ASSIGNMENT 3

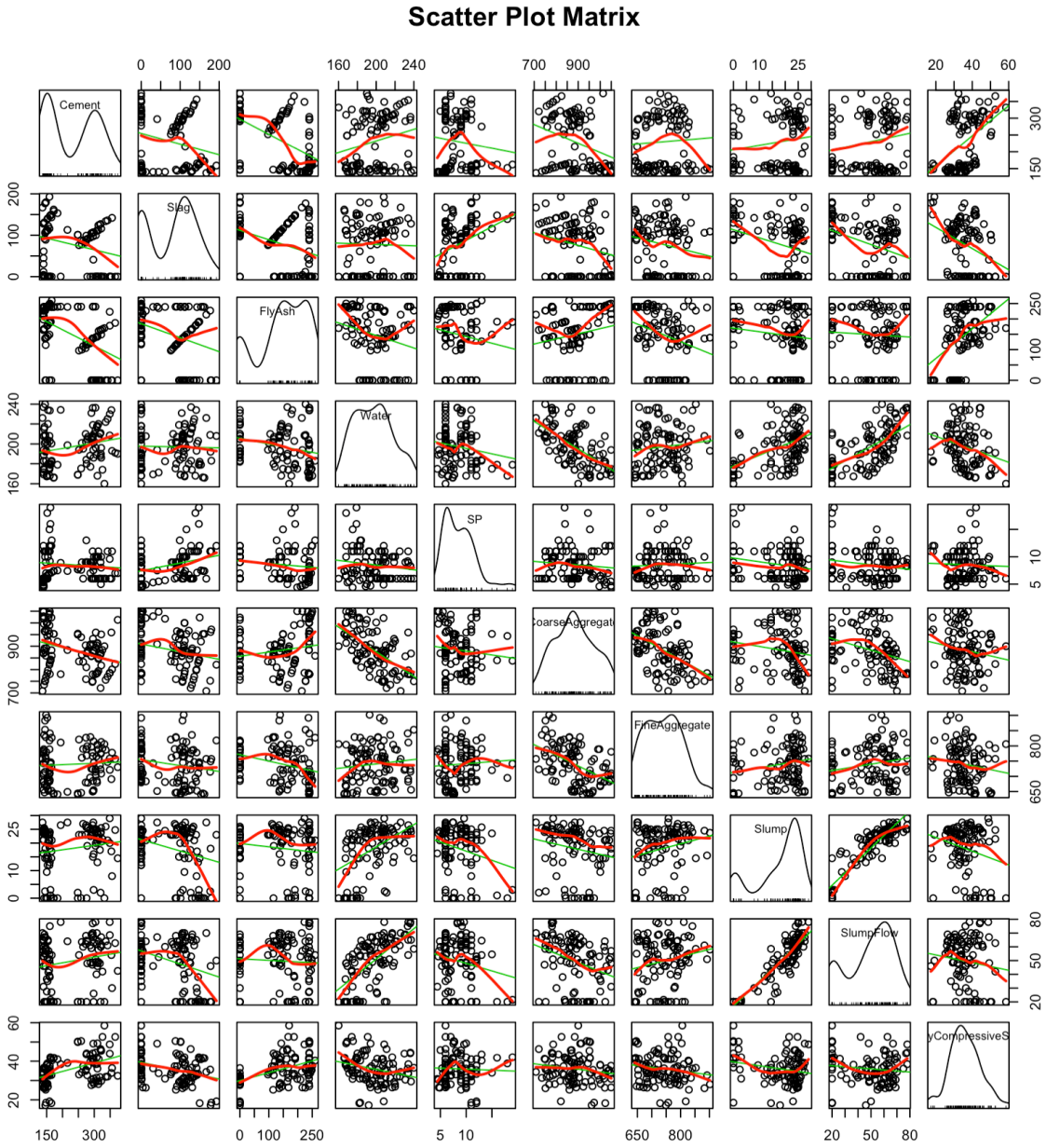
PROBLEM 1

Concrete_Slump_Test_Data <- read_excel("Concrete Slump Test Data.xlsx")
Concrete_Slump_Test_Data

Question 1.1:

Scatterplot:

conc <- Concrete_Slump_Test_Data[,-1]
scatterplotMatrix(conc,spread = FALSE,lty.smooth=2,main="Scatter Plot Matrix")



Interpretation:
We selected Slump flow as the Output variable, as we see that we can find clear relations among Slump flow and other parameters when compared to the other 2 output variables. We choose all the 7 input variables for our model, as they all have a linear relationship with Slump flow.

