Rakesh Veetekat

Conceptual and theoretical questions

1. Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representation and logit representation for the logistic regression model are equivalent.

1.
$$p(X) = \frac{e^{B_0+B_1X}}{1+e^{B_0+B_1X}}$$

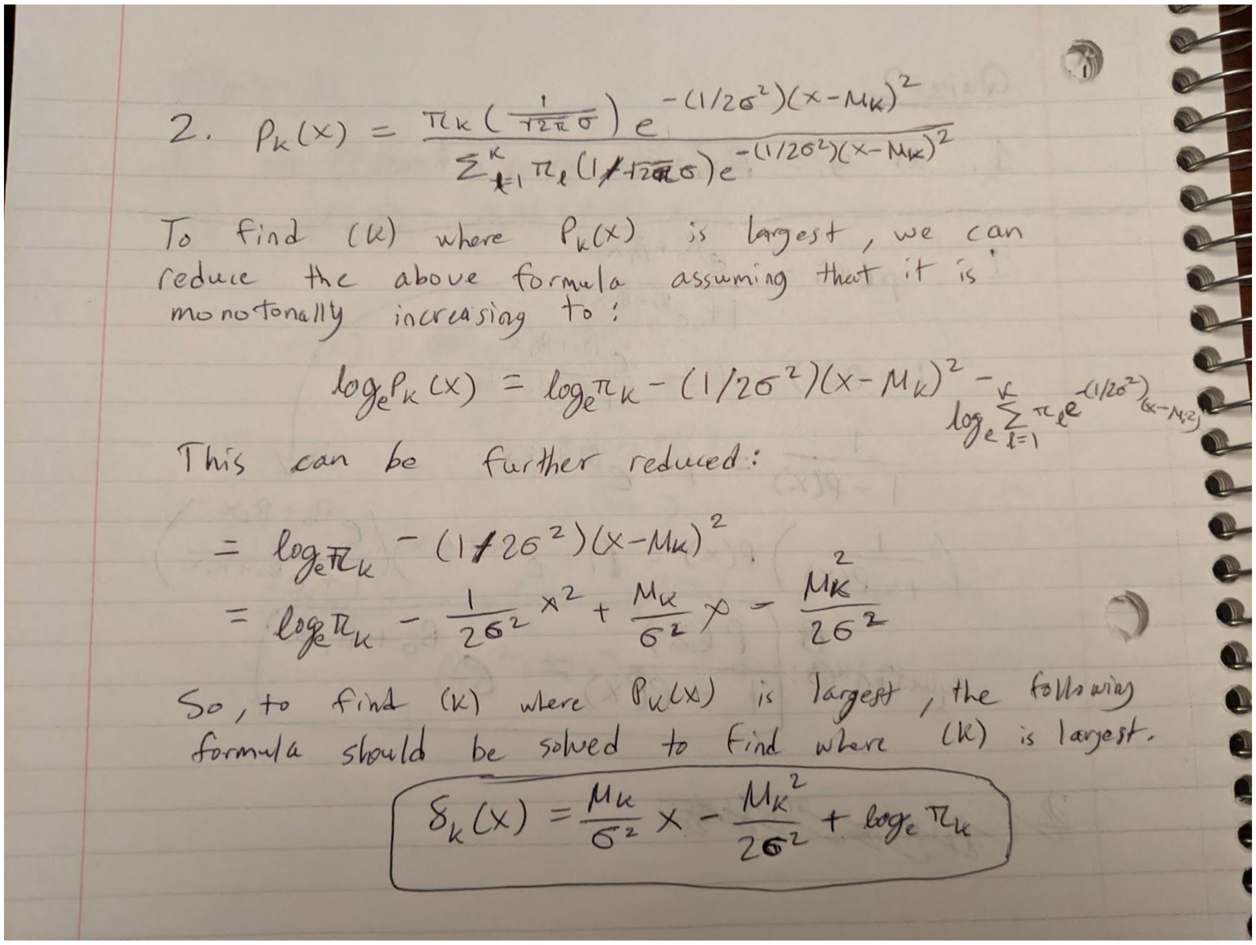
$$1 - p(X) = 1 - \frac{e^{B_0+B_1X}}{1+e^{B_0+B_1X}}$$

$$\frac{1}{1-p(X)} = 1 + e^{B_0+B_1X}$$

$$\left(\frac{1}{1-p(X)}\right) p(X) = \left(1 + e^{B_0+B_1X}\right)\left(\frac{e^{B_0+B_1X}}{1+e^{B_0+B_1X}}\right)$$

equation 4.3: $\boxed{\frac{p(X)}{1-p(X)} = e^{B_0+B_1X}}$

2. It was stated in the text that classifying an observation to the class for which (4.12) is largest is equivalent to classifying an observation to the class for which (4.13) is largest. Prove that this is the case. In other words, under the assumption that the observations in the kth class are drawn from a N(μk, σ2) distribution, the Bayes' classifier assigns an observation to the class for which the discriminant function is maximized.



2. $p_k(x) = \dfrac{\pi_k\left(\frac{1}{\sqrt{2\pi}\sigma}\right) e^{-(1/2\sigma^2)(x-\mu_k)^2}}{\sum_{l=1}^{K} \pi_l (1/\sqrt{2\pi}\sigma) e^{-(1/2\sigma^2)(x-\mu_k)^2}}$

To find (k) where $P_k(x)$ is largest, we can reduce the above formula assuming that it is monotonally increasing to:

$$\log_e P_k(x) = \log_e \pi_k - (1/2\sigma^2)(x-\mu_k)^2 - \log_e \sum_{l=1}^{K} \pi_l e^{-(1/2\sigma^2)(x-\mu_l)^2}$$

This can be further reduced:

$$= \log_e \pi_k - (1/2\sigma^2)(x-\mu_k)^2$$

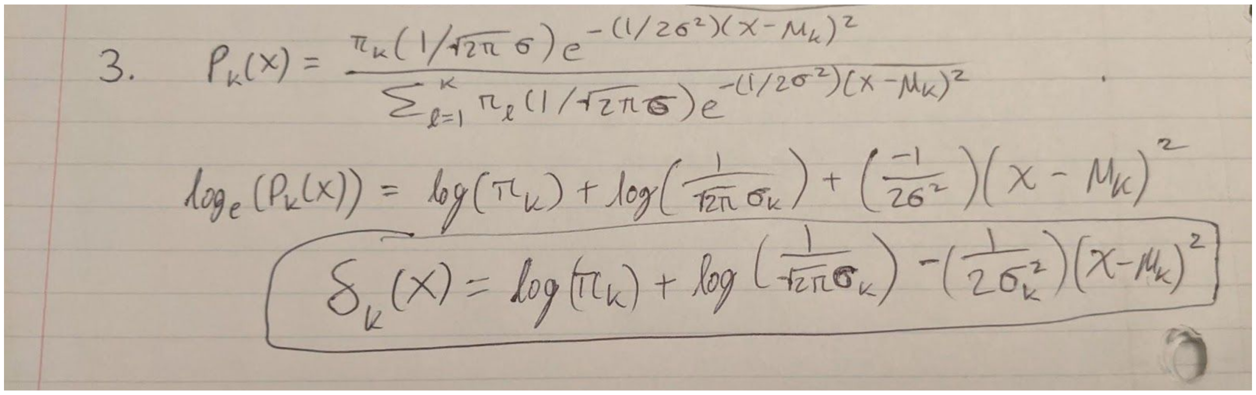
$$= \log_e \pi_k - \frac{1}{2\sigma^2}x^2 + \frac{\mu_k}{\sigma^2}x - \frac{\mu_k^2}{2\sigma^2}$$

So, to find (k) where $P_k(x)$ is largest, the following formula should be solved to find where (k) is largest.

$$\boxed{\delta_k(x) = \frac{\mu_k}{\sigma^2}x - \frac{\mu_k^2}{2\sigma^2} + \log_e \pi_k}$$

3. This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class specific mean vector and a class specific covariance matrix. We consider the simple case where p = 1; i.e. there is only one feature.

   Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal distribution, X ~ N(μk, σ2 k). Recall that the density function for the one-dimensional normal distribution is given in (4.11). Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.



3. $$P_k(x) = \frac{\pi_k(1/\sqrt{2\pi}\,\sigma)e^{-(1/2\sigma^2)(x-\mu_k)^2}}{\sum_{l=1}^{K} \pi_l(1/\sqrt{2\pi}\,\sigma)e^{-(1/2\sigma^2)(x-\mu_k)^2}}$$

$$\log_e(P_k(x)) = \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi}\,\sigma_k}\right) + \left(\frac{-1}{2\sigma^2}\right)(x-\mu_k)^2$$

$$\boxed{\delta_k(x) = \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi}\,\sigma_k}\right) - \left(\frac{1}{2\sigma_k^2}\right)(x-\mu_k)^2}$$

4. We now examine the differences between LDA and QDA.
    a. If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

We would expect QDA to perform better on the training set because of its higher flexibility, and LDA will perform better on the test set because QDA will likely overfit since the Bayes decision boundary is linear.

    b. If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

Since the boundary is non-linear, we would expect QDA to perform better on both the training and test set because LDA usually performs worse in this scenario.

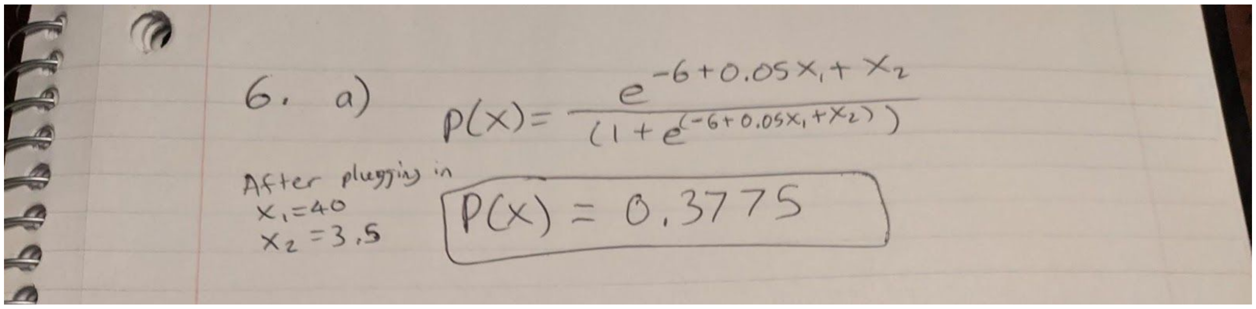
    c. In general, as the sample size n increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?

In general, we would expect it to improve because QDA is flexible and a more flexible method will likely fit the sample better as the size n increases.

    d. True or False: Even if the Bayes decision boundary for a given problem is linear, we will probably achieve a superior test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.

False because less sample points means that the variance in a flexible model, like QDA, may lead to overfit, and therefore, probably a worse test rate to LDA.

5. 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, βˆ0 = −6, βˆ1 = 0.05, βˆ2 = 1.
    a. Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

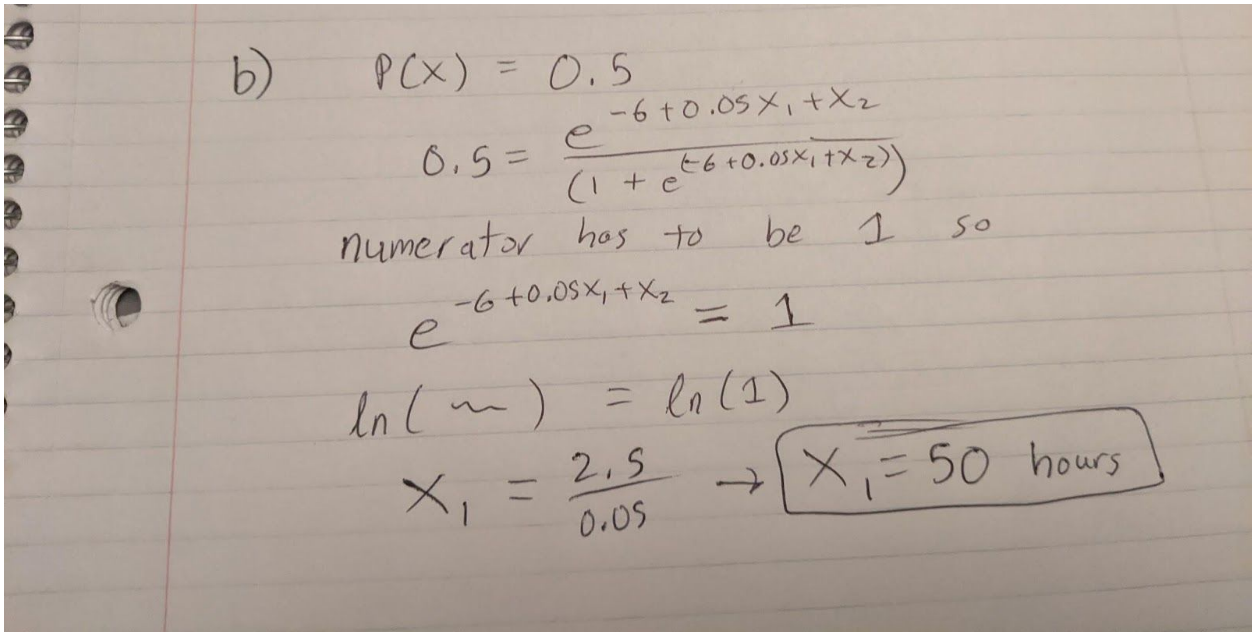


6. a) $p(x) = \frac{e^{-6+0.05x_1+x_2}}{(1+e^{(-6+0.05x_1+x_2)})}$

After plugging in
$x_1=40$
$x_2=3.5$

$\boxed{P(x) = 0.3775}$

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?



