Tanmay Gupta
tg289@njit.edu

Homework-2

4. I collect a set of data (n = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. Y = β0 + β1X + β2X2 + β3X3 +.

(a) Suppose that the true relationship between X and Y is linear, i.e. Y = β0 + β1X +. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

Answer-> Without knowing more details about the training data, it is difficult to know which training RSS is lower between linear or cubic. However, as the true relationship between X and Y is linear, we may expect the least squares line to be close to the true regression line, and consequently the RSS for the linear regression may be lower than for the cubic regression.

(b) Answer (a) using test rather than training RSS.

Answer-> In this case the test RSS depends upon the test data, so we have not enough information to conclude. However, we may assume that polynomial regression will have a higher test RSS as the overfit from training would have more error than the linear regression.

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

Answer-> Polynomial regression has lower train RSS than the linear fit because of higher flexibility: no matter what the underlying true relationshop is, the more flexible model will closer follow points and reduce train RSS.

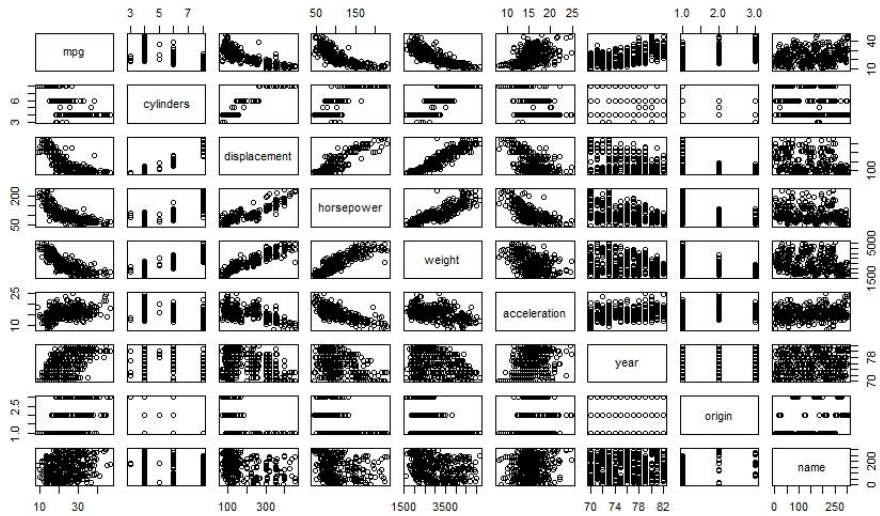
(d) Answer (c) using test rather than training RSS.

Answer-> There is not enough information to tell which test RSS would be lower for either regression given the problem statement is defined as not knowing "how far it is from linear". If it is closer to linear than cubic, the linear regression test RSS could be lower than the cubic regression test RSS. Or, if it is closer to cubic than linear, the cubic regression test RSS could be lower than the linear regression test RSS. It is dues to bias-variance tradeoff: it is not clear what level of flexibility will fit data better.

9. This question involves the use of multiple linear regression on the
Auto data set.
(a) Produce a scatterplot matrix which includes all of the variables in the data set.

Answer->

```
> pairs(Auto)
```



(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, cor() which is qualitative.

Answer->

```
> names(Auto)
 [1] "mpg"          "cylinders"    "displacement" "horsepower"    "weight"       "acceleration" "year"        [8]
"origin"       "name"
```

```
> cor(Auto[1:8])
                    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 -0.568931
6
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.6145
351
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.4551
715
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.585005
4
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.21274
58
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
```

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results.
Comment on the output. For instance:
i. Is there a relationship between the predictors and the response?

