions, the last command produces a confusion matrix in order to determine how many observations were correctly or incorrectly classified. The diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions. Hence the model correctly predicted that the market would go up on 557 days and it would go down on 54 days, for a total of $557 + 54 = 611$ correct predictions. In this case, logistic regression correctly predicted the movement of the market $\frac{557}{557+54} = 91.16\%$, when the market goes up. For weeks when the market goes down, the model is right only $\frac{54}{557+54} = 8.84\%$ of the time. And also, the total number of observations is $54 + 48 + 430 + 557 = 1089$, we may conclude that th percentage of correct predictions on the training data is $\frac{54+557}{1089} = 56.11\%$. which also means 43.89% is the training error rate, which is often overly optimistic.

###(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
##
## Call:
## glm(formula = Weekly$Direction ~ Lag2, family = binomial, data = Weekly,
##     subset = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
##          direction.2009
## pred.fit2 Down Up
##      Down    9  5
##      Up     34 56
```

In this situation, the total number of observations is $9 + 5 + 34 + 56 = 104$. So we can conclude that the percentage of correct predictions on the test data is $\frac{9+56}{104} = 62.5\%$, which equals to a 37.5% test error rate. The model correctly predicted that the market would go up on 56 days and it would go down on 9 days, for a total of $56 + 9 = 65$ correct predictions. Also, the confusion matrix shows that on days when logistic regression predicts an increase in the market, it has a $\frac{56}{34+56} = 62.2\%$ accuracy rate. This suggests a possible trading strategy of buying on days when the model predicts an increase market, and avoiding trades on days when a decrease is predicted. We could also say that for weeks when the market goes up, the model is right $\frac{56}{56+5} = 91.80\%$ of the time. For weeks when the market goes down, the model is right only $\frac{9}{9+34} = 20.93\%$ of the time.

###(e) Repeat (d) using LDA.

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
## Call:
## lda(Weekly$Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```
##       direction.2009
##        Down Up
##   Down    9  5
##   Up     34 56
```

In this case, we may conclude that the percentage of correct predictions on the test data is 62.5%. In other words 37.5% is the test error rate. We could also say that for weeks when the market goes up, the model is right 91.80% of the time. For weeks when the market goes down, the model is right only 20.93% of the time. These results are very close to those obtained with the logistic regression model which is not surprising.

###(f) Repeat (d) using QDA.

```
## Call:
## qda(Weekly$Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
##       direction.2009
##        Down Up
##   Down    0  0
##   Up     43 61
```

In this case, the total number of observations is $0+0+43+61 = 104$, so we may conclude that the percentage of correct prediction the test data is $\frac{61}{104} = 58.65\%$. In other words 41.35% is the test error rate. We could also say that for weeks when the market goes up, the model is right 100% of the time. For weeks when the market goes down, the model is right 0% of the time. Also, the confusion matrix shows that on days when QDA predicts an increase in the market, it achieves a correctness of 58.65%. This suggests that the quadratic form assumed by QDA may capture the true relationship more accurately than the linear forms assumed by LDA and logistic regression.

###(g) Repeat (d) using KNN with K = 1.

```
## Warning: package 'class' was built under R version 4.1.2
```

```
##         direction.2009
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

From the table above, we find that the total number of observations is $21 + 30 + 22 + 31 = 104$, so we may conclude that the percentage of correct predictions on the test data is $\frac{21+31}{104} = 50\%$. In other words 50% is the test error rate. We could also say that for weeks when the market goes up, the model is right 50.82% of the time. For weeks when the market goes down, the model is right only 48.84% of the time.

###(h) Repeat (d) using naive Bayes.

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##      Down        Up
## 0.4477157 0.5522843
##
## Conditional probabilities:
##       Lag2
## Y             [,1]     [,2]
##   Down -0.03568254 2.199504
##   Up    0.26036581 2.317485
```

The output contains the estimated mean and standard deviation for each variable in each class. The mean for "Lag2" is -0.0357 for "Direction=Down", and the standard deviation is 2.1995.

```
##         direction.2009
## nb.class Down Up
##     Down    0  0
##     Up     43 61
```

```
## [1] 0.5865385
```

Naive Bayes performs very well on this data, with accurate predictions over 58% of the time. This is better than LDA, but similar as QDA.

###(i) Which of these methods appears to provide the best results on this data?