b) $P(x) = 0.5$

$0.5 = \frac{e^{-6+0.05x_1+x_2}}{(1+e^{(-6+0.05x_1+x_2)})}$

numerator has to be 1 so

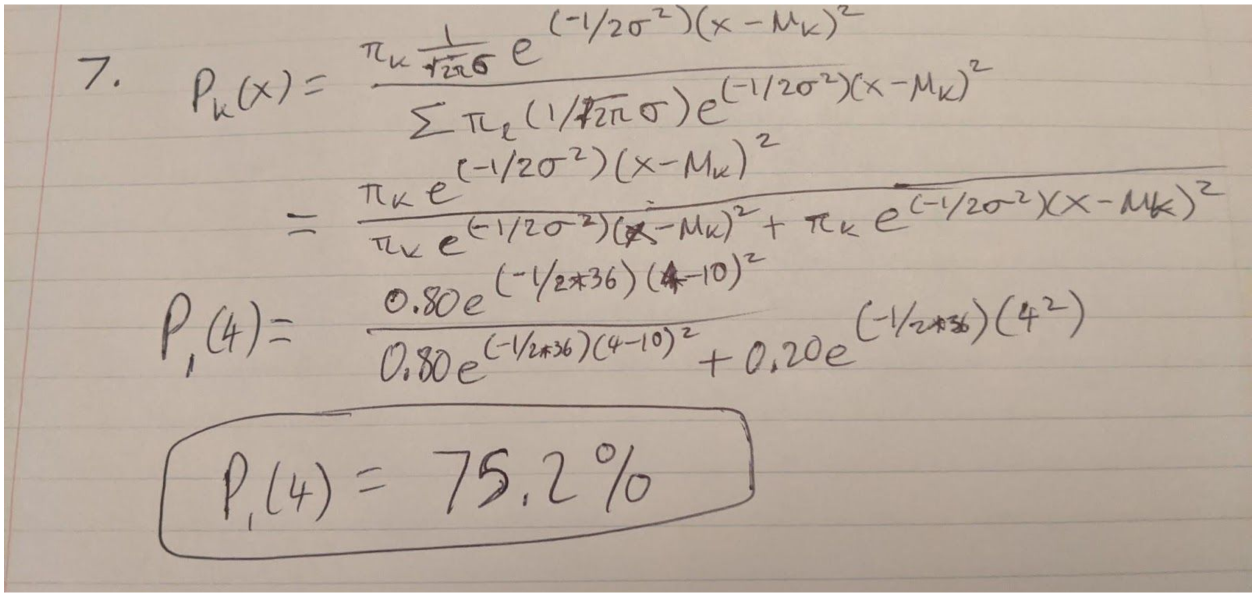
$e^{-6+0.05x_1+x_2} = 1$

$\ln(\sim) = \ln(1)$

$x_1 = \frac{2.5}{0.05} \rightarrow \boxed{x_1 = 50 \text{ hours}}$

6. Suppose that we wish to predict whether a given stock will issue a dividend this year ("Yes" or "No") based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was X¯ = 10, while the mean for those that didn't was X¯ = 0. In addition, the variance of X for these two sets of companies was ˆσ2 = 36. Finally, 80 % of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was X = 4 last year.

   Hint: Recall that the density function for a normal random variable is f(x) = √ 1 2πσ2 e−(x−μ)2/2σ2 . You will need to use Bayes' theorem.



7. $$P_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{(-1/2\sigma^2)(x-\mu_k)^2}}{\sum \pi_l (1/\sqrt{2\pi}\sigma) e^{(-1/2\sigma^2)(x-\mu_k)^2}}$$

$$= \frac{\pi_k e^{(-1/2\sigma^2)(x-\mu_k)^2}}{\pi_k e^{(-1/2\sigma^2)(\cancel{x}-\mu_k)^2} + \pi_k e^{(-1/2\sigma^2)(x-\mu_k)^2}}$$

$$P_1(4) = \frac{0.80 e^{(-1/2*36)(\cancel{4}-10)^2}}{0.80 e^{(-1/2*36)(4-10)^2} + 0.20 e^{(-1/2*36)(4^2)}}$$

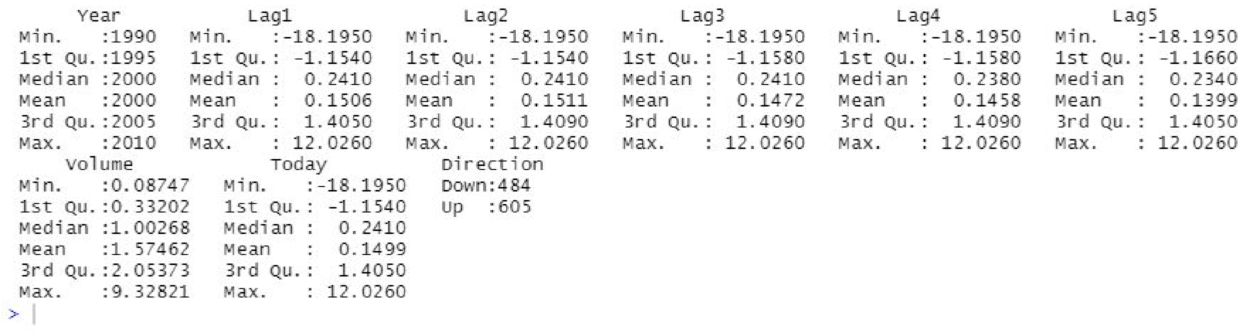
$$\boxed{P_1(4) = 75.2\%}$$

7. 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 (i.e. K = 1) 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?

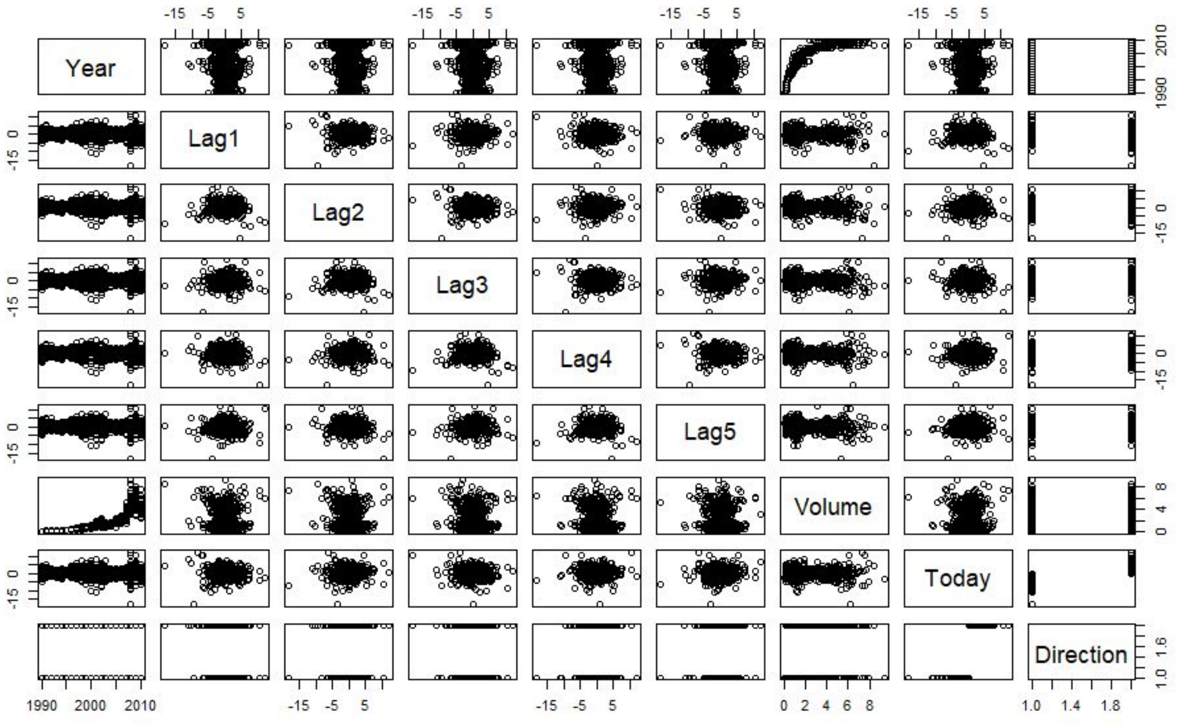
We can deduce that the KNN with K=1 has a test error rate of 36% because it has an average error rate of 18% and its training error rate should be 0%. We should prefer the logistic regression method because it has a 30% test error rate, which is lower than the KNN with K=1 because it has a 36% test error rate.