Answer->
--Checking the relationship->

```
> model <- lm(mpg ~ . - name, data = Auto)
> summary(model)

Call:
lm(formula = mpg ~ . - name, data = Auto)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-9.5903 -2.1565 -0.1169  1.8690 13.0604 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780    
displacement   0.019896   0.007515   2.647  0.00844 ** 
horsepower    -0.016951   0.013787  -1.230  0.21963    
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548    
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,	Adjusted R-squared:  0.8182 
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

--The F-statistic is 252.4. The p-value corresponding to the F-statistc is 2.037105910^{-139}, this shows that there is relationship between “mpg” and other predictors.

ii. Which predictors appear to have a statistically significant relationship to the response?

Answer-> We can tell this by checking the p-values associated with each predators t-statistic. We may conclude that the predictors are statistically significant except “horsepower”, “cylinder” and “acceleration”.

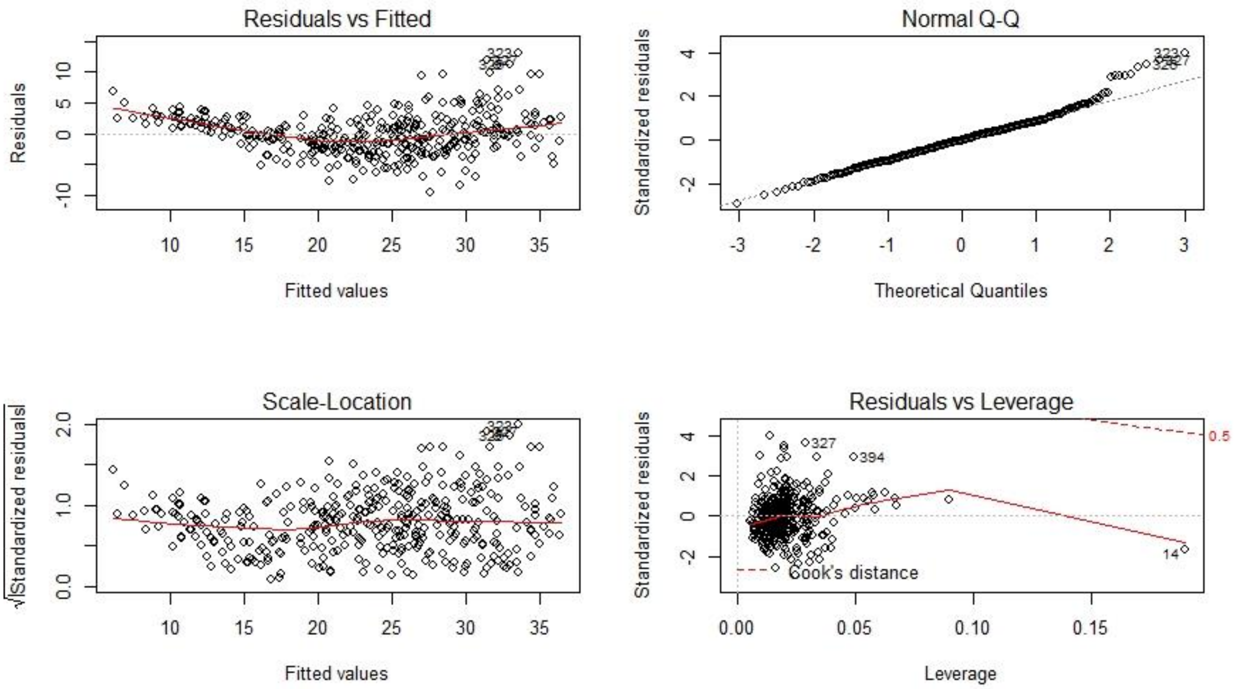
iii. What does the coefficient for the year variable suggest?

Answer-> The coefficient ot the “year” variable suggests that the average effect of an increase of 1 year is an increase of 0.7507727 in “mpg” (all other predictors remaining constant). In other words, cars become more fuel efficient every year by almost 1 mpg / year

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

Answer->

```
> par(mfrow = c(2, 2))
> plot(model)
```



As before, the plot of residuals versus fitted values indicates the presence of mild non linearity in the data. The plot of standardized residuals versus leverage indicates the presence of a few outliers (higher than 2 or lower than -2) and one high leverage point (point 14).

15. This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.
(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

Answers-
--Fitting the model and checking relationship->