If we compare the test error rates, we see that logistic regression and LDA have the minimum error rates, followed by QDA, Naive Bayes and KNN.

###(j) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

```
##         direction.2009
## glm.pred Down Up
##     Down    1  1
##     Up     42 60
```

```
## [1] 0.5865385
```

```
## [1] 0.5769231
```

```
##          direction.2009
## qda.class Down Up
##      Down   12 13
##      Up     31 48
```

```
## [1] 0.5769231
```

```
##          direction.2009
## knn.pred1 Down Up
##      Down   17 18
##      Up     26 43
```

```
## [1] 0.5769231
```

```
##          direction.2009
## knn.pred2 Down Up
##      Down    9 12
##      Up     34 49
```

```
## [1] 0.5576923
```

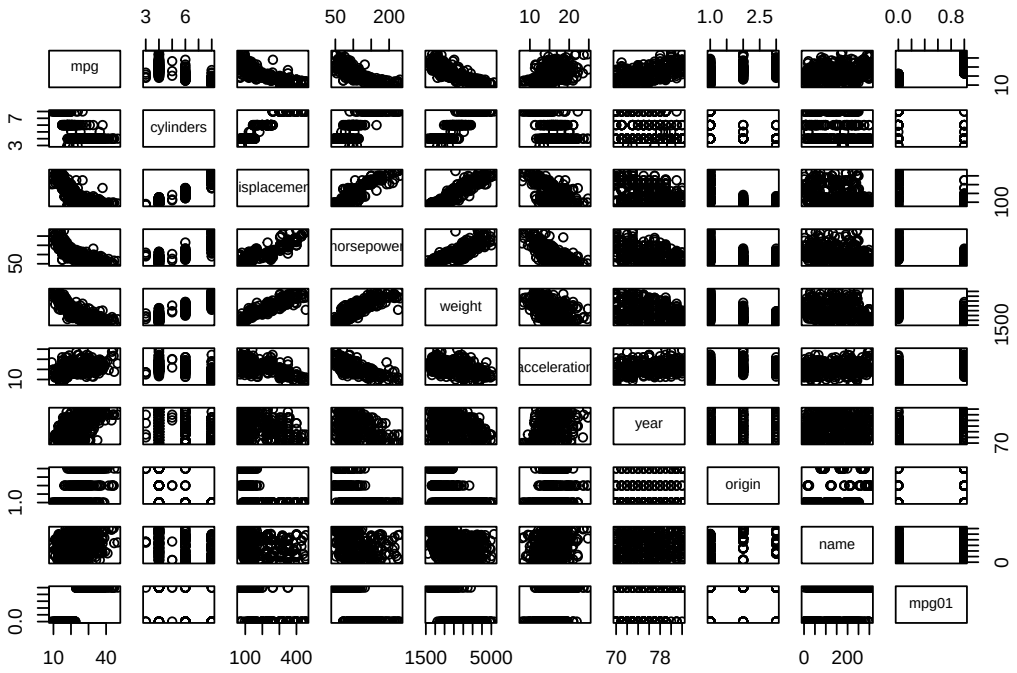
From all of these pemutations above, the original LDA and logistic regression have better performance in terms of their test error rates.