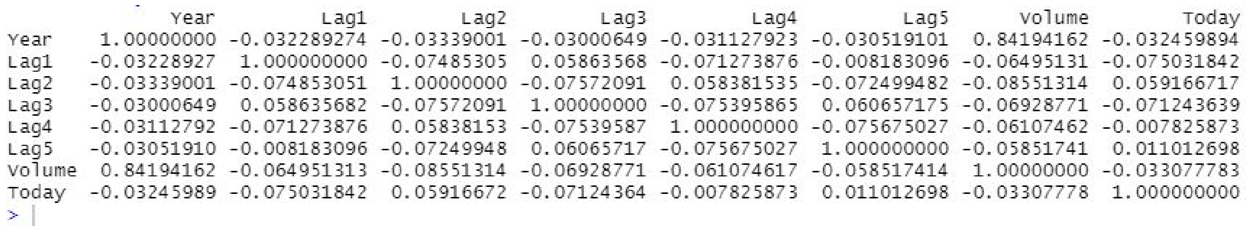
Applied

8. This question should be answered using the Weekly data set, which is part of the ISLR 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?



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

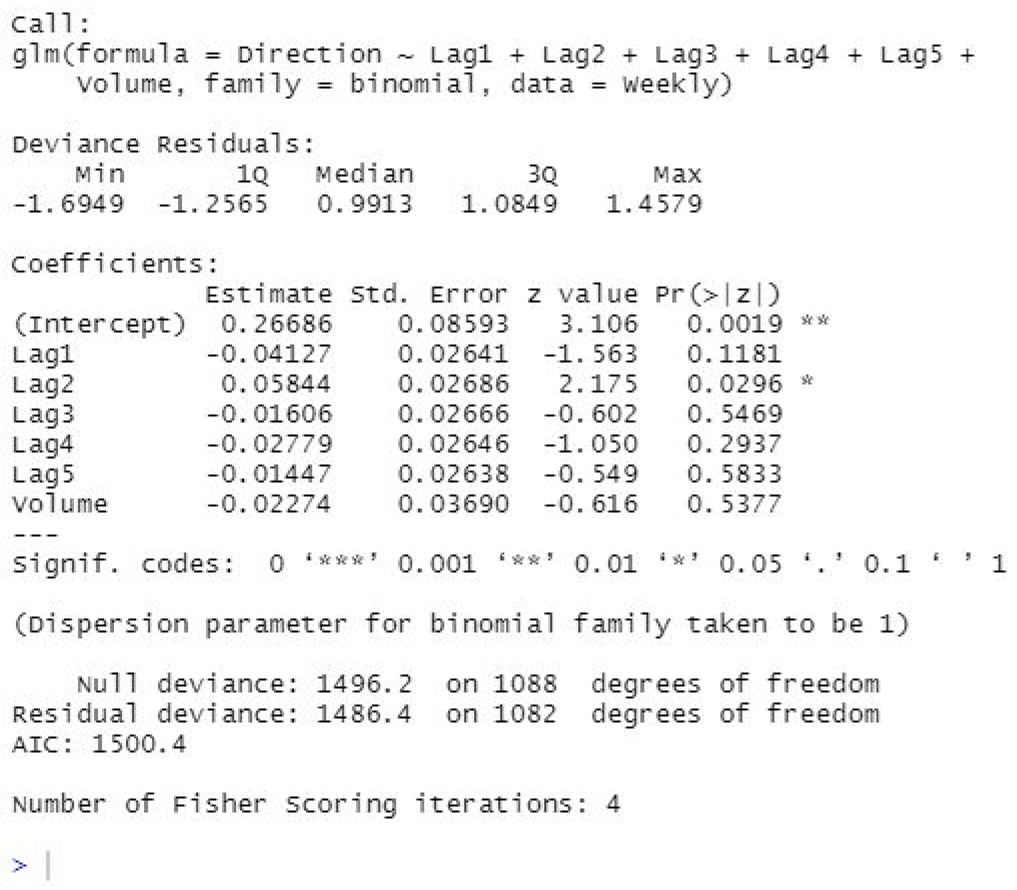




```
           Year         Lag1        Lag2        Lag3         Lag4         Lag5      Volume        Today
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923 -0.030519101  0.84194162 -0.032459894
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876 -0.008183096 -0.06495131 -0.075031842
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535 -0.072499482 -0.08551314  0.059166717
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865  0.060657175 -0.06928771 -0.071243639
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000 -0.075675027 -0.06107462 -0.007825873
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027  1.000000000 -0.05851741  0.011012698
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617 -0.058517414  1.00000000 -0.033077783
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873  0.011012698 -0.03307778  1.000000000
> |
```

From the images above, it can be seen that “Year” and “Volume” appear to have the only significant relationship with the moderately high correlation of 0.84194162 between them.

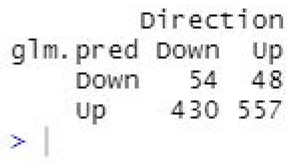
b. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?



```
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

> |
```

The only predictor that has relevant statistical significance is “Lag2” because its of its very low p-value.

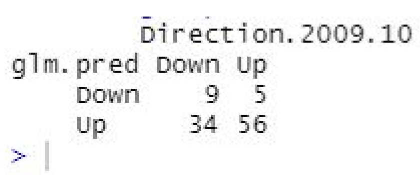
c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.



```
        Direction
glm.pred Down  Up
    Down   54  48
    Up    430 557
> |
```

Percentage of the model making correct predictions: (54 + 557) / (1089 weeks) = 56.1%
Weeks when market goes up, percentage of correct prediction: 557 / (48 + 557) = 92.1%
Weeks when market goes down, percentage of correct prediction: 54 / (430 + 54) = 11.2%

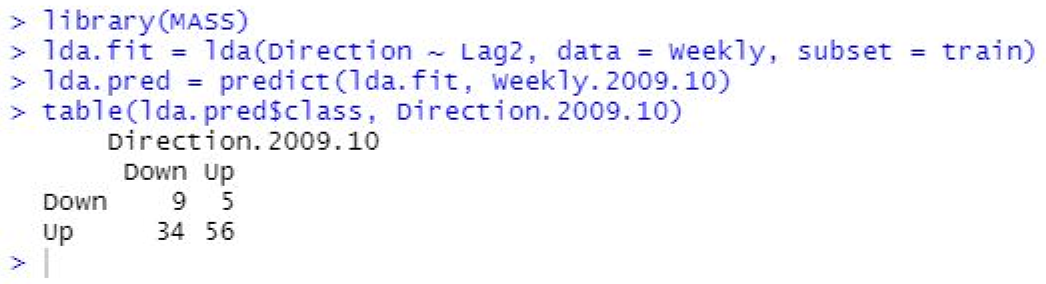
d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).



```
        Direction.2009.10
glm.pred Down Up
    Down    9  5
    Up     34 56
> |
```