```
> attach(Boston)
> model_1 <- lm(crim ~ zn)
> summary(model_2)
Call:
lm(formula = crim ~ zn)
Residuals:
   Min     1Q Median     3Q    Max
-4.429 -4.222 -2.620  1.250 84.523
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
zn          -0.07393    0.01609  -4.594 5.51e-06 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 8.435 on 504 degrees of freedom
Multiple R-squared:  0.04019,	Adjusted R-squared:  0.03828
```

```
F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
> fit.indus <- lm(crim ~ indus)
> summary(fit.indus)
Call:
lm(formula = crim ~ indus)
Residuals:
    Min      1Q  Median      3Q     Max 
-11.972  -2.698  -0.736   0.712  81.813 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.06374    0.66723  -3.093  0.00209 ** 
indus        0.50978    0.05102   9.991  < 2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.866 on 504 degrees of freedom
Multiple R-squared:  0.1653,	Adjusted R-squared:  0.1637 
F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16

> chas <- as.factor(chas)
> fit.chas <- lm(crim ~ chas)
> summary(fit.chas)
Call:
lm(formula = crim ~ chas)
Residuals:
   Min     1Q Median     3Q    Max 
-3.738 -3.661 -3.435  0.018 85.232 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.7444     0.3961   9.453   <2e-16 ***
chas1        -1.8928     1.5061  -1.257    0.209    
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 8.597 on 504 degrees of freedom
Multiple R-squared:  0.003124,	Adjusted R-squared:  0.001146 
F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094

> fit.nox <- lm(crim ~ nox)
> summary(fit.nox)
Call:
lm(formula = crim ~ nox)
Residuals:
    Min      1Q  Median      3Q     Max 
-12.371  -2.738  -0.974   0.559  81.728 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
nox           31.249      2.999  10.419  < 2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.81 on 504 degrees of freedom
Multiple R-squared:  0.1772,	Adjusted R-squared:  0.1756 
F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16

> fit.rm <- lm(crim ~ rm)
> summary(fit.rm)
```

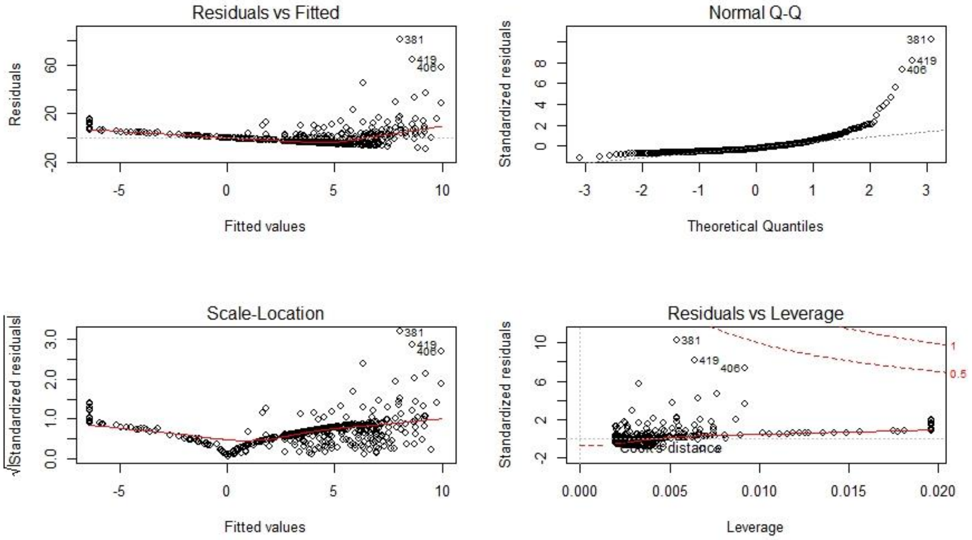
```
Call:
lm(formula = crim ~ rm)
Residuals:
   Min     1Q Median     3Q    Max 
-6.604 -3.952 -2.654  0.989 87.197 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   20.482      3.365   6.088 2.27e-09 ***
rm            -2.684      0.532  -5.045 6.35e-07 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 8.401 on 504 degrees of freedom
Multiple R-squared:  0.04807,	Adjusted R-squared:  0.04618 
F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07