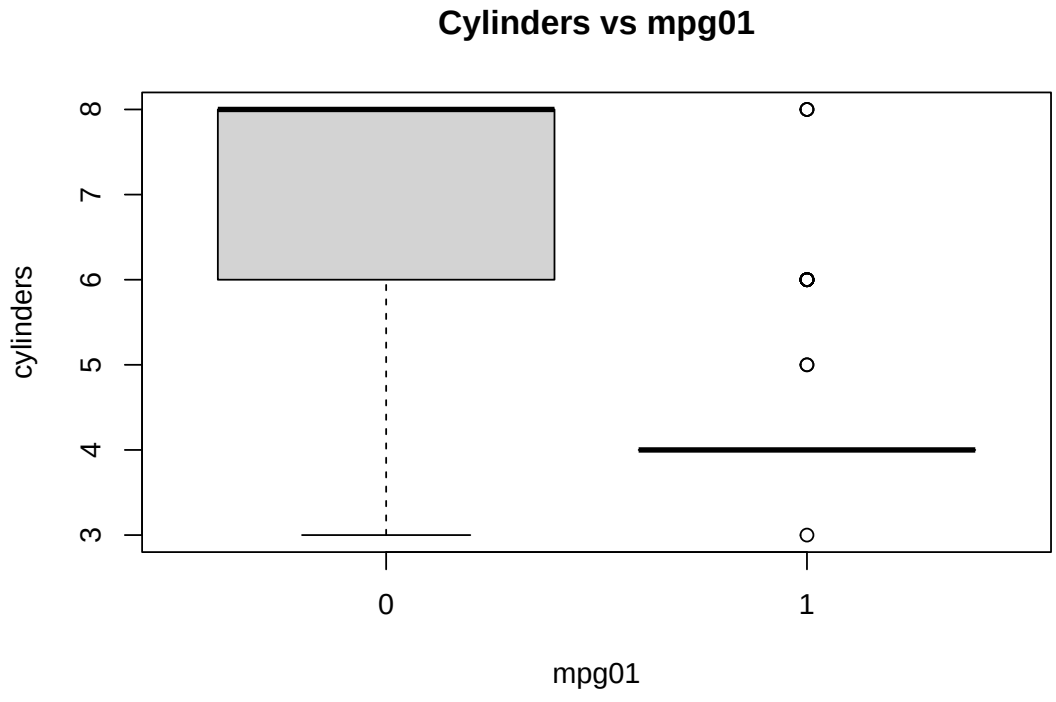
##4.14 In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the "Auto" data set.

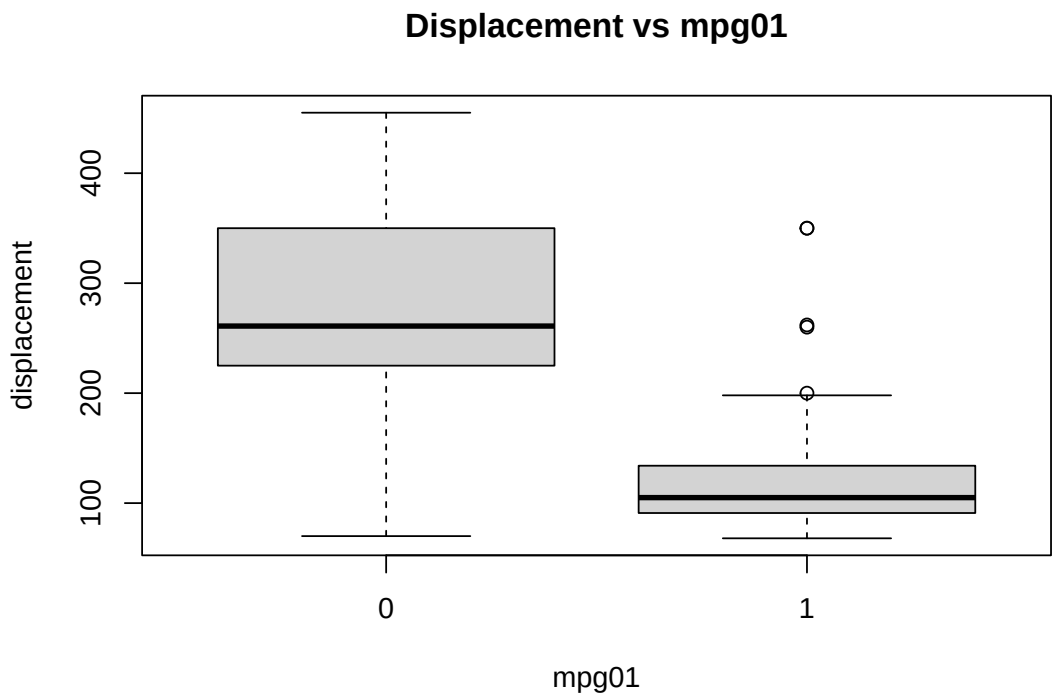
###(a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a 0 value below its median. You can compute the median using the median function. Note you may find it helpful to use the data.frame function to create a single data set containing both mpg01 and the other Auto variables.