Similar to part (c), percentage of correct predictions: (9 + 56) / 104 = 62.5%
When market goes up, percentage of correct predictions: 56 / (56 + 5) = 91.8%
When market goes down, percentage of correct predictions: 9 / (9 + 34) = 20.9%

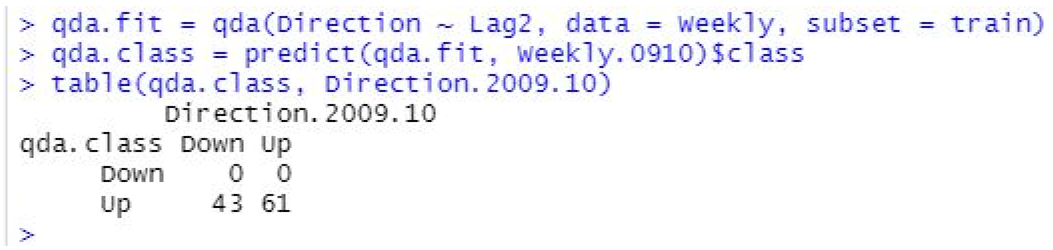
e. Repeat (d) using LDA.



```
> library(MASS)
> lda.fit = lda(Direction ~ Lag2, data = Weekly, subset = train)
> lda.pred = predict(lda.fit, Weekly.2009.10)
> table(lda.pred$class, Direction.2009.10)
    Direction.2009.10
     Down Up
 Down    9  5
 Up     34 56
> |
```

The resulting percentages will be the same as part (d) because the confusion matrix is the same.

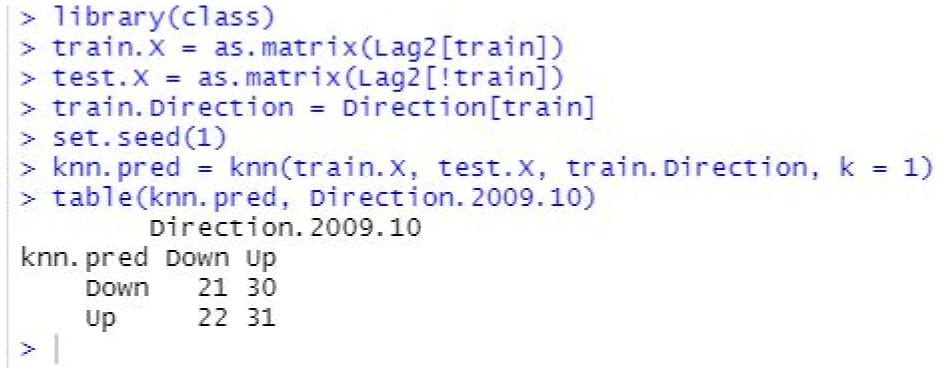
f. Repeat (d) using QDA.



```
> qda.fit = qda(Direction ~ Lag2, data = Weekly, subset = train)
> qda.class = predict(qda.fit, Weekly.0910)$class
> table(qda.class, Direction.2009.10)
         Direction.2009.10
qda.class Down Up
     Down    0  0
     Up     43 61
>
```

We can calculate the percentages in a similar fashion to part (d).
Percentage of correct predictions: (0 + 61) / 104 = 58.7%
When market goes up, percentage of correct predictions: 61 / (61 + 0) = 100%
When market goes down, percentage of correct predictions: 0 / (0 + 43) = 0%

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



```
> library(class)
> train.X = as.matrix(Lag2[train])
> test.X = as.matrix(Lag2[!train])
> train.Direction = Direction[train]
> set.seed(1)
> knn.pred = knn(train.X, test.X, train.Direction, k = 1)
> table(knn.pred, Direction.2009.10)
        Direction.2009.10
knn.pred Down Up
    Down   21 30
    Up     22 31
> |
```

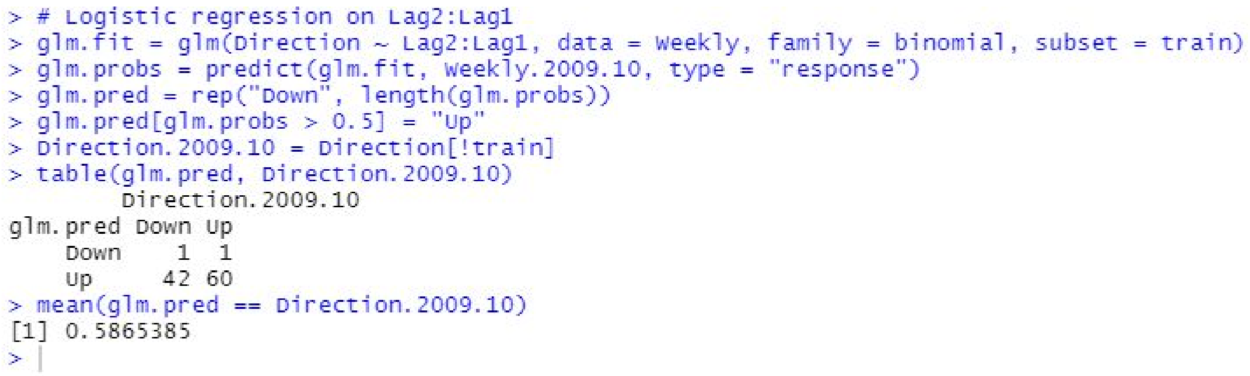
Percentage of correct predictions: (21 + 31) / 104 = 50%

When market goes up, percentage of correct predictions: 31 / (31 + 30) = 50.8%
When market goes down, percentage of correct predictions: 21 / (21 + 22) = 48.8%

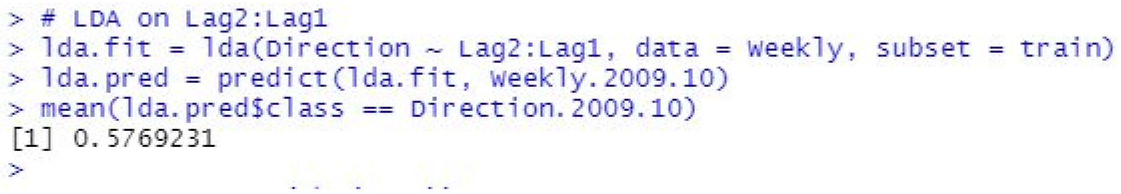
h. Which of these methods appears to provide the best results on this data?

The logistic regression model and LDA provide the best results on the data with the highest percentage of correct predictions value, followed by QDA and then KNN with K = 1.