> fit.age <- lm(crim ~ age)
> summary(fit.age)
Call:
lm(formula = crim ~ age)
Residuals:
   Min     1Q Median     3Q    Max 
-6.789 -4.257 -1.230  1.527 82.849 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
age          0.10779    0.01274   8.463 2.85e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 8.057 on 504 degrees of freedom
Multiple R-squared:  0.1244,	Adjusted R-squared:  0.1227 
F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16

> fit.dis <- lm(crim ~ dis)
> summary(fit.dis)
Call:
lm(formula = crim ~ dis)
Residuals:
   Min     1Q Median     3Q    Max 
-6.708 -4.134 -1.527  1.516 81.674 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   9.4993     0.7304  13.006   <2e-16 ***
dis          -1.5509     0.1683  -9.213   <2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.965 on 504 degrees of freedom
Multiple R-squared:  0.1441,	Adjusted R-squared:  0.1425 
F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16

> fit.rad <- lm(crim ~ rad)
> summary(fit.rad)
Call:
lm(formula = crim ~ rad)
Residuals:
    Min      1Q  Median      3Q     Max 
-10.164  -1.381  -0.141   0.660  76.433 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
rad          0.61791    0.03433  17.998  < 2e-16 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 6.718 on 504 degrees of freedom
Multiple R-squared:  0.3913,	Adjusted R-squared:   0.39 
F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16

> fit.tax <- lm(crim ~ tax)
> summary(fit.tax)
Call:
lm(formula = crim ~ tax)
Residuals:
    Min      1Q  Median      3Q     Max 
-12.513  -2.738  -0.194   1.065  77.696 
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
tax          0.029742   0.001847   16.10   <2e-16 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 6.997 on 504 degrees of freedom
Multiple R-squared:  0.3396,	Adjusted R-squared:  0.3383 
F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16

> fit.ptratio <- lm(crim ~ ptratio)
> summary(fit.ptratio)
Call:
lm(formula = crim ~ ptratio)
Residuals:
   Min     1Q Median     3Q    Max 
-7.654 -3.985 -1.912  1.825 83.353 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
ptratio       1.1520     0.1694   6.801 2.94e-11 ***
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 8.24 on 504 degrees of freedom
Multiple R-squared:  0.08407,	Adjusted R-squared:  0.08225 
F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11


> fit.black <- lm(crim ~ black)
> summary(fit.black)
Call:
lm(formula = crim ~ black)
Residuals:
    Min      1Q  Median      3Q     Max 
-13.756  -2.299  -2.095  -1.296  86.822 
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 16.553529   1.425903  11.609   <2e-16 ***
black       -0.036280   0.003873  -9.367   <2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.946 on 504 degrees of freedom
Multiple R-squared:  0.1483,       Adjusted R-squared:  0.1466
F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16

> fit.lstat <- lm(crim ~ lstat)
> summary(fit.lstat)
Call:
lm(formula = crim ~ lstat)
Residuals:
    Min      1Q  Median      3Q     Max
-13.925  -2.822  -0.664   1.079  82.862
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
lstat        0.54880    0.04776  11.491  < 2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.664 on 504 degrees of freedom
Multiple R-squared:  0.2076,       Adjusted R-squared:  0.206
F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16


> fit.medv <- lm(crim ~ medv)
> summary(fit.medv)
Call:
lm(formula = crim ~ medv)
Residuals:
   Min     1Q Median     3Q    Max
-9.071 -4.022 -2.343  1.298 80.957
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.79654    0.93419   12.63   <2e-16 ***
medv        -0.36316    0.03839   -9.46   <2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 7.934 on 504 degrees of freedom
Multiple R-squared:  0.1508,       Adjusted R-squared:  0.1491
F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