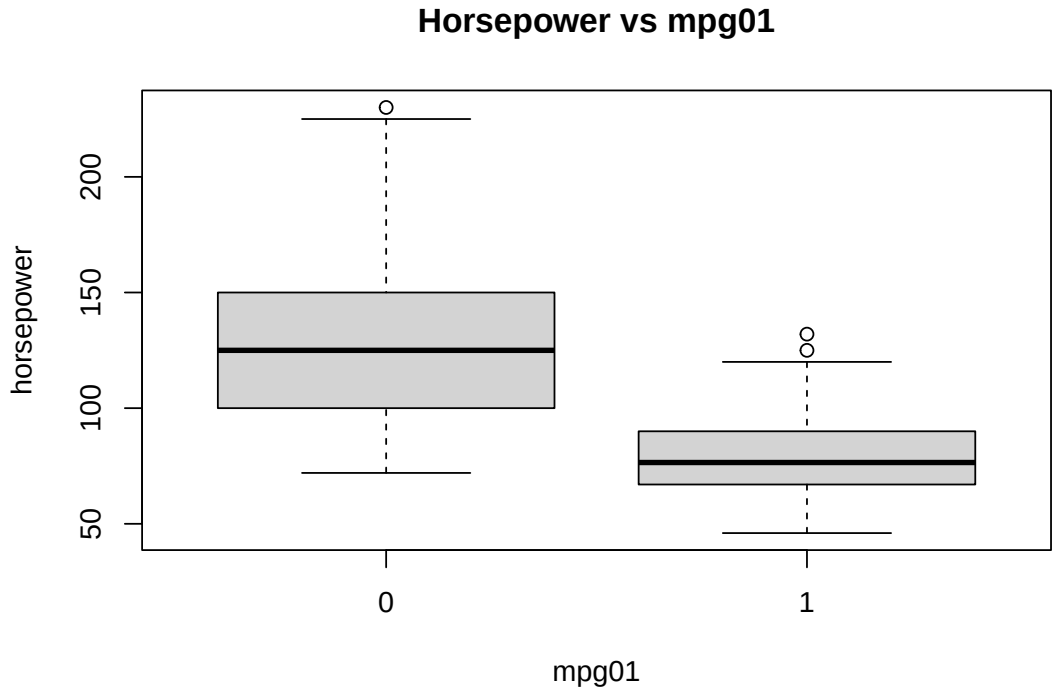
###(b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

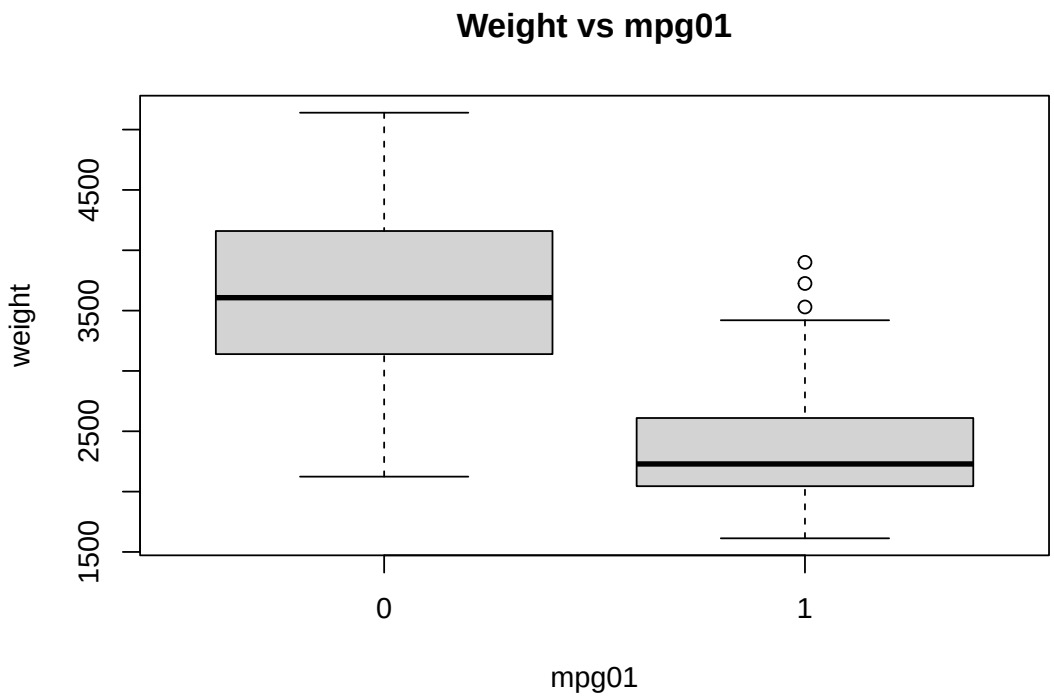
```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
## mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566
##              acceleration       year     origin      mpg01
## mpg             0.4233285  0.5805410  0.5652088  0.8369392
## cylinders      -0.5046834 -0.3456474 -0.5689316 -0.7591939
## displacement   -0.5438005 -0.3698552 -0.6145351 -0.7534766
## horsepower     -0.6891955 -0.4163615 -0.4551715 -0.6670526
## weight         -0.4168392 -0.3091199 -0.5850054 -0.7577566
## acceleration    1.0000000  0.2903161  0.2127458  0.3468215
## year            0.2903161  1.0000000  0.1815277  0.4299042
## origin          0.2127458  0.1815277  1.0000000  0.5136984
## mpg01           0.3468215  0.4299042  0.5136984  1.0000000
```

