Multiple linear regression

Reg_No : 16BCE2310

Name: Aditya Ranjan

Lab Slot: L59+L60

1st application with iris data sets

- ✓ **Normalize the data set (**explain in 6 to 7 sentences if it is needed to normalize & Why? Moreover, drop **Species** attribute as it is string so that you will be left with 4 attributes only)

Formula/definition of normalization is as below,

[

If you want to normalize your data, you can do so as you suggest and simply calculate the following:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

where $x = (x_1, \dots, x_n)$ and z_i is now your i^{th} normalized data. As a proof of concept (although you did not ask for it) here is some `R` code and accompanying graph to illustrate this point:

]

```
> x1=iris[,1]
> x1=(x1-min(x1))/(max(x1)-min(x1))
> x2=iris[,2]
> x2=(x2-min(x2))/(max(x2)-min(x2))
> x3=iris[,3]
> x3=(x3-min(x3))/(max(x3)-min(x3))
> x4=iris[,4]
> x4=(x4-min(x4))/(max(x4)-min(x4))
> data=data.frame(x1,x2,x3,x4)
> data
           x1         x2         x3         x4
1  0.22222222 0.62500000 0.06779661 0.04166667
2  0.16666667 0.41666667 0.06779661 0.04166667
3  0.11111111 0.50000000 0.05084746 0.04166667
4  0.08333333 0.45833333 0.08474576 0.04166667
5  0.19444444 0.66666667 0.06779661 0.04166667
6  0.30555556 0.79166667 0.11864407 0.12500000
7  0.08333333 0.58333333 0.06779661 0.08333333
8  0.19444444 0.58333333 0.08474576 0.04166667
9  0.02777778 0.37500000 0.06779661 0.04166667
10 0.16666667 0.45833333 0.08474576 0.00000000
11 0.30555556 0.70833333 0.08474576 0.04166667
```

- ✓ Clean the data (if it is needed)

Not required

- ✓ Which is the dependent variable and which are the independent variables (of your choice)

Dependent - Sepal.Length

Independent - Sepal.Width + Petal.Length + Petal.Width+ Species

- ✓ Prepare the data set with training and testing (consider 80% - 20%);

```
> dt=sample(nrow(iris), nrow(iris)*.8)
> irisTrain<- iris[dt,]
> irisTrain
    Sepal.Length Sepal.Width Petal.Length Petal.Width
71           5.9         3.2          4.8         1.8
114          5.7         2.5          5.0         2.0
28           5.2         3.5          1.5         0.2
74           6.1         2.8          4.7         1.2
48           4.6         3.2          1.4         0.2
> irisTest<-iris[-dt,]
> irisTest
    Sepal.Length Sepal.Width Petal.Length Petal.Width
10           4.9         3.1          1.5         0.1
11           5.4         3.7          1.5         0.2
12           4.8         3.4          1.6         0.2
15           5.8         4.0          1.2         0.2
16           5.7         4.4          1.5         0.4
29           5.2         3.4          1.4         0.2
```

- ✓ training and testing (consider 70% - 30%)

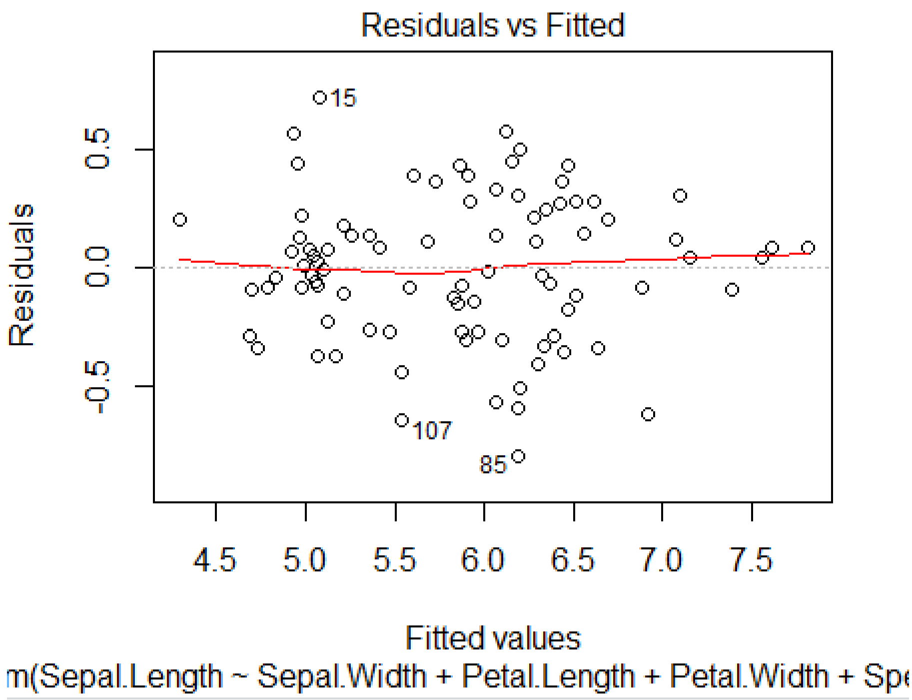
```
> dt=sample(nrow(iris), nrow(iris)*.7)
> irisTrain<- iris[dt,]
> irisTrain
    Sepal.Length Sepal.Width Petal.Length Petal.Width
73           6.3         2.5          4.9         1.5
91           5.5         2.6          4.4         1.2
33           5.2         4.1          1.5         0.1
24           5.1         3.3          1.7         0.5
```