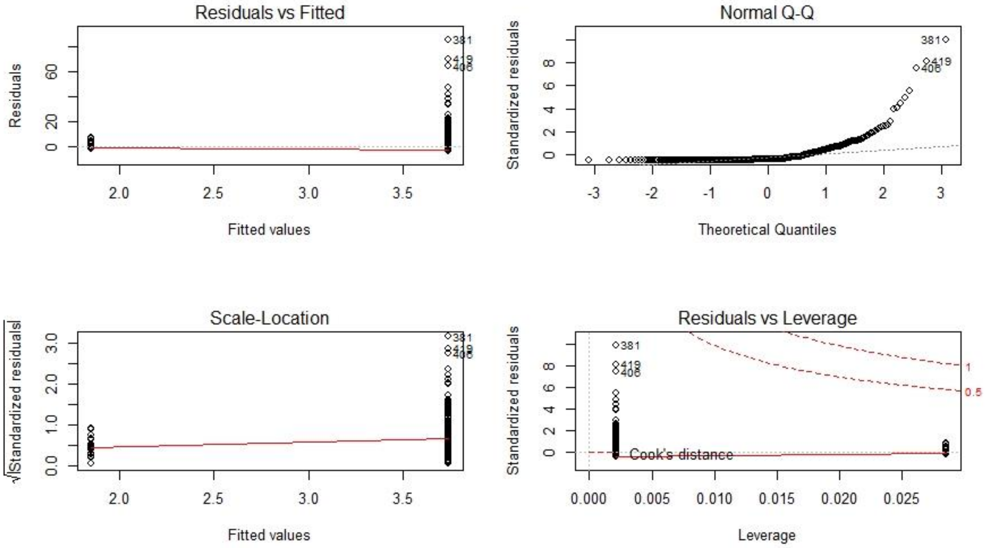
To find which predictors are significant, we have to test H0:β1=0. All predictors have a p-value less than 0.05 except “chas”, so we may conclude that there is a statistically significant association between each predictor and the response except for the “chas” predictor.

Some plots to strengthen the assertion-

<u>for crim and medv</u>-



<u>for crim and chas-</u>



(b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : βj = 0?

Answer->

```
> model.for.all <- lm(crim ~ ., data = Boston)
> summary(model.for.all)
Call:
lm(formula = crim ~ ., data = Boston)
Residuals:
   Min     1Q Median     3Q    Max
-9.924 -2.120 -0.353  1.019 75.051
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.033228   7.234903   2.354 0.018949 *
zn            0.044855   0.018734   2.394 0.017025 *
```

```
indus       -0.063855   0.083407  -0.766 0.444294
chas        -0.749134   1.180147  -0.635 0.525867
nox        -10.313535   5.275536  -1.955 0.051152 .
rm           0.430131   0.612830   0.702 0.483089
age          0.001452   0.017925   0.081 0.935488
dis         -0.987176   0.281817  -3.503 0.000502 ***
rad          0.588209   0.088049   6.680 6.46e-11 ***
tax         -0.003780   0.005156  -0.733 0.463793
ptratio     -0.271081   0.186450  -1.454 0.146611
black       -0.007538   0.003673  -2.052 0.040702 *
lstat        0.126211   0.075725   1.667 0.096208 .
medv        -0.198887   0.060516  -3.287 0.001087 **
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.439 on 492 degrees of freedom
Multiple R-squared:  0.454,	Adjusted R-squared:  0.4396
F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

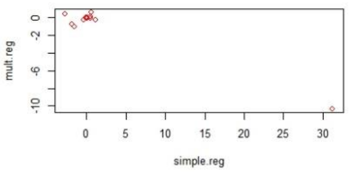
--We can see that for zn, dis, rad, black, medy the T-value is not between -2 and 2 , rest all have t-value between -2 and 2.

--We may reject the null hypothesis for "zn", "dis", "rad", "black" and "medv"

(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a
single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

Answer->