Question 1.2:
Building different regression models:
Multiple linear regression:
fit_MLR<-
lm(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate,data=conc)
summary(fit_MLR)

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate, data = conc)

Residuals:
    Min      1Q  Median      3Q     Max
-30.880 -10.428   1.815   9.601  22.953

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -252.87467  350.06649  -0.722   0.4718
Cement             0.05364    0.11236   0.477   0.6342
Slag              -0.00569    0.15638  -0.036   0.9710
FlyAsh             0.06115    0.11402   0.536   0.5930
Water              0.73180    0.35282   2.074   0.0408 *
SP                 0.29833    0.66263   0.450   0.6536
CoarseAggregate    0.07366    0.13510   0.545   0.5869
FineAggregate      0.09402    0.14191   0.663   0.5092
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.84 on 95 degrees of freedom
Multiple R-squared:  0.5022,    Adjusted R-squared:  0.4656
F-statistic: 13.69 on 7 and 95 DF,  p-value: 3.915e-12
```

Interpretation:
The coefficients show the effect of a predictor variable over the outcome variable, keeping the other input variables constant. Altogether, the predictor variables account for 50% of the total variance.

Multiple linear regression with interactions:

```
fit_LRI<-
lm(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate+FineAggregat
e:FlyAsh+Cement:Water,data=conc)
summary(fit_LRI)
```

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + FineAggregate:FlyAsh +
    Cement:Water, data = conc)

Residuals:
     Min       1Q   Median       3Q      Max
-29.1753  -8.3803   0.1527   8.9835  22.8191

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)           -2.547e+02  3.712e+02  -0.686   0.4943
Cement                 8.718e-02  2.324e-01   0.375   0.7084
Slag                  -4.822e-02  1.667e-01  -0.289   0.7731
FlyAsh                 3.315e-01  2.592e-01   1.279   0.2040
Water                  7.657e-01  4.548e-01   1.684   0.0956 .
SP                     3.797e-01  6.932e-01   0.548   0.5852
CoarseAggregate        4.949e-02  1.425e-01   0.347   0.7292
FineAggregate          1.313e-01  1.469e-01   0.894   0.3738
FlyAsh:FineAggregate  -3.964e-04  3.490e-04  -1.136   0.2589
Cement:Water          -2.912e-04  8.899e-04  -0.327   0.7442
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.88 on 93 degrees of freedom
Multiple R-squared:  0.5096,    Adjusted R-squared:  0.4621
F-statistic: 10.74 on 9 and 93 DF,  p-value: 2.88e-11
```

Interpretation:
We create interactions between variables with similar linear relationship. We see that the residual error has increased, this is not a better model.

Second order Regression- multiple linear regression with interactions between every pair of variables

```
fit_SLR<-lm(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate+

Cement:Slag+Cement:FlyAsh+Cement:Water+Cement:SP+Cement:CoarseAggregate+Cement:Fi
neAggregate+ Slag:FlyAsh+Slag:Water+Slag:SP+Slag:CoarseAggregate+Slag:FineAggregate+
FlyAsh:Water+FlyAsh:SP+FlyAsh:CoarseAggregate+FlyAsh:FineAggregate+
Water:SP+Water:CoarseAggregate+Water:FineAggregate+
      SP:CoarseAggregate+SP:FineAggregate+
```