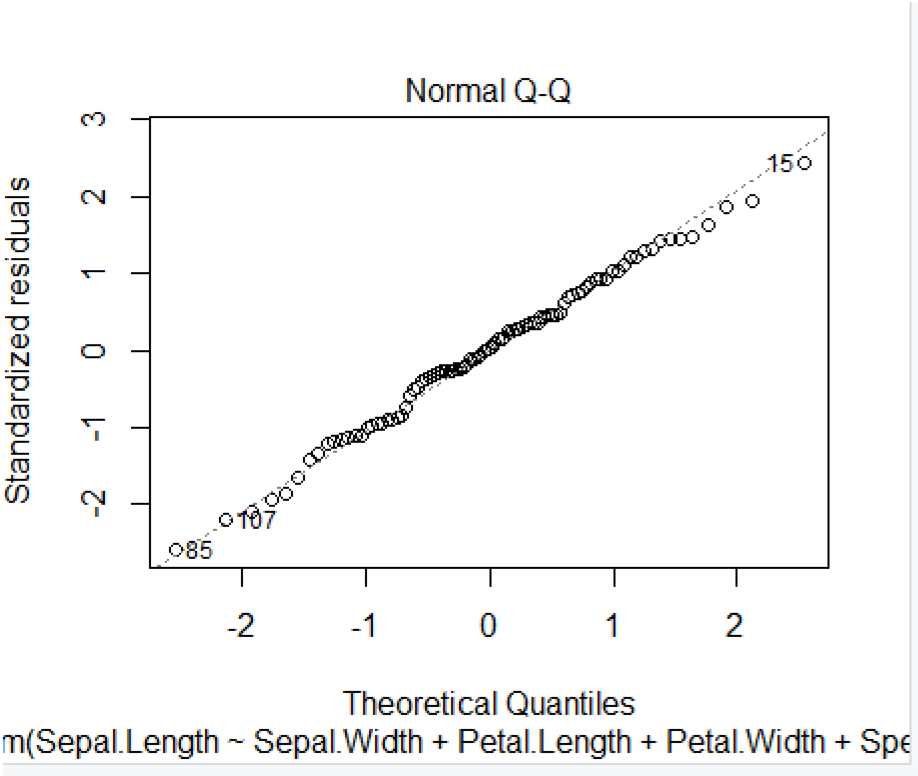
```
47             5.1         3.8          1.6         0.2
50             5.0         3.3          1.4         0.2
92             6.1         3.0          4.6         1.4
58             4.9         2.4          3.3         1.0
51             7.0         3.2          4.7         1.4
```

- ✓ training and testing (consider 60% - 40%)

```
> dt=sample(nrow(iris), nrow(iris)*.6)
> irisTrain<- iris[dt,]
> irisTrain
    Sepal.Length Sepal.Width Petal.Length Petal.Width
20           5.1         3.8          1.5         0.3
90           5.5         2.5          4.0         1.3
150          5.9         3.0          5.1         1.8
56           5.7         2.8          4.5         1.3
86           6.0         3.4          4.5         1.6
140          6.9         3.1          5.4         2.1
93           5.8         2.6          4.0         1.2
```

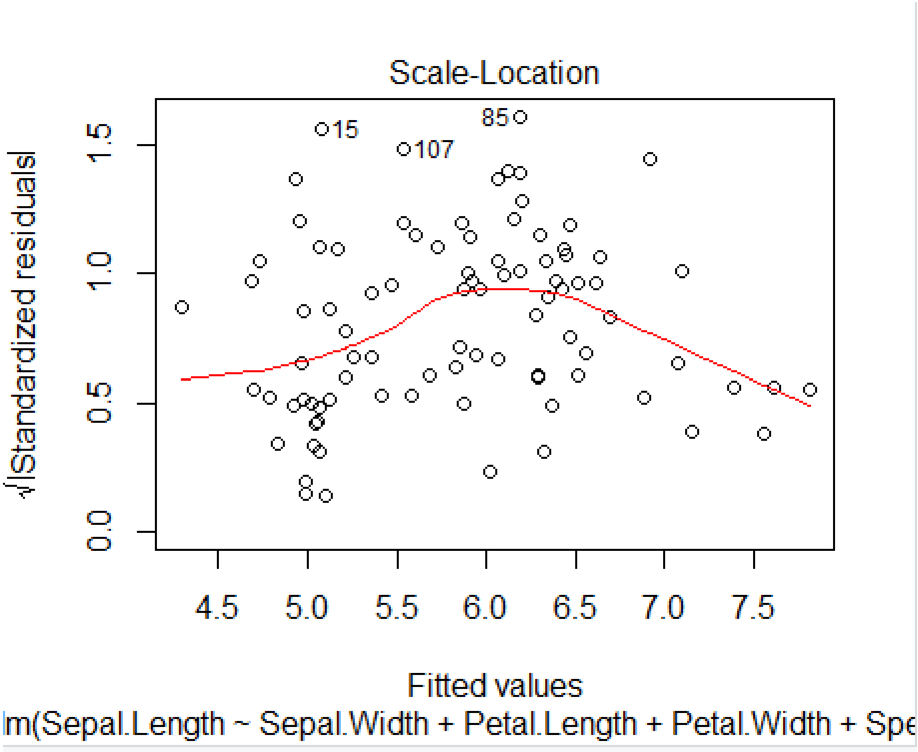
- ✓ display the graph of multiple linear regression



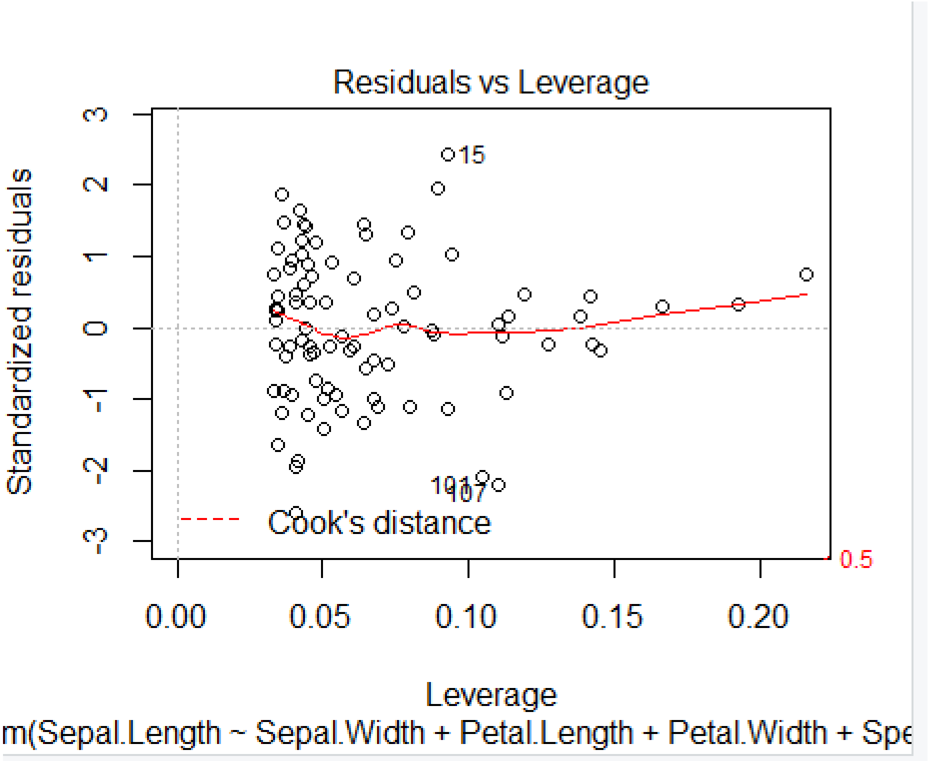


✓ Develop the multiple linear regression and fit the model with train and test data. Find Coefficients & equations with all split data(in case of train & test)

```
> mod=lm(formula = Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width
+ Species, data = irisTrain)
> mod

Call:
lm(formula = Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width +
    Species, data = irisTrain)

Coefficients:
      (Intercept)        Sepal.Width       Petal.Length
           2.1047             0.4861             0.9490
      Petal.Width  Speciesversicolor   Speciesvirginica
          -0.5522            -0.8133            -1.1081
```



Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5
15
85
107
Fitted values
lm(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + Spe



Residuals vs Leverage
Standardized residuals
-3
-2
-1
0
1
2
3
0.00
0.05
0.10
0.15
0.20
15
101
107
Cook's distance
0.5
Leverage
lm(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + Spe

- ✓ Find new equation value with different value of attributes(explain the results in 3 to 4 sentences)

```
> mod1=lm(formula =  Sepal.Width ~Sepal.Length + Petal.Length + Petal.Widt
h+ Species, data = irisTrain)
> mod1

Call:
lm(formula = Sepal.Width ~ Sepal.Length + Petal.Length + Petal.Width +
    Species, data = irisTrain)

Coefficients:
      (Intercept)       Sepal.Length       Petal.Length
           1.7001             0.3576            -0.1419
      Petal.Width  Speciesversicolor   Speciesvirginica
           0.6345            -1.2982            -1.5821

> mod1=lm(formula = Petal.Length~ Sepal.Width+Sepal.Length  + Petal.Width+
Species, data = irisTrain)
> mod1

Call:
lm(formula = Petal.Length ~ Sepal.Width + Sepal.Length + Petal.Width +
    Species, data = irisTrain)

Coefficients:
      (Intercept)        Sepal.Width       Sepal.Length
          -1.1466            -0.1149             0.5653
      Petal.Width  Speciesversicolor   Speciesvirginica
           0.6552             1.5328             1.9689


> mod1=lm(formula = Petal.Width~Petal.Length+ Sepal.Width+Sepal.Length  +
Species, data = irisTrain)
> mod1

Call:
lm(formula = Petal.Width ~ Petal.Length + Sepal.Width + Sepal.Length +
    Species, data = irisTrain)

Coefficients:
      (Intercept)       Petal.Length        Sepal.Width
          -0.2318             0.2906             0.2280
     Sepal.Length  Speciesversicolor   Speciesvirginica
          -0.1459             0.5432             0.9361
```

- ✓ display the graph of multiple linear regression(fitted model).Print the summary(what is your observation on summary result : mention in 4 to 5 lines)

```
> summary(mod)

Call:
lm(formula = Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + 
    Species, data = irisTrain)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.79189 -0.21416  0.00893  0.20906  0.72244 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        2.10475    0.36787   5.721 1.58e-07 ***
Sepal.Width        0.48612    0.11563   4.204 6.52e-05 ***
Petal.Length       0.94896    0.09624   9.860 1.11e-15 ***
Petal.Width       -0.55217    0.20352  -2.713  0.00809 ** 
Speciesversicolor -0.81330    0.34273  -2.373  0.01993 *  
Speciesvirginica  -1.10814    0.46786  -2.369  0.02015 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3116 on 84 degrees of freedom
Multiple R-squared:  0.8637,	Adjusted R-squared:  0.8555 
F-statistic: 106.4 on 5 and 84 DF,  p-value: < 2.2e-16
```

- ✓ Which of train and test split gives you best result (higher value of R2 & other metrics)

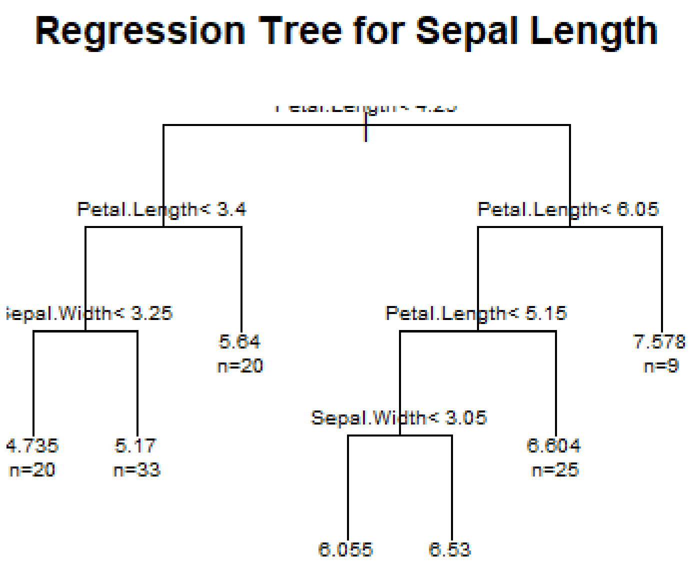
60-40 split gives the best

```
Adjusted R-squared:  0.8884
```

Which is highest among the others

DECISION TREE

```
> fit <- rpart(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width+ Sp
ecies,
+               method="anova", data=iris )
> plot(fit, uniform=TRUE,
+      main="Regression Tree for Sepal Length")
> text(fit, use.n=TRUE, cex = .6)
```



2nd application with airquality data sets

- ✓ **Normalize the data set**

```
normalize <- function(x) {
+    return ((x - min(x,na.rm=TRUE)) / (max(x,na.rm=TRUE) - min(x,na.rm=T
RUE)))
+ }
> dfNorm <- as.data.frame(lapply(airquality, normalize))
> dfNorm
        Ozone      Solar.R       Wind       Temp Month
1   0.23952096 0.559633028 0.30000000 0.26829268  0.00
2   0.20958084 0.339449541 0.33157895 0.39024390  0.00
3   0.06586826 0.434250765 0.57368421 0.43902439  0.00
4   0.10179641 0.935779817 0.51578947 0.14634146  0.00
```

```
5           NA          NA 0.66315789 0.00000000   0.00
6   0.16167665          NA 0.69473684 0.24390244   0.00
7   0.13173653 0.892966361 0.36315789 0.21951220   0.00
8   0.10778443 0.281345566 0.63684211 0.07317073   0.00
9   0.04191617 0.036697248 0.96842105 0.12195122   0.00
```

- ✓ Clean the data (if it is needed)

```
✓ > x <- na.omit(dfNorm)
✓ > x
✓          Ozone     Solar.R       Wind       Temp Month
✓ 1   0.23952096 0.559633028 0.30000000 0.26829268  0.00
✓ 2   0.20958084 0.339449541 0.33157895 0.39024390  0.00
✓ 3   0.06586826 0.434250765 0.57368421 0.43902439  0.00
✓ 4   0.10179641 0.935779817 0.51578947 0.14634146  0.00
✓ 7   0.13173653 0.892966361 0.36315789 0.21951220  0.00
✓ 8   0.10778443 0.281345566 0.63684211 0.07317073  0.00
✓ 9   0.04191617 0.036697248 0.96842105 0.12195122  0.00
✓ 12  0.08982036 0.761467890 0.42105263 0.31707317  0.00
✓ 13  0.05988024 0.865443425 0.39473684 0.24390244  0.00
✓ 14  0.07784431 0.816513761 0.48421053 0.29268293  0.00
```