```
> simple.reg <- vector("numeric",0)
> simple.reg <- c(simple.reg, fit.zn$coefficient[2])
> simple.reg <- c(simple.reg, fit.indus$coefficient[2])
> simple.reg <- c(simple.reg, fit.chas$coefficient[2])
> simple.reg <- c(simple.reg, fit.nox$coefficient[2])
> simple.reg <- c(simple.reg, fit.rm$coefficient[2])
> simple.reg <- c(simple.reg, fit.age$coefficient[2])
> simple.reg <- c(simple.reg, fit.dis$coefficient[2])
> simple.reg <- c(simple.reg, fit.rad$coefficient[2])
> simple.reg <- c(simple.reg, fit.tax$coefficient[2])
> simple.reg <- c(simple.reg, fit.ptratio$coefficient[2])
> simple.reg <- c(simple.reg, fit.black$coefficient[2])
> simple.reg <- c(simple.reg, fit.lstat$coefficient[2])
> simple.reg <- c(simple.reg, fit.medv$coefficient[2])
> mult.reg <- vector("numeric", 0)
> mult.reg <- c(mult.reg, model.for.all$coefficients)
> mult.reg <- mult.reg[-1]
> plot(simple.reg, mult.reg, col = "red")
```



There is a difference between the simple and multiple regression coefficients. This difference is due to the fact that in the simple regression case, the slope term represents the average effect of an increase in the predictor, ignoring other predictors. In contrast, in the multiple regression case, the slope term represents the average effect of an increase in the predictor, while holding other predictors fixed. It does make sense for the multiple regression to suggest no relationship between the response and some of the predictors while the simple linear regression implies the opposite because the correlation between the predictors show some strong relationships between some of the predictors.

```
> cor(Boston[-c(1, 4)])
                zn      indus        nox         rm        age        dis        rad        tax    ptratio      black
zn       1.0000000 -0.5338282 -0.5166037  0.3119906 -0.5695373  0.6644082 -0.3119478 -0.3145633 -0.391
6785  0.1755203
indus   -0.5338282  1.0000000  0.7636514 -0.3916759  0.6447785 -0.7080270  0.5951293  0.7207602  0.38
32476 -0.3569765
nox     -0.5166037  0.7636514  1.0000000 -0.3021882  0.7314701 -0.7692301  0.6114406  0.6680232  0.188
9327 -0.3800506
rm       0.3119906 -0.3916759 -0.3021882  1.0000000 -0.2402649  0.2052462 -0.2098467 -0.2920478 -0.35
55015  0.1280686
age     -0.5695373  0.6447785  0.7314701 -0.2402649  1.0000000 -0.7478805  0.4560225  0.5064556  0.261
5150 -0.2735340
dis      0.6644082 -0.7080270 -0.7692301  0.2052462 -0.7478805  1.0000000 -0.4945879 -0.5344316 -0.232
4705  0.2915117
rad     -0.3119478  0.5951293  0.6114406 -0.2098467  0.4560225 -0.4945879  1.0000000  0.9102282  0.464
7412 -0.4444128
tax     -0.3145633  0.7207602  0.6680232 -0.2920478  0.5064556 -0.5344316  0.9102282  1.0000000  0.460
8530 -0.4418080
ptratio -0.3916785  0.3832476  0.1889327 -0.3555015  0.2615150 -0.2324705  0.4647412  0.4608530  1.00
00000 -0.1773833
black    0.1755203 -0.3569765 -0.3800506  0.1280686 -0.2735340  0.2915117 -0.4444128 -0.4418080 -0.17
73833  1.0000000
lstat   -0.4129946  0.6037997  0.5908789 -0.6138083  0.6023385 -0.4969958  0.4886763  0.5439934  0.374
0443 -0.3660869
medv     0.3604453 -0.4837252 -0.4273208  0.6953599 -0.3769546  0.2499287 -0.3816262 -0.4685359 -0.5
077867  0.3334608
             lstat       medv
zn      -0.4129946  0.3604453
indus    0.6037997 -0.4837252
nox      0.5908789 -0.4273208
rm      -0.6138083  0.6953599
age      0.6023385 -0.3769546
```

```
dis     -0.4969958  0.2499287
rad      0.4886763 -0.3816262
tax      0.5439934 -0.4685359
ptratio  0.3740443 -0.5077867
black   -0.3660869  0.3334608
lstat    1.0000000 -0.7376627
medv    -0.7376627  1.0000000
```

--So for example, when “age” is high there is a tendency in “dis” to be low, hence in simple linear regression which only examines “crim” versus “age”, we observe that higher values of “age” are associated with higher values of “crim”, even though “age” does not actually affect “crim”. So “age” is a surrogate for “dis”; “age” gets credit for the effect of “dis” on “crim”.

(d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form
Y = β0 + β1X + β2X2 + β3X3.

Answer->

--checking association->