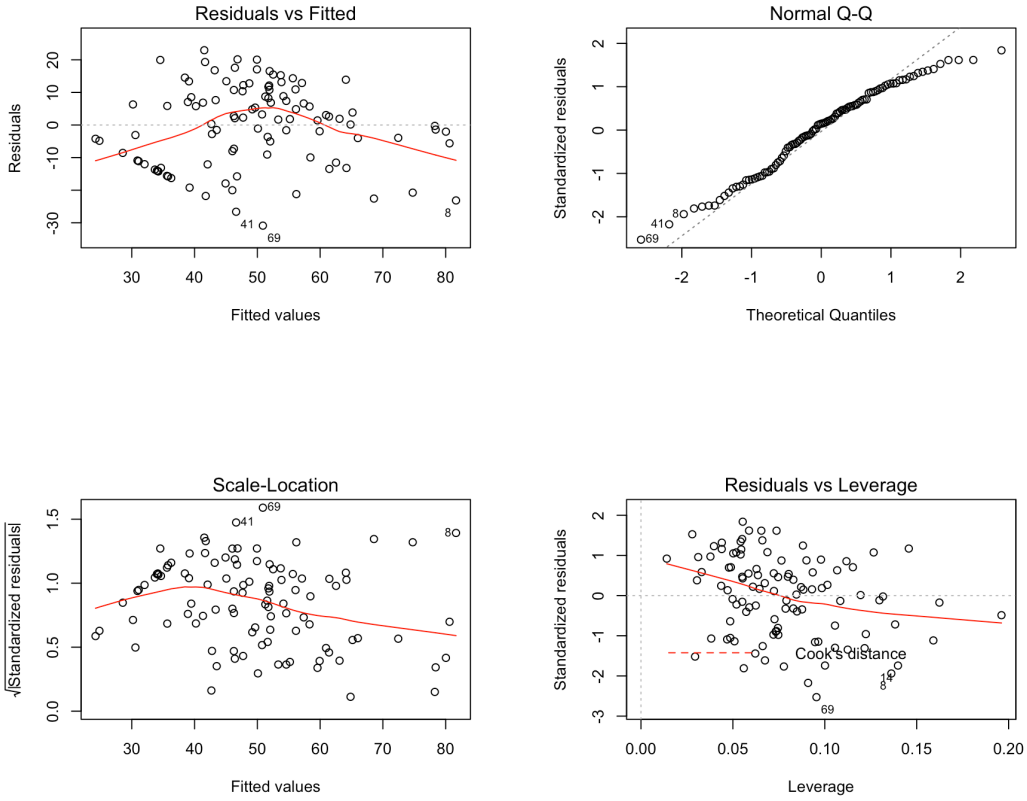
CoarseAggregate:FineAggregate, data = conc)
summary(fit_SLR)

```
    FlyAsh:FineAggregate + Water:SP + Water:CoarseAggregate +
    Water:FineAggregate + SP:CoarseAggregate + SP:FineAggregate +
    CoarseAggregate:FineAggregate, data = conc)

Residuals:
     Min       1Q   Median       3Q      Max
-23.8222  -6.0751   0.2499   4.7302  21.2758

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  1.567e+03  1.277e+03   1.227 0.223715
Cement                      -1.638e+00  1.387e+00  -1.181 0.241227
Slag                        -5.560e+00  1.495e+00  -3.719 0.000386 ***
FlyAsh                      -3.498e+00  1.162e+00  -3.010 0.003568 **
Water                       -6.165e+00  2.778e+00  -2.219 0.029543 *
SP                          -9.203e+01  1.474e+02  -0.624 0.534359
CoarseAggregate             -5.943e-01  5.978e-01  -0.994 0.323325
FineAggregate               -9.309e-01  7.902e-01  -1.178 0.242545
Cement:Slag                 -2.639e-04  5.594e-04  -0.472 0.638511
Cement:FlyAsh                3.774e-04  4.528e-04   0.834 0.407198
Cement:Water                 4.472e-03  2.183e-03   2.049 0.044004 *
Cement:SP                    4.826e-02  5.069e-02   0.952 0.344250
Cement:CoarseAggregate       5.554e-04  5.822e-04   0.954 0.343217
Cement:FineAggregate         3.448e-04  6.659e-04   0.518 0.606098
Slag:FlyAsh                  9.259e-04  4.603e-04   2.011 0.047927 *
Slag:Water                   1.246e-02  2.541e-03   4.903 5.44e-06 ***
Slag:SP                      4.740e-02  7.788e-02   0.609 0.544640
Slag:CoarseAggregate         1.928e-03  5.389e-04   3.577 0.000618 ***
Slag:FineAggregate           1.972e-03  7.217e-04   2.732 0.007860 **
FlyAsh:Water                 5.582e-03  1.770e-03   3.153 0.002331 **
FlyAsh:SP                    4.320e-02  5.692e-02   0.759 0.450241
FlyAsh:CoarseAggregate       1.428e-03  4.753e-04   3.005 0.003624 **
FlyAsh:FineAggregate         1.433e-03  5.691e-04   2.519 0.013940 *
Water:SP                     5.024e-02  1.347e-01   0.373 0.710204
Water:CoarseAggregate        2.135e-03  1.191e-03   1.793 0.077110 .
Water:FineAggregate          4.104e-03  1.841e-03   2.229 0.028857 *
SP:CoarseAggregate           3.877e-02  5.893e-02   0.658 0.512625
SP:FineAggregate             3.905e-02  6.008e-02   0.650 0.517680
CoarseAggregate:FineAggregate -1.164e-04  4.208e-04  -0.276 0.782943
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.27 on 74 degrees of freedom
Multiple R-squared:  0.7519,    Adjusted R-squared:  0.658
F-statistic:  8.01 on 28 and 74 DF,  p-value: 3.907e-13
```