- ✓ Which is the dependent variable and which are the independent variables of (your choice)

Ozone be the dependent variable and others independent

- ✓ Prepare the data set with training and testing (consider 80% - 20%);

```
> data(airquality)
> dt=sample(nrow(airquality), nrow(airquality)*.8)
> airqualityTrain<- airquality[dt,]
> airqualityTrain
    Ozone Solar.R Wind Temp Month Day
98     66      NA  4.6   87     8   6
15     18      65 13.2   58     5  15
59     NA      98 11.5   80     6  28
46     NA     322 11.5   79     6  15
1      41     190  7.4   67     5   1
104    44     192 11.5   86     8  12
63     49     248  9.2   85     7   2
77     48     260  6.9   81     7  16
```

```
> airqualityTest<-airquality[-dt,]
> airqualityTest
   Ozone Solar.R Wind Temp Month Day
5     NA      NA 14.3   56     5   5
16    14     334 11.5   64     5  16
20    11      44  9.7   62     5  20
25    NA      66 16.6   57     5  25
26    NA     266 14.9   58     5  26
35    NA     186  9.2   84     6   4
37    NA     264 14.3   79     6   6
38    29     127  9.7   82     6   7
40    71     291 13.8   90     6   9
44    23     148  8.0   82     6  13
65    NA     101 10.9   84     7   4
73    10     264 14.3   73     7  12
```

- ✓ training and testing (consider 70% - 30%)

```
> data(airquality)
> dt=sample(nrow(airquality), nrow(airquality)*.7)
> airqualityTrain<- airquality[dt,]
> airqualityTrain
   Ozone Solar.R Wind Temp Month Day
20    11      44  9.7   62     5  20
59    NA      98 11.5   80     6  28
75    NA     291 14.9   91     7  14
29    45     252 14.9   81     5  29
51    13     137 10.3   76     6  20
87    20      81  8.6   82     7  26
13    11     290  9.2   66     5  13

> airqualityTest<-airquality[-dt,]
> airqualityTest
   Ozone Solar.R Wind Temp Month Day
1     41     190  7.4   67     5   1
5     NA      NA 14.3   56     5   5
6     28      NA 14.9   66     5   6
9      8      19 20.1   61     5   9
12    16     256  9.7   69     5  12
16    14     334 11.5   64     5  16
17    34     307 12.0   66     5  17
19    30     322 11.5   68     5  19
21     1       8  9.7   59     5  21
30   115     223  5.7   79     5  30
35    NA     186  9.2   84     6   4
```

- ✓ training and testing (consider 60% - 40%)

```
> data(airquality)
> dt=sample(nrow(airquality), nrow(airquality)*.6)
> airqualityTrain<- airquality[dt,]
> airqualityTrain
    Ozone Solar.R Wind Temp Month Day
16     14     334 11.5   64     5  16
73     10     264 14.3   73     7  12
153    20     223 11.5   68     9  30
109    59      51  6.3   79     8  17
60     NA      31 14.9   77     6  29
120    76     203  9.7   97     8  28
92     59     254  9.2   81     7  31
135    21     259 15.5   76     9  12
```

```
> airqualityTest<-airquality[-dt,]
> airqualityTest
    Ozone Solar.R Wind Temp Month Day
3      12     149 12.6   74     5   3
5      NA      NA 14.3   56     5   5
7      23     299  8.6   65     5   7
9       8      19 20.1   61     5   9
10     NA     194  8.6   69     5  10
11      7      NA  6.9   74     5  11
13     11     290  9.2   66     5  13
14     14     274 10.9   68     5  14
17     34     307 12.0   66     5  17
21      1       8  9.7   59     5  21
25     NA      66 16.6   57     5  25
```

- ✓ Develop the multiple linear regression and fit the model with train and test data. Find Coefficients & equations with all split data.

```
✓ > mod=lm(formula = Ozone ~ Solar.R+ Wind+ Temp +Month+ Day, data = a
  irqualityTrain)
✓ > mod
✓
✓ Call:
✓ lm(formula = Ozone ~ Solar.R + Wind + Temp + Month + Day, data = air
  qualityTrain)
✓
✓ Coefficients:
✓ (Intercept)      Solar.R         Wind         Temp
✓   -36.03393      0.04848     -3.97470      1.52338
✓       Month          Day
✓    -2.19107      0.41959
✓
✓ > Pred <- predict(mod, airqualityTest)
```

```
> Pred
```

```
          3           5           7           9          10
  24.142800          NA   35.282242 -29.256939   37.543647
         11          13          14          17          21
         NA   36.501979   32.435582   27.875407   13.534972
         25          26          27          31          32
 -12.446782    5.950136          NA   65.909566   49.747314
         33          34          36          37          38
  39.749681    1.885668   58.889362   29.646166   46.277138
         40          41          42          43          49
  50.958589   57.501362   66.343050   71.559892   22.618844
         53          54          55          57          61
  71.931366   62.375824   63.747489   54.912814   64.741422
         62          65          69          71          76
  73.758477   39.843820   80.041351   67.477612   22.282092
         82          85          86          88          93
  43.084274   69.781761   67.620375   47.247771   46.849929
         94          95         100         103         106
  16.983458   46.934232   57.062281   42.997279   43.239914
        107         112         115         118         122
  30.472572   41.925862   32.624170   66.984405   91.720257
        127         129         131         132         136
  78.479433   17.580866   36.154258   30.103724   53.500369
        138         142         143         144         147
  18.421711   26.408644   55.503481   12.012471   20.866362
        150         152
  27.440087   46.745562

 >
```

- ✓ Find new equation value with different value of attributes. Print the summary(what is your observation on summary result : mention in 4 to 5 lines)