```
> fit.zn2 <- lm(crim ~ poly(zn, 3))
> summary(fit.zn2)

Call:
lm(formula = crim ~ poly(zn, 3))

Residuals:
   Min     1Q Median     3Q    Max 
-4.821 -4.614 -1.294  0.473 84.130 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3.6135     0.3722   9.709  < 2e-16 ***
poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 ** 
poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.372 on 502 degrees of freedom
Multiple R-squared:  0.05824,	Adjusted R-squared:  0.05261 
F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06

> fit.indus2 <- lm(crim ~ poly(indus, 3))
> summary(fit.indus2)

Call:
lm(formula = crim ~ poly(indus, 3))

Residuals:
   Min     1Q Median     3Q    Max 
-8.278 -2.514  0.054  0.764 79.713 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
```

```
(Intercept)       3.614      0.330  10.950  < 2e-16 ***
poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.423 on 502 degrees of freedom
Multiple R-squared:  0.2597,     Adjusted R-squared:  0.2552
F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.nox2 <- lm(crim ~ poly(nox, 3))
> summary(fit.nox2)

Call:
lm(formula = crim ~ poly(nox, 3))

Residuals:
   Min     1Q Median     3Q    Max
-9.110 -2.068 -0.255  0.739 78.302

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6135     0.3216  11.237  < 2e-16 ***
poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.234 on 502 degrees of freedom
Multiple R-squared:  0.297,      Adjusted R-squared:  0.2928
F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.rm2 <- lm(crim ~ poly(rm, 3))
> summary(fit.rm2)

Call:
lm(formula = crim ~ poly(rm, 3))

Residuals:
    Min      1Q  Median      3Q     Max
-18.485  -3.468  -2.221  -0.015  87.219

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.33 on 502 degrees of freedom
Multiple R-squared:  0.06779,   Adjusted R-squared:  0.06222
F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```
> fit.age2 <- lm(crim ~ poly(age, 3))
> summary(fit.age2)

Call:
lm(formula = crim ~ poly(age, 3))

Residuals:
   Min     1Q Median     3Q    Max 
-9.762 -2.673 -0.516  0.019 82.842 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3.6135     0.3485  10.368  < 2e-16 ***
poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
poly(age, 3)3  21.3532     7.8397   2.724  0.00668 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.84 on 502 degrees of freedom
Multiple R-squared:  0.1742,	Adjusted R-squared:  0.1693 
F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.dis2 <- lm(crim ~ poly(dis, 3))
> summary(fit.dis2)

Call:
lm(formula = crim ~ poly(dis, 3))

Residuals:
    Min      1Q  Median      3Q     Max 
-10.757  -2.588   0.031   1.267  76.378 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.331 on 502 degrees of freedom
Multiple R-squared:  0.2778,	Adjusted R-squared:  0.2735 
F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.rad2 <- lm(crim ~ poly(rad, 3))
> summary(fit.rad2)

Call:
lm(formula = crim ~ poly(rad, 3))

Residuals:
    Min      1Q  Median      3Q     Max 
-10.381  -0.412  -0.269   0.179  76.217 
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6135     0.2971  12.164  < 2e-16 ***
poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.682 on 502 degrees of freedom
Multiple R-squared:    0.4,	Adjusted R-squared:  0.3965
F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
> fit.tax2 <- lm(crim ~ poly(tax, 3))
> summary(fit.tax2)

Call:
lm(formula = crim ~ poly(tax, 3))

Residuals:
    Min      1Q  Median      3Q     Max
-13.273  -1.389   0.046   0.536  76.950

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.6135     0.3047  11.860  < 2e-16 ***
poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.854 on 502 degrees of freedom
Multiple R-squared:  0.3689,	Adjusted R-squared:  0.3651
F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.black2 <- lm(crim ~ poly(black, 3))
> summary(fit.black2)

Call:
lm(formula = crim ~ poly(black, 3))

Residuals:
    Min      1Q  Median      3Q     Max
-13.096  -2.343  -2.128  -1.439  86.790