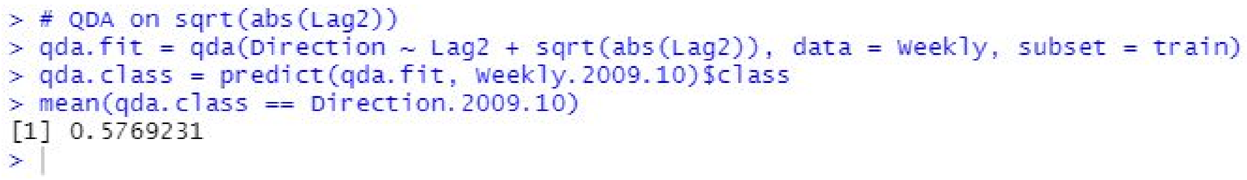
i. Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with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



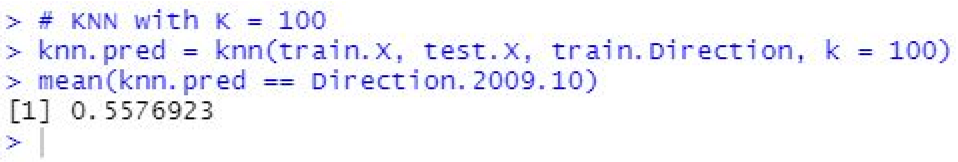
```
> # Logistic regression on Lag2:Lag1
> glm.fit = glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = train)
> glm.probs = predict(glm.fit, Weekly.2009.10, type = "response")
> glm.pred = rep("Down", length(glm.probs))
> glm.pred[glm.probs > 0.5] = "Up"
> Direction.2009.10 = Direction[!train]
> table(glm.pred, Direction.2009.10)
        Direction.2009.10
glm.pred Down Up
    Down    1  1
    Up     42 60
> mean(glm.pred == Direction.2009.10)
[1] 0.5865385
> 
```



```
> # LDA on Lag2:Lag1
> lda.fit = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
> lda.pred = predict(lda.fit, Weekly.2009.10)
> mean(lda.pred$class == Direction.2009.10)
[1] 0.5769231
> 
```



```
> # QDA on sqrt(abs(Lag2))
> qda.fit = qda(Direction ~ Lag2 + sqrt(abs(Lag2)), data = Weekly, subset = train)
> qda.class = predict(qda.fit, Weekly.2009.10)$class
> mean(qda.class == Direction.2009.10)
[1] 0.5769231
> 
```

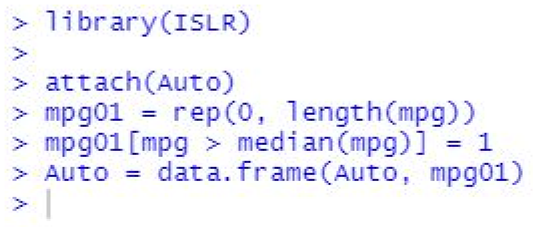


```
> # KNN with K = 100
> knn.pred = knn(train.X, test.X, train.Direction, k = 100)
> mean(knn.pred == Direction.2009.10)
[1] 0.5576923
> 
```

From the different combinations shown above, the logistic regression model performs the best, and is followed by the LDA, QDA, and KNN performing at comparable rates to each other.

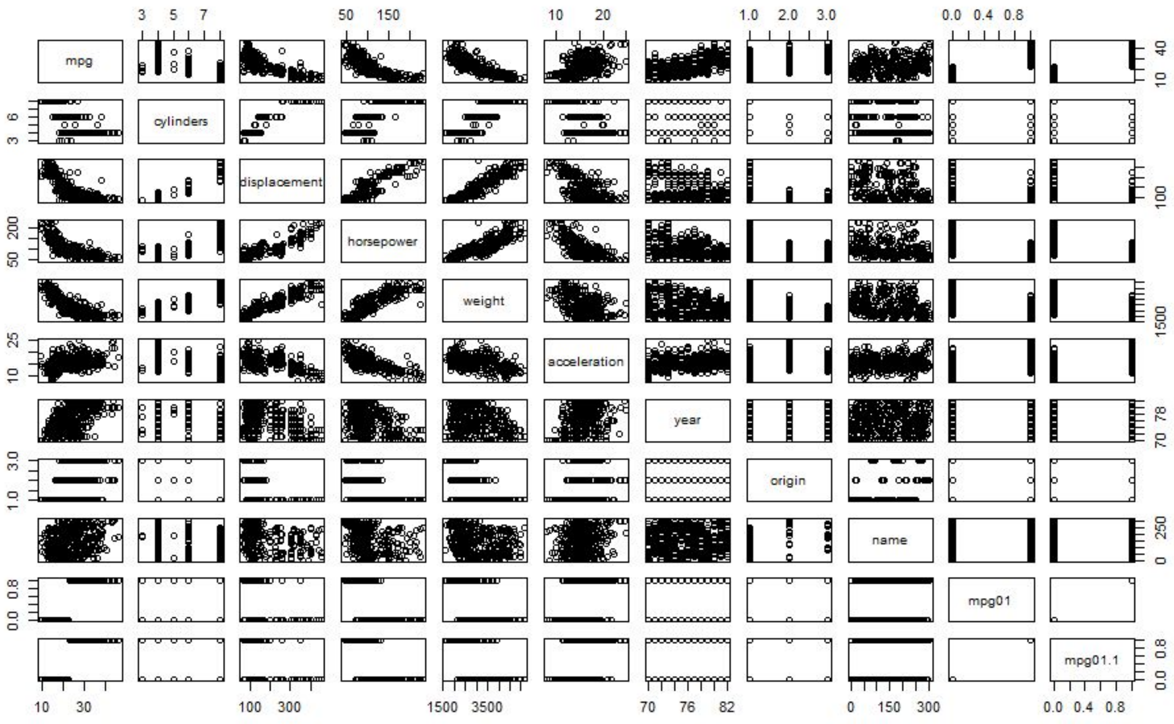
9. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.
   a. Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute

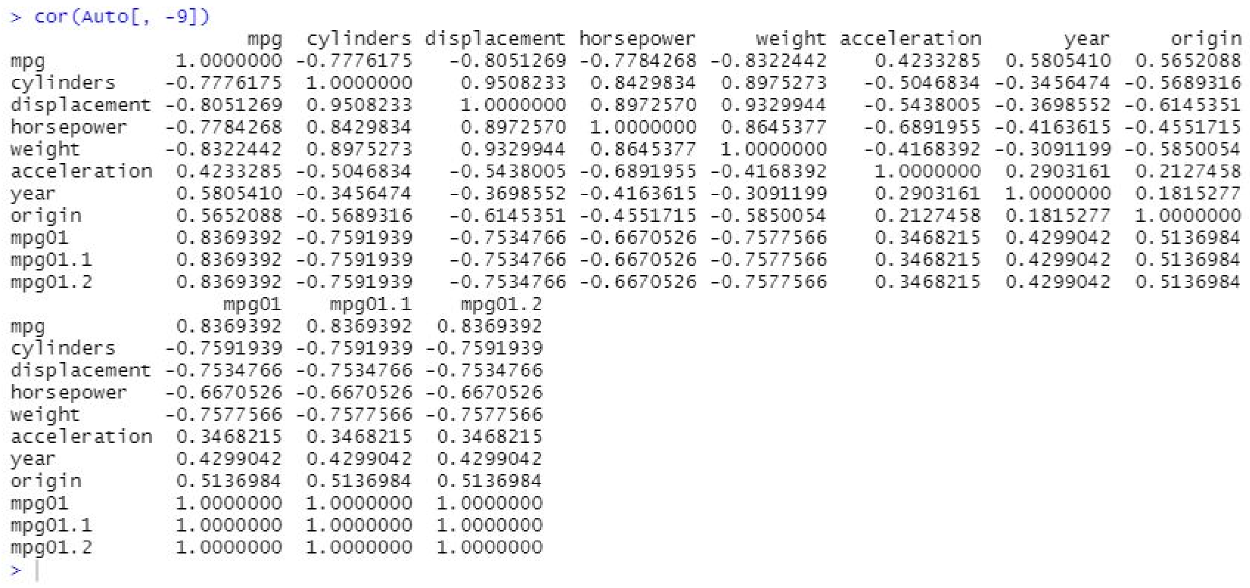
the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.



```
> library(ISLR)
>
> attach(Auto)
> mpg01 = rep(0, length(mpg))
> mpg01[mpg > median(mpg)] = 1
> Auto = data.frame(Auto, mpg01)
> |
```

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.