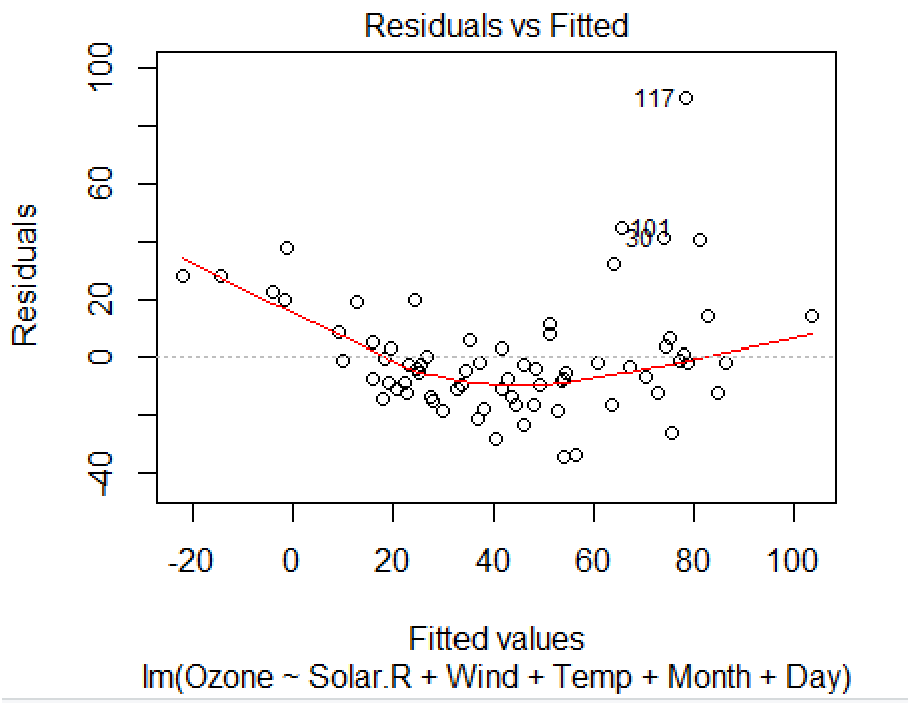
```
> mod=lm(formula = Ozone ~ Solar.R+ Wind+ Temp +Month+ Day, data = airqual
ityTrain)
> summary(mod)

Call:
lm(formula = Ozone ~ Solar.R + Wind + Temp + Month + Day, data = airqualit
yTrain)

Residuals:
    Min      1Q  Median      3Q     Max
-34.200 -11.923  -3.474   5.940  89.653

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -36.03393   30.25536  -1.191  0.23792
Solar.R       0.04848    0.03148   1.540  0.12828
Wind         -3.97470    0.81233  -4.893  6.7e-06 ***
Temp          1.52338    0.38209   3.987  0.00017 ***
```

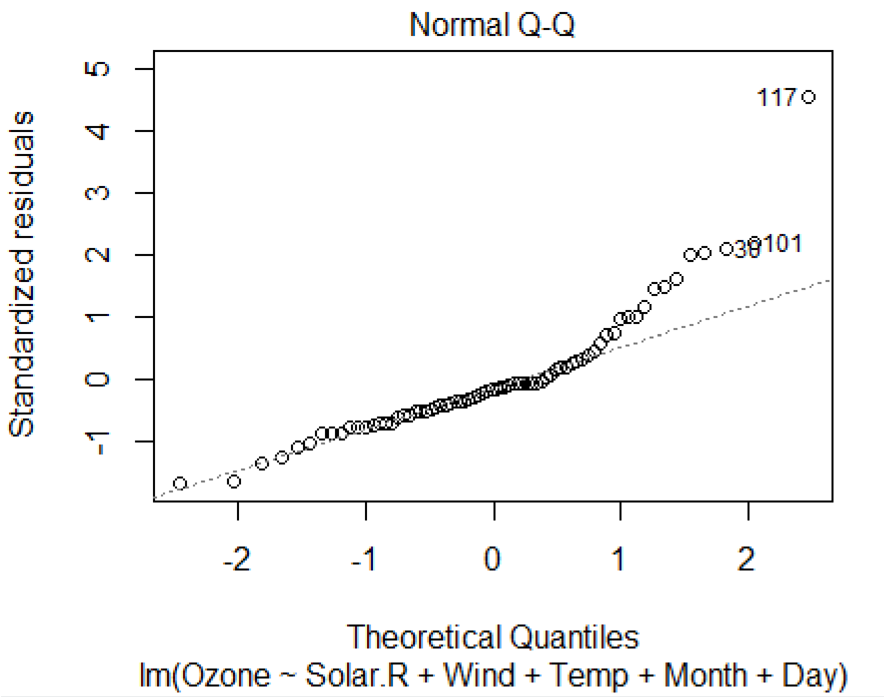
```
Month        -2.19107    1.95069  -1.123  0.26541
Day           0.41959    0.28984   1.448  0.15244
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.82 on 66 degrees of freedom
  (19 observations deleted due to missingness)
Multiple R-squared:  0.6322,   Adjusted R-squared:  0.6043
F-statistic: 22.68 on 5 and 66 DF,  p-value: 3.611e-13

> mod1=lm(formula = Solar.R~Ozone + Wind+ Temp +Month+ Day, data = airqual
ityTrain)
> summary(mod1)

Call:
lm(formula = Solar.R ~ Ozone + Wind + Temp + Month + Day, data = airqualit
yTrain)

Residuals:
     Min       1Q   Median       3Q      Max 
-147.899  -63.645   -0.563   68.657  178.191 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)  
(Intercept) -114.1186   116.6364  -0.978   0.3314  
Ozone          0.7156     0.4646   1.540   0.1283  
Wind           8.0487     3.5059   2.296   0.0249 *
Temp           4.0181     1.5586   2.578   0.0122 *
Month        -13.3143     7.3859  -1.803   0.0760 .
Day           -1.4525     1.1168  -1.301   0.1979  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 79.98 on 66 degrees of freedom
  (19 observations deleted due to missingness)
Multiple R-squared:  0.2346,   Adjusted R-squared:  0.1766
F-statistic: 4.045 on 5 and 66 DF,  p-value: 0.002907

> mod2=lm(formula = Wind~Solar.R+Ozone + Temp +Month+ Day, data = airquali
tyTrain)
> summary(mod2)

Call:
lm(formula = Wind ~ Solar.R + Ozone + Temp + Month + Day, data = airqualit
yTrain)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.3758 -2.0339 -0.4254  1.2798  8.9832 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 16.845950   3.384536   4.977 4.89e-06 ***
Solar.R      0.009188   0.004002   2.296   0.0249 *  
Ozone       -0.066970   0.013687  -4.893 6.70e-06 ***
Temp        -0.092835   0.054053  -1.717   0.0906 .  
Month        0.073683   0.255455   0.288   0.7739    
Day          0.057731   0.037548   1.538   0.1290    
```