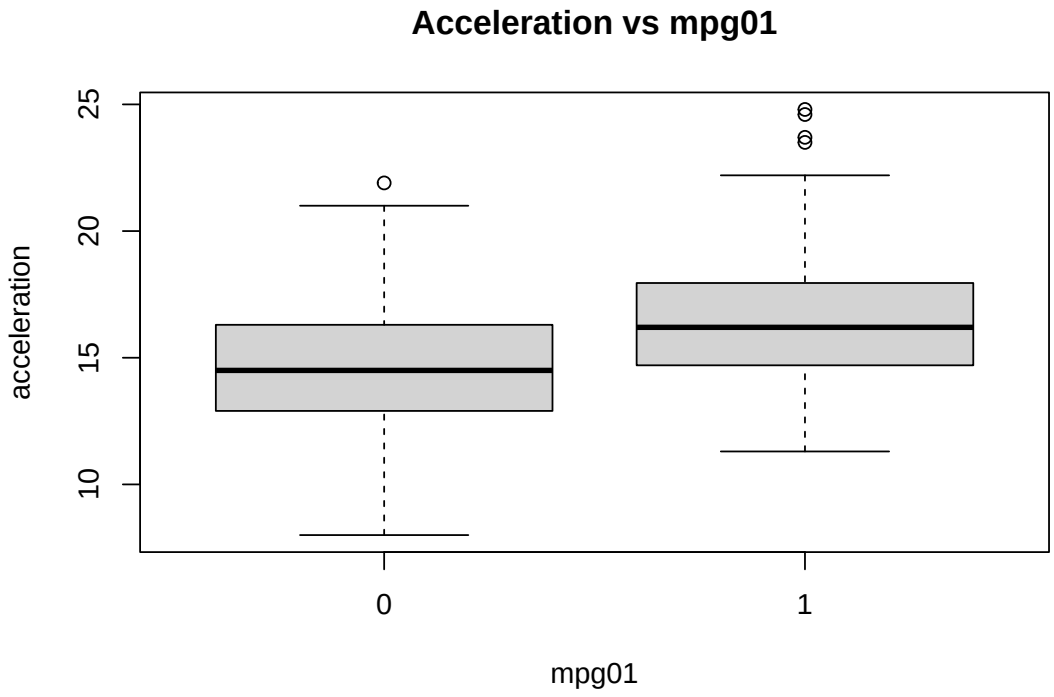

mpg
cylinders
isplacemen
horsepower
weight
acceleration
year
origin
name
mpg01

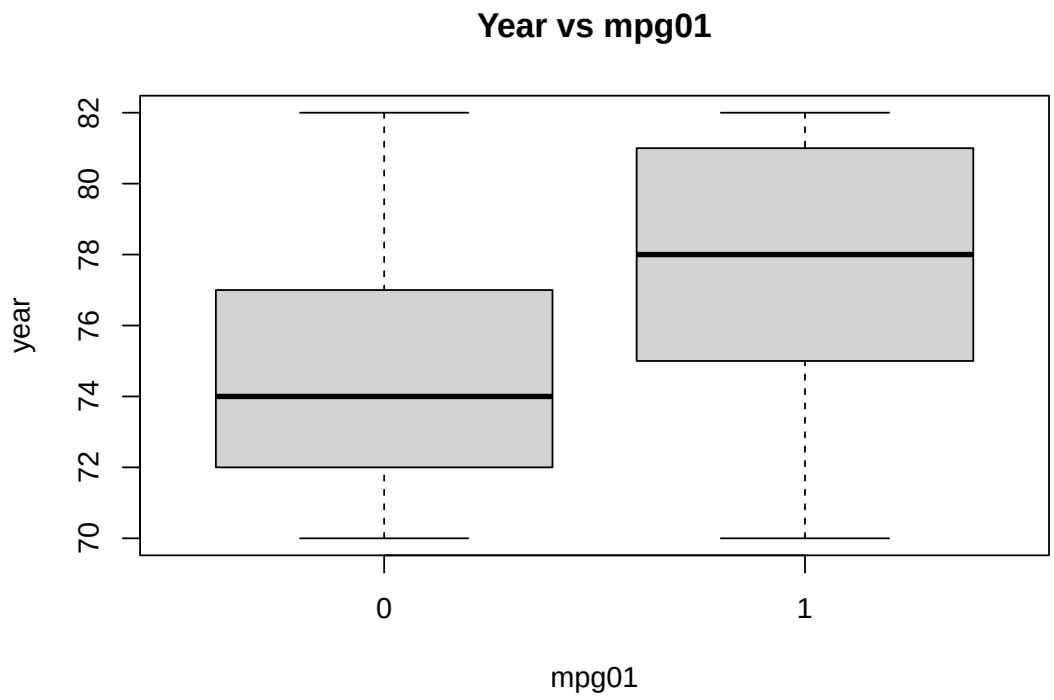

Cylinders vs mpg01
cylinders
mpg01
0
1


Displacement vs mpg01
displacement
100
200
300
400
0
1
mpg01


Horsepower vs mpg01
horsepower
50
100
150
200
0
1
mpg01


Weight vs mpg01
weight
1500
2500
3500
4500
0
1
mpg01


Acceleration vs mpg01
acceleration
10
15
20
25
0
1
mpg01



We may conclude that there exists some association between "mpg01" and "cylinders", "weight", "displacement" and "horsepower".

###(c) Split the data into a training set and a test set.

###(d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
##     subset = train)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
##
## Coefficients of linear discriminants:
##                         LD1
## cylinders    -0.6741402638
## weight       -0.0011465750
## displacement  0.0004481325
## horsepower    0.0059035377

##    mpg01.test
```

```
##      0  1
##   0 86  9
##   1 14 73
```

```
## [1] 0.1263736
```

From the table, we find that the total number of observations is $86 + 9 + 14 + 73 = 182$, so the test error rate should be $\frac{14+9}{182} = 12.637\%$, which can be confirmed by mean function.

###(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
## Call:
## qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
##     subset = train)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
```

```
##    mpg01.test
##      0  1
##   0 89 13
##   1 11 69
```

```
## [1] 0.1318681
```

The same as LDA, here we may conclude that we have a test error rate of 13.187%.