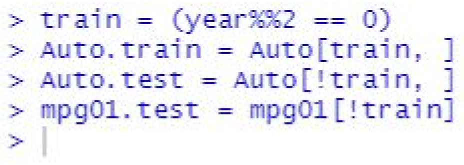




```
> cor(Auto[, -9])
                    mpg  cylinders displacement horsepower     weight acceleration       year     origin
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285  0.5805410  0.5652088
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -0.3456474 -0.5689316
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.6145351
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.4551715
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.5850054
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.2127458
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000  0.1815277
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458  0.1815277  1.0000000
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.5136984
mpg01.1       0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.5136984
mpg01.2       0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215  0.4299042  0.5136984
                  mpg01    mpg01.1    mpg01.2
mpg           0.8369392  0.8369392  0.8369392
cylinders    -0.7591939 -0.7591939 -0.7591939
displacement -0.7534766 -0.7534766 -0.7534766
horsepower   -0.6670526 -0.6670526 -0.6670526
weight       -0.7577566 -0.7577566 -0.7577566
acceleration  0.3468215  0.3468215  0.3468215
year          0.4299042  0.4299042  0.4299042
origin        0.5136984  0.5136984  0.5136984
mpg01         1.0000000  1.0000000  1.0000000
mpg01.1       1.0000000  1.0000000  1.0000000
mpg01.2       1.0000000  1.0000000  1.0000000
> |
```

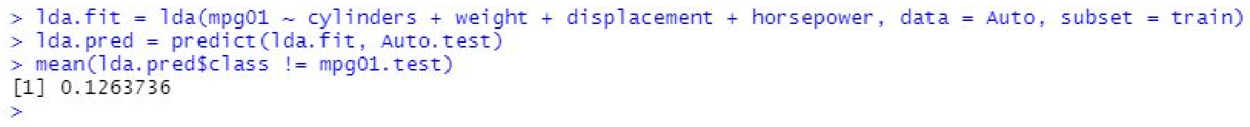
From the scatterplots and correlation matrix, it appears that there are significant relationships between “mpg01” and “cylinders”, “displacement”, “horsepower”, and “weight”.

c. Split the data into a training set and a test set.



```
> train = (year%%2 == 0)
> Auto.train = Auto[train, ]
> Auto.test = Auto[!train, ]
> mpg01.test = mpg01[!train]
> |
```

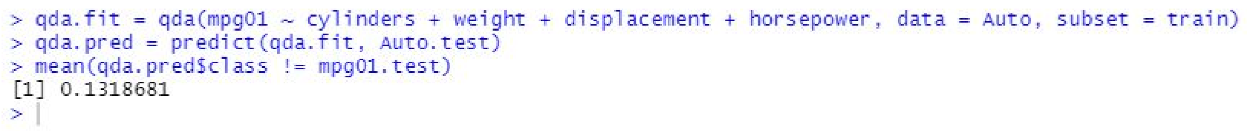
d. Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?



```
> lda.fit = lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)
> lda.pred = predict(lda.fit, Auto.test)
> mean(lda.pred$class != mpg01.test)
[1] 0.1263736
>
```

The test error of the model here is 12.6%.

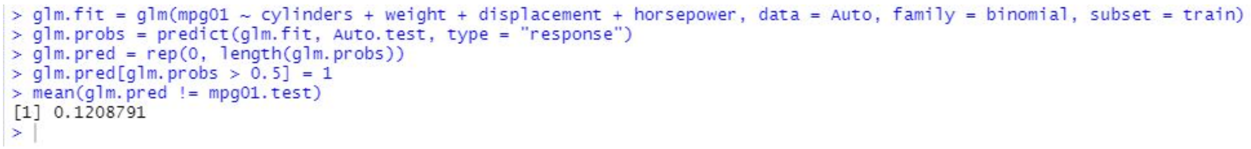
e. Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?



```
> qda.fit = qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)
> qda.pred = predict(qda.fit, Auto.test)
> mean(qda.pred$class != mpg01.test)
[1] 0.1318681
> |
```

The test error for this model is 13.2%.

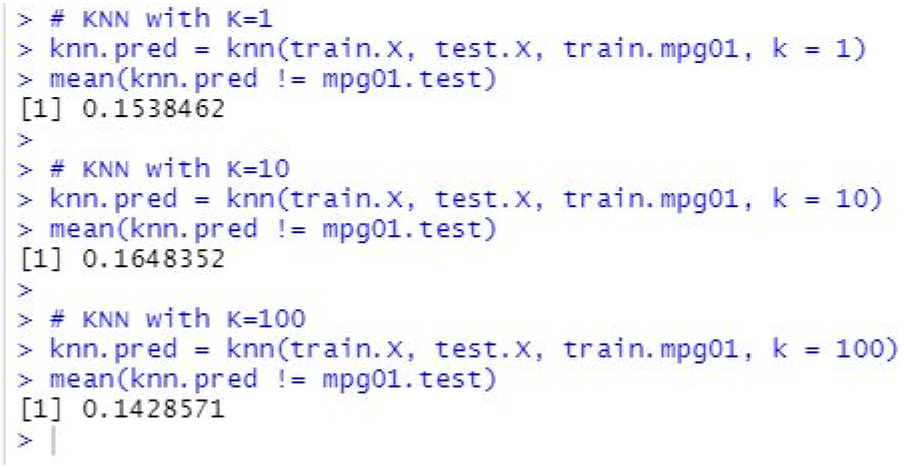
f. Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?



```
> glm.fit = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = binomial, subset = train)
> glm.probs = predict(glm.fit, Auto.test, type = "response")
> glm.pred = rep(0, length(glm.probs))
> glm.pred[glm.probs > 0.5] = 1
> mean(glm.pred != mpg01.test)
[1] 0.1208791
> |
```

The test error for this model is 12.1%.

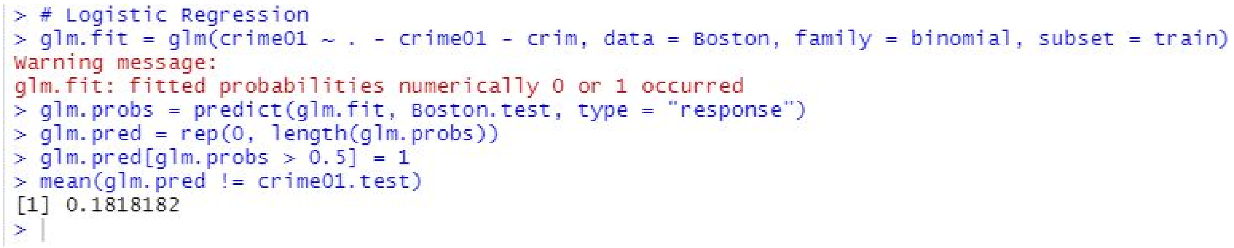
g. Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?



```
> # KNN with K=1
> knn.pred = knn(train.X, test.X, train.mpg01, k = 1)
> mean(knn.pred != mpg01.test)
[1] 0.1538462
>
> # KNN with K=10
> knn.pred = knn(train.X, test.X, train.mpg01, k = 10)
> mean(knn.pred != mpg01.test)
[1] 0.1648352
>
> # KNN with K=100
> knn.pred = knn(train.X, test.X, train.mpg01, k = 100)
> mean(knn.pred != mpg01.test)
[1] 0.1428571
> |
```

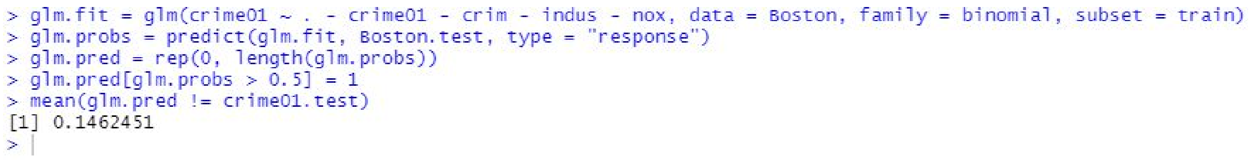
KNN with K=100 performs the best with a test error of 14.3% after running it with K=1, K=10, and K=100.