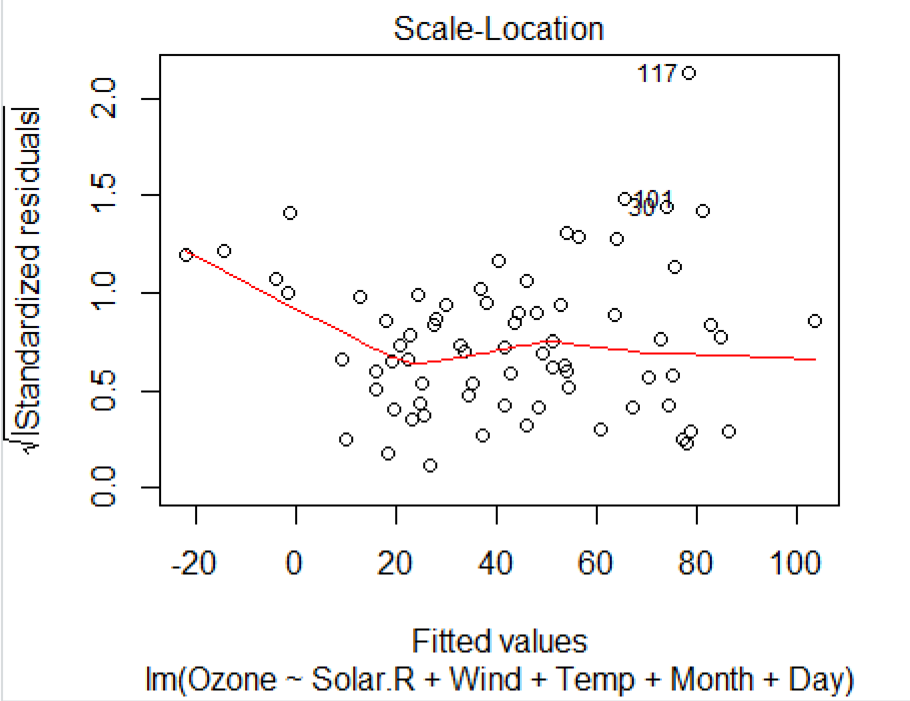
```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.702 on 66 degrees of freedom
  (19 observations deleted due to missingness)
Multiple R-squared:  0.527,    Adjusted R-squared:  0.4912
F-statistic: 14.71 on 5 and 66 DF,  p-value: 1.117e-09

> mod2=lm(formula =  Temp ~Wind+Solar.R+Ozone ++Month+ Day, data = airqual
ityTrain)
> summary(mod2)

Call:
lm(formula = Temp ~ Wind + Solar.R + Ozone + +Month + Day, data = airquali
tyTrain)

Residuals:
     Min       1Q   Median       3Q      Max 
-18.2915  -3.6529   0.5891   3.8535  13.2141 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 56.894353   5.400038  10.536 8.90e-16 ***
Wind        -0.460823   0.268315  -1.717  0.09058 .  
Solar.R      0.022769   0.008832   2.578  0.01218 *  
Ozone        0.127411   0.031957   3.987  0.00017 ***
Month        2.229772   0.499006   4.468 3.17e-05 ***
Day         -0.027927   0.085072  -0.328  0.74375    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.021 on 66 degrees of freedom
  (19 observations deleted due to missingness)
Multiple R-squared:  0.6326,   Adjusted R-squared:  0.6047 
F-statistic: 22.72 on 5 and 66 DF,  p-value: 3.486e-13

> mod

Call:
lm(formula = Ozone ~ Solar.R + Wind + Temp + Month + Day, data = airqualit
yTrain)

Coefficients:
(Intercept)      Solar.R         Wind         Temp  
  -36.03393      0.04848     -3.97470      1.52338  
      Month          Day  
   -2.19107      0.41959  

> mod1

Call:
lm(formula = Solar.R ~ Ozone + Wind + Temp + Month + Day, data = airqualit
yTrain)

Coefficients:
(Intercept)        Ozone         Wind         Temp  
  -114.1186       0.7156       8.0487       4.0181  
      Month          Day  
```

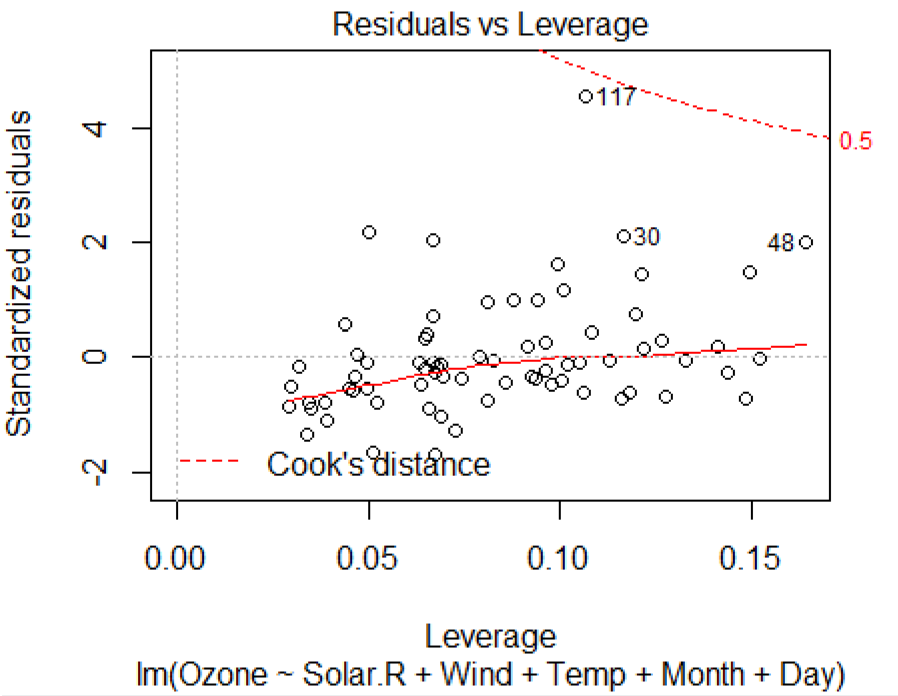
```
   -13.3143      -1.4525

> mod2

Call:
lm(formula = Temp ~ Wind + Solar.R + Ozone + +Month + Day, data = airquali
tyTrain)

Coefficients:
(Intercept)         Wind      Solar.R        Ozone
   56.89435     -0.46082      0.02277      0.12741
      Month          Day
    2.22977     -0.02793
```