###(f) Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
##     family = binomial, data = Auto, subset = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48027  -0.03413   0.10583   0.29634   2.57584
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
## cylinders    -1.028032   0.653607  -1.573   0.1158
## weight       -0.002922   0.001137  -2.569   0.0102 *
## displacement  0.002462   0.015030   0.164   0.8699
## horsepower   -0.050611   0.025209  -2.008   0.0447 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.58  on 209  degrees of freedom
## Residual deviance:  83.24  on 205  degrees of freedom
## AIC: 93.24
##
## Number of Fisher Scoring iterations: 7

##         mpg01.test
## pred.glm  0  1
##        0 89 11
##        1 11 71
```

The number of total observations here is 89+11+11+71 = 182, so the test error should be $\frac{11+11}{182} = 12.0879\%$.

###(g) Perform naive Bayes on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.4571429 0.5428571
##
## Conditional probabilities:
##    cylinders
## Y       [,1]      [,2]
##   0 6.812500 1.4165377
##   1 4.070175 0.3928408
##
##    weight
## Y       [,1]     [,2]
##   0 3604.823 624.9159
##   1 2314.763 334.7228
##
##    displacement
## Y       [,1]     [,2]
##   0 271.7396 89.15194
##   1 111.6623 28.27696
##
##    horsepower
## Y        [,1]     [,2]
##   0 133.14583 38.49319
##   1  77.92105 15.19731

##         mpg01.test
```

```
## nb.probs  0  1
##        0 88 11
##        1 12 71
```

```
## [1] 0.8736264
```

Naive Bayes performs very well on this data, with accurate predictions about 87.36% of this time.