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.6135     0.3536  10.218   <2e-16 ***
poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
poly(black, 3)2   5.9264     7.9546   0.745    0.457
poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.955 on 502 degrees of freedom
```

```
Multiple R-squared:  0.1498,	Adjusted R-squared:  0.1448 
F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.lstat2 <- lm(crim ~ poly(lstat, 3))
> summary(fit.lstat2)

Call:
lm(formula = crim ~ poly(lstat, 3))

Residuals:
    Min      1Q  Median      3Q     Max 
-15.234  -2.151  -0.486   0.066  83.353 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       3.6135     0.3392  10.654   <2e-16 ***
poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *  
poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.629 on 502 degrees of freedom
Multiple R-squared:  0.2179,	Adjusted R-squared:  0.2133 
F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16

> fit.medv2 <- lm(crim ~ poly(medv, 3))
> summary(fit.medv2)

Call:
lm(formula = crim ~ poly(medv, 3))

Residuals:
    Min      1Q  Median      3Q     Max 
-24.427  -1.976  -0.437   0.439  73.655 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)       3.614      0.292  12.374  < 2e-16 ***
poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.569 on 502 degrees of freedom
Multiple R-squared:  0.4202,	Adjusted R-squared:  0.4167 
F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Conclusion- For “zn”, “rm”, “rad”, “tax” and “lstat” as predictor, the p-values suggest that the cubic coefficient is not statistically significant; for “indus”, “nox”, “age”, “dis”, “ptratio” and “medv” as predictor, the p-values suggest the adequacy of the cubic fit; for “black” as predictor, the p-values suggest that the quandratic and cubic coefficients are not statistically significant, so in this latter case no non-linear effect is visible

4. When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that
parametric approaches often perform poorly when p is large. We will now investigate this curse.

(a) Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55, 0.65]. On average,
what fraction of the available observations will we use to make the prediction?

Answer-> If X is an uniformly distributed random variable on [0,1], then $P(X\le x)=F(x)=x$ where $0\le x\le 1$. Thus, for any observation xi on [0,1] the probability that a value is on the interval [x_1-.05,x_1+.05] is **10%** since
$P(X\le x)=\int xi+.05xi-.05f(x)dx=F(x1-.05)-F(x1+.05)=(xi+.05)-(xi-.05)=.1$ or 10%

(b) Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1,X2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation's response using only observations that are within 10% of the range of X1 and within 10% of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?

Answer-> Similarly to the previous problem, If X1, and X2 are uniformly distributed random variables then there is a 10% chance of any observation being in the 10% neighborhood of either X1, or X2.

Assuming that X1 and X2 are independent and let $x\in X1$ and $y\in X2$, and $A:=[x-.05,x+0.05]$, $B:=[y-0.05,y+0.05]$ so that A and B are neighborhood of X1, and X2 respectively. Then
$p(x\in A,y\in B)=P(x\in A)P(y\in B)=0.01$ or 1%
Thus, on average only 1% of observations are within the 10% neighborhood of both cases.

(c) Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations
within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

Answer-> 0.1^100

(d) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.

Answer-> From the previous question one can note that if all features are independent, and uniformly distributed on [0,1] then with each new feature the amount of neighbors close to an observation decreases. In practice this means that KNN may use neighbors that are very far from a particular observation, yielding very poor results.

(e) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p = 1, 2, and 100, what is the length of
each side of the hypercube? Comment on your answer.
Note: A hypercube is a generalization of a cube to an arbitrary number of dimensions. When p = 1, a hypercube is simply a line segment, when p = 2 it is a square, and when p = 100 it is a 100-dimensional cube.

Answer-> The length of each side will be equal to the dimensions of the hypercube. for example- if the dimension is 1, then side is 1 , if the dimension is 2 then the length of each side is 2 and if it is 50 then each side is 50.

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 (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?

Answer-> For the case that K = 1, KNN will give a classification error of 0% for training data, since the closest point is itself. Since the average error is 18%, then the error on the test set is 36%, which is 6% higher than the test error obtained by logistic regression. This indicates that logistic regression is a better approach.