- ✓ display the graph of multiple linear regression



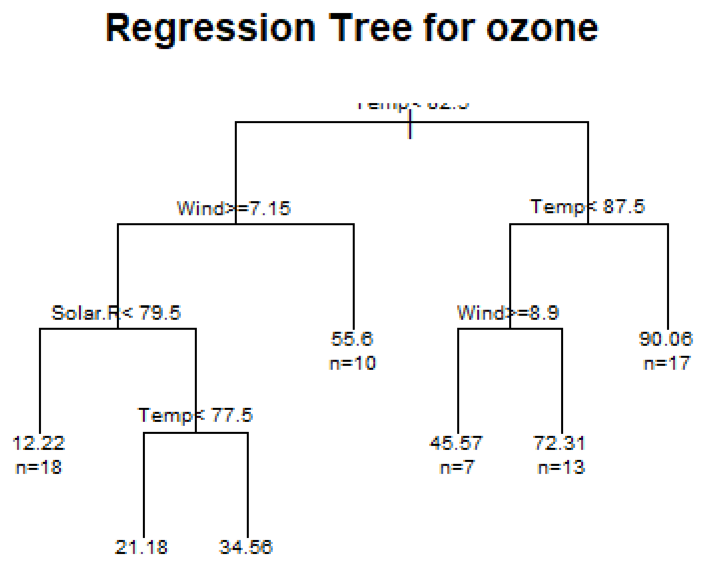

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Ozone ~ Solar.R + Wind + Temp + Month + Day)
117
30
101


Scale-Location
√|Standardized residuals|
Fitted values
lm(Ozone ~ Solar.R + Wind + Temp + Month + Day)
117
101
30



DECISION TREE

```
> fit <- rpart(Ozone ~ Solar.R+ Wind+ Temp +Month+ Day,
+              method="anova", data=airquality )
> plot(fit, uniform=TRUE,
+      main="Regression Tree for Sepal Length")
> text(fit, use.n=TRUE, cex = .6)
> plot(fit, uniform=TRUE,
+      main="Regression Tree for ozone")
> text(fit, use.n=TRUE, cex = .6)
```



Decision tree (regression)

Reg_No

Lab Slot:

`1st application with VADeaths data sets`

- ✓ **Normalize the data set**

Formula/definition of normalization is as below,

[

If you want to normalize your data, you can do so as you suggest and simply calculate the following:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

where $x = (x_1, \ldots, x_n)$ and z_i is now your i^{th} normalized data. As a proof of concept (although you did not ask for it) here is some `R` code and accompanying graph to illustrate this point:

]

```
> x1=VADeaths[,1]
> x1=(x1-min(x1))/(max(x1)-min(x1))
> x2=VADeaths[,2]
> x2=(x2-min(x2))/(max(x2)-min(x2))
> x3=VADeaths[,3]
> x3=(x3-min(x3))/(max(x3)-min(x3))
> x4=VADeaths[,4]
 > x4=(x4-min(x4))/(max(x4)-min(x4))
 > data=data.frame(x1,x2,x3,x4)
 > data
             x1         x2        x3        x4
50-54 0.0000000 0.00000000 0.0000000 0.0000000
55-59 0.1178637 0.06578947 0.1597846 0.1250000
60-64 0.2799263 0.25438596 0.3877917 0.2620192
65-69 0.5395948 0.48684211 0.7037702 0.6418269
70-74 1.0000000 1.00000000 1.0000000 1.0000000

 >
```

- ✓ Clean the data (if it is needed)

Not required

- ✓ Which is the dependent variable and which are the independent variables of (your choice)

Let rural man be dependent and others independent

- ✓ Prepare the data set with training and testing (consider 80% - 20%);

```
> data(VADeaths)
> dt=sample(nrow(VADeaths), nrow(VADeaths)*.8)
> VADeathsTrain<- VADeaths[dt,]
> VADeathsTrain
      Rural Male Rural Female Urban Male Urban Female
65-69       41.0         30.9       54.6         35.1
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
50-54       11.7          8.7       15.4          8.4
> VADeathsTest<-VADeaths[-dt,]
> VADeathsTest
  Rural Male Rural Female   Urban Male Urban Female
        66.0         54.3         71.1         50.0
```

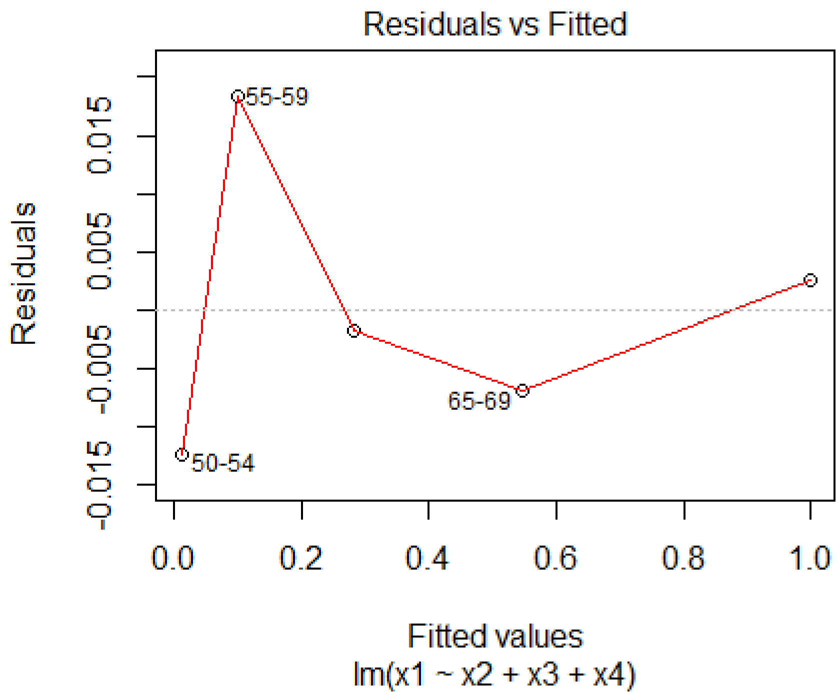
- ✓ training and testing (consider 70% - 30%)

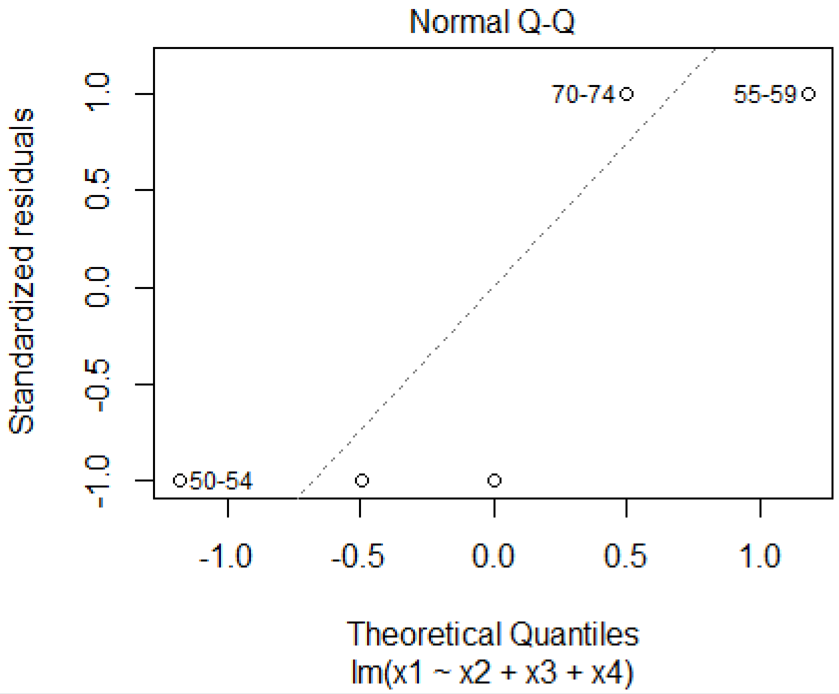
```
> data(VADeaths)
> dt=sample(nrow(VADeaths), nrow(VADeaths)*.7)
> VADeathsTrain<- VADeaths[dt,]
> VADeathsTrain
      Rural Male Rural Female Urban Male Urban Female
60-64       26.9         20.3       37.0         19.3
55-59       18.1         11.7       24.3         13.6
50-54       11.7          8.7       15.4          8.4
> VADeathsTest<-VADeaths[-dt,]
> VADeathsTest
      Rural Male Rural Female Urban Male Urban Female
65-69         41         30.9       54.6         35.1
70-74         66         54.3       71.1         50.0
```