###(h) Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

```
##         mpg01.test
## pred.knn  0  1
##        0 83 11
##        1 17 71
```

```
## [1] 0.1538462
```

Here we may conclude that we have a test error rate of 15.3846% for K = 1.

```
##         mpg01.test
## pred.knn  0  1
##        0 77  7
##        1 23 75
```

```
## [1] 0.1648352
```

```
##         mpg01.test
## pred.knn  0  1
##        0 81  7
##        1 19 75
```

```
## [1] 0.1428571
```

We do the same thing on K=10 and K=100, and we get the conclusions that for K=10, the test error rate is 16.4835%, for K=100, the test error rate is 14.2857%. Therefore, the K value of 100 seems to perform better than 1 and 10.

##4.15 This problem involves writing functions. ###(a) Write a function, Power(), that prints out the result of raising 2 to the 3rd power. In other words, your function should compute 2^3 and print out the results. Hint: Recall that x^a raises x to the power a. Use the print() function to output the result.

```
## [1] 8
```

###(b) Create a new function, Power2(), that allows you to pass any two numbers, x and a, and prints out the value of x^a.

```
## [1] 6561
```

###(c) Using the Power2() function that you just wrote, compute 10^3, 8^17, and 131^3.

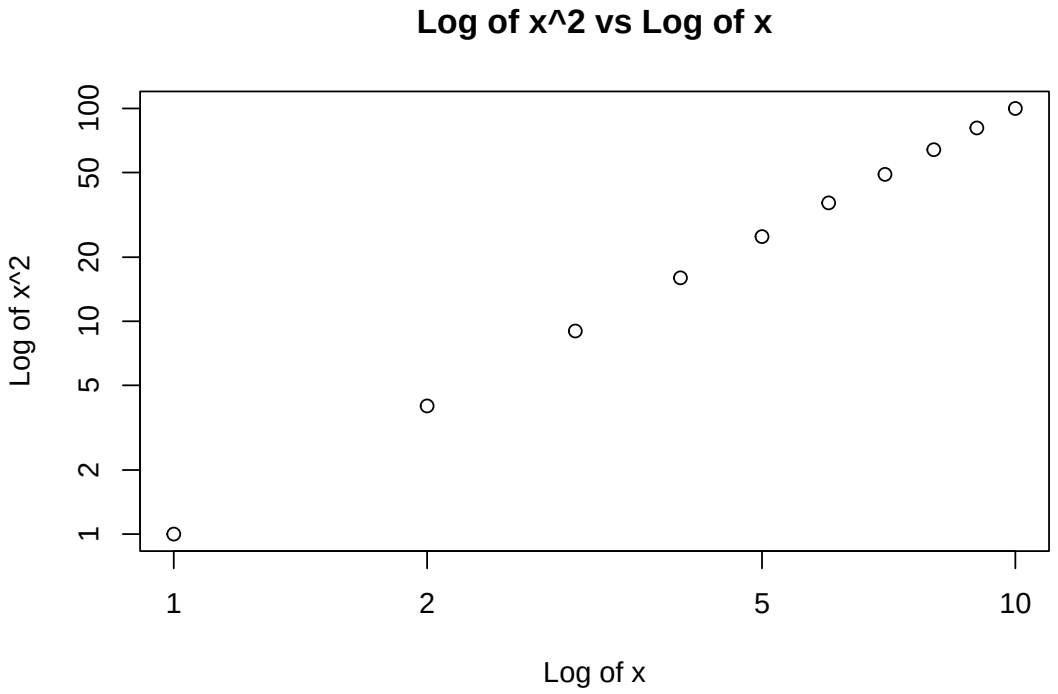
```
## [1] 1000
```

```
## [1] 2.2518e+15
```