10. Using the Boston data set, fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA, and KNN models using various subsets of the predictors. Describe your findings.



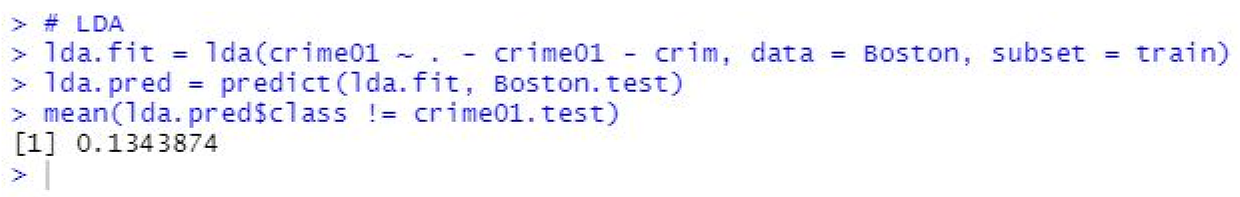
```
> # Logistic Regression
> glm.fit = glm(crime01 ~ . - crime01 - crim, data = Boston, family = binomial, subset = train)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> glm.probs = predict(glm.fit, Boston.test, type = "response")
> glm.pred = rep(0, length(glm.probs))
> glm.pred[glm.probs > 0.5] = 1
> mean(glm.pred != crime01.test)
[1] 0.1818182
> |
```

This logistic regression model has a test error rate of 18.2%.



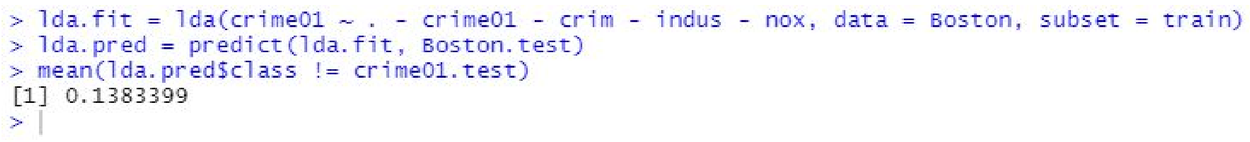
```
> glm.fit = glm(crime01 ~ . - crime01 - crim - indus - nox, data = Boston, family = binomial, subset = train)
> glm.probs = predict(glm.fit, Boston.test, type = "response")
> glm.pred = rep(0, length(glm.probs))
> glm.pred[glm.probs > 0.5] = 1
> mean(glm.pred != crime01.test)
[1] 0.1462451
> |
```

This logistic regression model with more predictors has a test error rate of 14.6%.



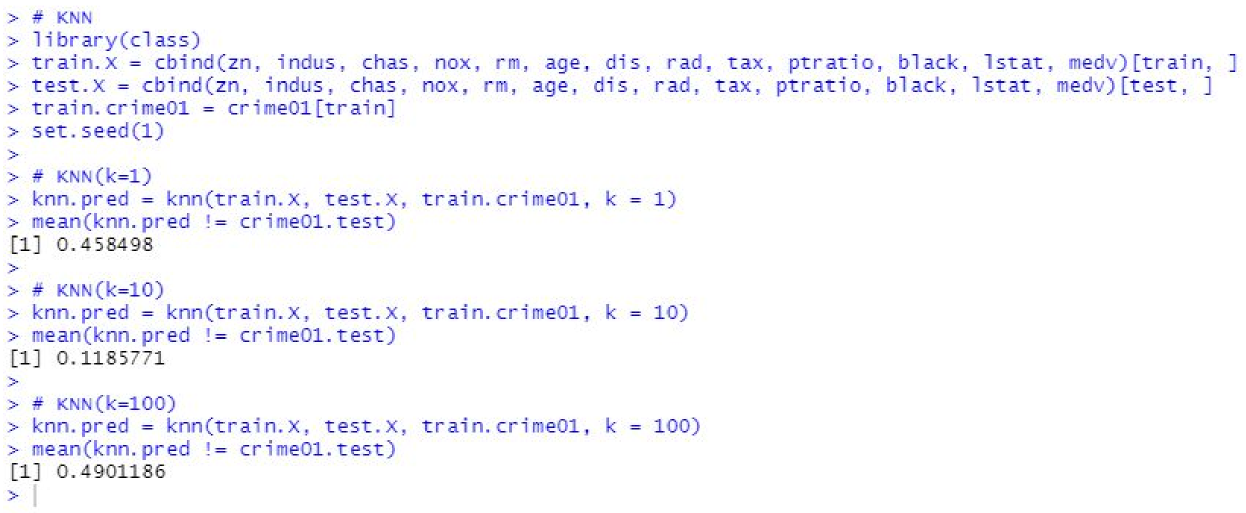
```
> # LDA
> lda.fit = lda(crime01 ~ . - crime01 - crim, data = Boston, subset = train)
> lda.pred = predict(lda.fit, Boston.test)
> mean(lda.pred$class != crime01.test)
[1] 0.1343874
> 
```

This LDA has a test error rate of 13.4%.



```
> lda.fit = lda(crime01 ~ . - crime01 - crim - indus - nox, data = Boston, subset = train)
> lda.pred = predict(lda.fit, Boston.test)
> mean(lda.pred$class != crime01.test)
[1] 0.1383399
> 
```

Meanwhile, this LDA with more predictors has a test error rate of 13.8%.



```
> # KNN
> library(class)
> train.X = cbind(zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, black, lstat, medv)[train, ]
> test.X = cbind(zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, black, lstat, medv)[test, ]
> train.crime01 = crime01[train]
> set.seed(1)
> 
> # KNN(k=1)
> knn.pred = knn(train.X, test.X, train.crime01, k = 1)
> mean(knn.pred != crime01.test)
[1] 0.458498
> 
> # KNN(k=10)
> knn.pred = knn(train.X, test.X, train.crime01, k = 10)
> mean(knn.pred != crime01.test)
[1] 0.1185771
> 
> # KNN(k=100)
> knn.pred = knn(train.X, test.X, train.crime01, k = 100)
> mean(knn.pred != crime01.test)
[1] 0.4901186
> 
```

This KNN has varying test error rates for different values of K with all of the predictors included.
For K=1, the test error rate is 45.8%.
For K=10, the test error rate is 11.9%.
For K=100, the test error rate is 49%.