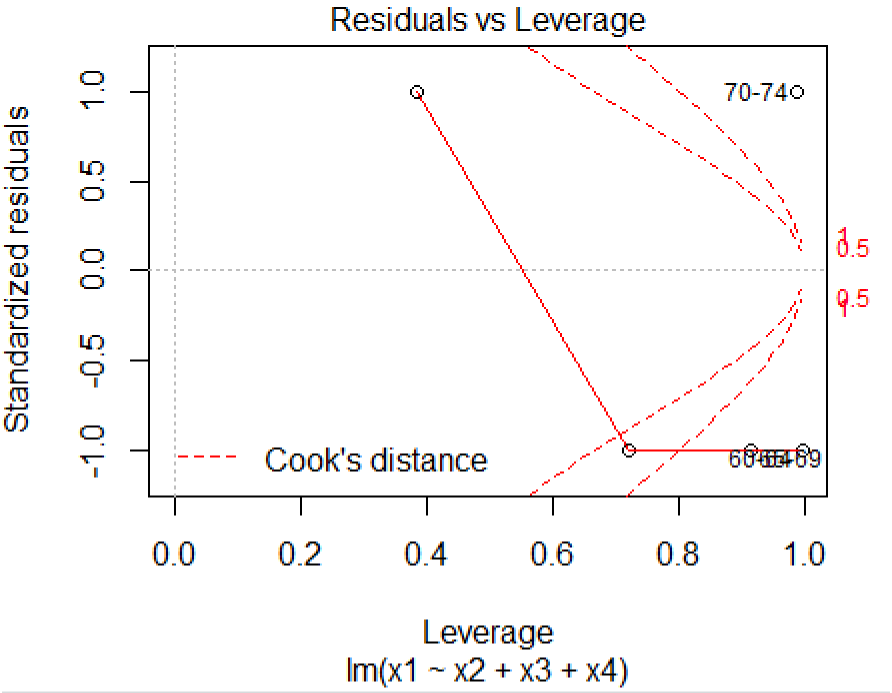
- ✓ training and testing (consider 60% - 40%)

```
✓ > data(VADeaths)
✓ > dt=sample(nrow(VADeaths), nrow(VADeaths)*.6)
✓ > VADeathsTrain<- VADeaths[dt,]
✓ > VADeathsTrain
✓       Rural Male Rural Female Urban Male Urban Female
✓ 55-59       18.1         11.7       24.3         13.6
✓ 50-54       11.7          8.7       15.4          8.4
✓ 60-64       26.9         20.3       37.0         19.3
✓ > VADeathsTest<-VADeaths[-dt,]
✓ > VADeathsTest
✓       Rural Male Rural Female Urban Male Urban Female
✓ 65-69         41         30.9       54.6         35.1
✓ 70-74         66         54.3       71.1         50.0
```

- ✓ display the graph of multiple linear regression(explain in 4 to 5 sentences)


Residuals vs Fitted
Residuals
0.015
0.005
-0.005
-0.015
0.0
0.2
0.4
0.6
0.8
1.0
55-59
65-69
50-54
Fitted values
lm(x1 ~ x2 + x3 + x4)


Normal Q-Q
Standardized residuals
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
70-74
55-59
50-54
Theoretical Quantiles
lm(x1 ~ x2 + x3 + x4)



✓ Develop the multiple linear regression and fit the model with train and test data. Find Coefficients & equations with all split data(explain your outputs in 6 to 7 sentences)

```
> mod1=lm(formula =  x1~x2+x3+x4, data = dataTrain)
> Pred <- predict(mod1, dataTest)
Warning message:
In predict.lm(mod1, dataTest) :
  prediction from a rank-deficient fit may be misleading
> Pred
      50-54       55-59
-0.01604839  0.07722308

>
```

✓ Find new equation value with different value of attributes.

```
> mod2=lm(formula =  x2~x1+x3+x4, data = dataTrain)
> mod3=lm(formula =  x3~x1+x2+x4, data = dataTrain)
> mod4=lm(formula =  x4~x1+x2+x3, data = dataTrain)
> mod2

Call:
lm(formula = x2 ~ x1 + x3 + x4, data = dataTrain)

Coefficients:
(Intercept)           x1           x3           x4
    0.02184      1.36072     -0.38256           NA

> mod3

Call:
lm(formula = x3 ~ x1 + x2 + x4, data = dataTrain)

Coefficients:
(Intercept)           x1           x2           x4
    0.05708      3.55689     -2.61397           NA

> mod4

Call:
lm(formula = x4 ~ x1 + x2 + x3, data = dataTrain)

Coefficients:
(Intercept)           x1           x2           x3
    -0.1356       4.2568      -3.1212           NA
```

- ✓ display the graph of multiple linear regression. Print the summary(what is your observation on summary result : mention in 4 to 5 lines)

```
> summary(mod1)

Call:
lm(formula = x1 ~ x2 + x3 + x4, data = dataTrain)

Residuals:
ALL 3 residuals are 0: no residual degrees of freedom!

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.01605         NA      NA       NA
x2           0.73490         NA      NA       NA
x3           0.28114         NA      NA       NA
x4                NA         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:    NaN 
F-statistic:   NaN on 2 and 0 DF,  p-value: NA
```

- ✓ Which of train and test split gives you best result (higher value of R2)

60-40 gives us the best result with highest R2 value

UPLOAD Your Programs On THE FOLLOWING LINK
https://drive.google.com/drive/folders/1W8XYL0j5aBYkG9n4bJD6yrkgHHdBac3K?usp=sharing