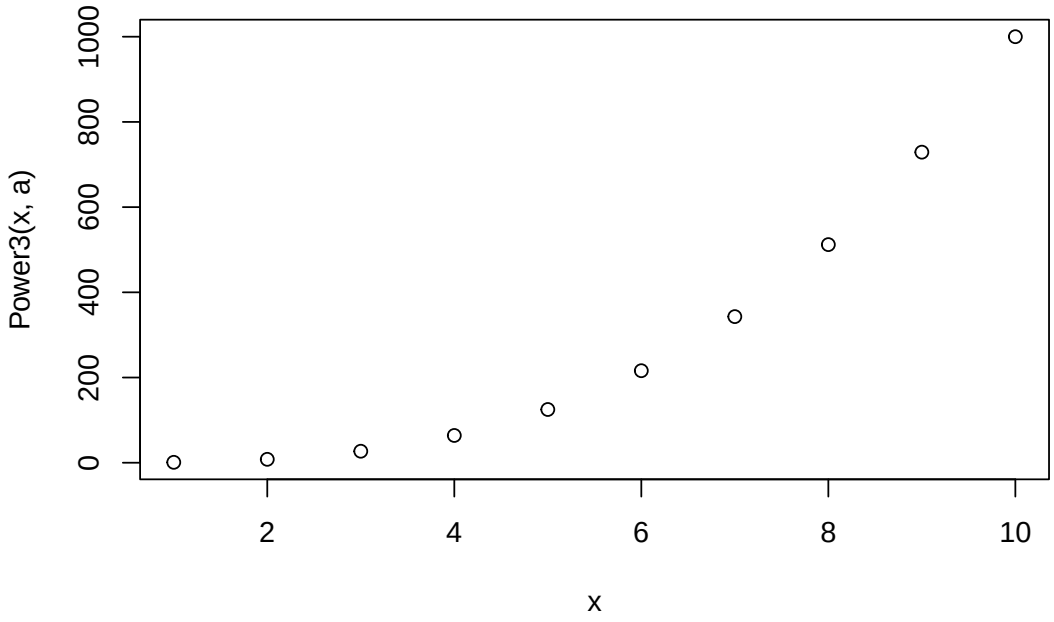
```
## [1] 2248091
```

###(d) Now create a new function, Power3(), that actually returns the result x^a as an R object, rather than simply printing it to the screen. That is, if you store the value x^a in an object called result within your function, then you can simply return() this result.

###(e) Now using the Power3() function, create a plot of f(x) = x^2. The x-axis should display a range of integers from 1 to 10, and the y-axis should display x^2. Label the axes appropriately, and use an appropriate title for the figure. Consider displaying either the x-axis, the y-axis, or both on the log-scale. You can do this by using log = "x",log = "y",or log = "xy" as arguments to the plot() function.



###(f) Create a function, PlotPower(), that allows you to create a plot of x against x^a for a fixed a and for a range of values of x.



##4.16 Using the Boston data set, fit classification models in order to predict whether a given census tract has a crime rate above or below the median. Explore logistic regression, LDA, naive Bayes, and KNN models using various subsets of the predictors. Describe your findings. Hint: You will have to create the response variable yourself, using the variables that are contained in the Boston data set.

Logistic Regression

Here I need to create our train and test data sets.

```
## The following objects are masked _by_ .GlobalEnv:
##
##     crim_lvl, train_16
```

```
##
## Call:
## glm(formula = crim_lvl ~ nox + medv, family = binomial, data = Boston.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.17657  -0.38729   0.00523   0.30375   2.65695
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -19.74864    2.42833  -8.133  4.2e-16 ***
## nox          33.97633    3.88025   8.756  < 2e-16 ***
## medv          0.06605    0.02524   2.617  0.00887 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.02  on 359  degrees of freedom
## Residual deviance: 215.41  on 357  degrees of freedom
## AIC: 221.41
##
## Number of Fisher Scoring iterations: 6
```

```
##               observed
## predicted      Below Above
##   Below median    63    16
##   Above median    12    55
```

The number of total observations above is $63 + 16 + 12 + 55 = 146$, so the test error rate is $\frac{16+12}{146} = 19.18\%$.

LDA

```
##               observed
## predicted      Below Above
##   Below median    62    14
##   Above median    13    57
```

For the LDA method, the test error rate here is $\frac{14+13}{62+14+13+57} = 18.49\%$.

Naive Bayes

KNN Models

```
##
## knn.pred.1  0  1
##          0 71  5
##          1  4 66
```

```
##
## knn.pred.2  0  1
##          0 68  5
##          1  7 66
```

When K=3, we have the test error rate as $\frac{4+5}{146} = 6.16\%$. When K=5, we have the test error rate as $\frac{5+7}{146} = 7.53\%$. The best model is the KNN model with K=3. But the difficulty with the KNN approach is that it doesn't tell us which predictors are important and how they can affect the probability of our outcome.If we want to lower the crime rate of a community, we should look at what low-crime communities are doing. Based on our variables selected, we should look at that low-crime communities are doing in terms of air pollution,tax rates and school funding.