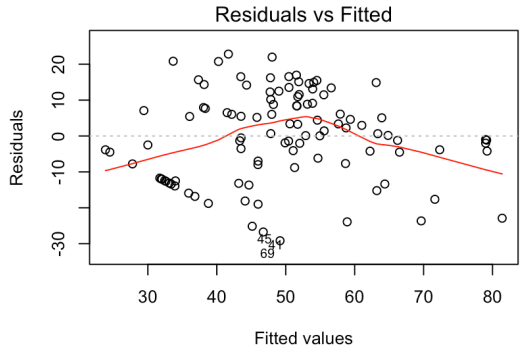
Interpretation:
The predictive power of the model has improved. Adjusted R squared value has increased to 0.65. Altogether, the predictor variables account for 75% of the total variance.

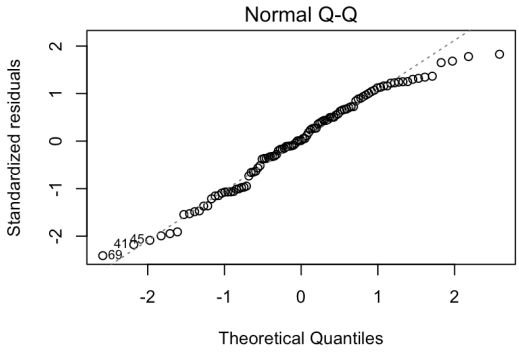
Question 1.3:

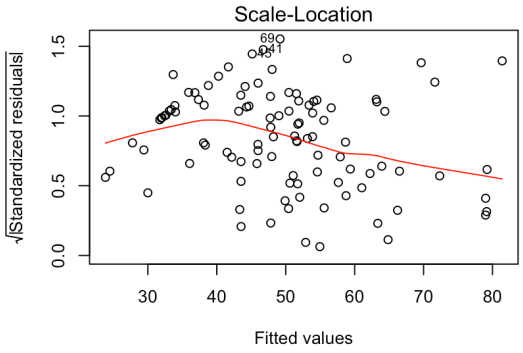
Regression diagnostics:
Typical approach:
par(mfrow=c(2,2))
plot(fit_MLR)

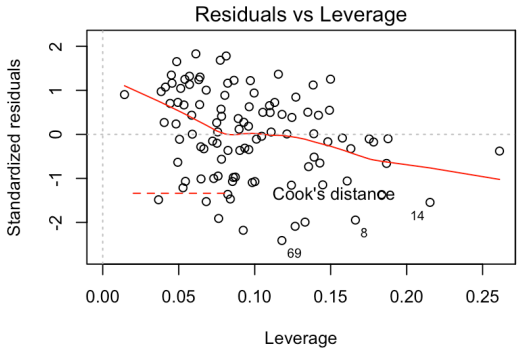


plot(fit_LRI)

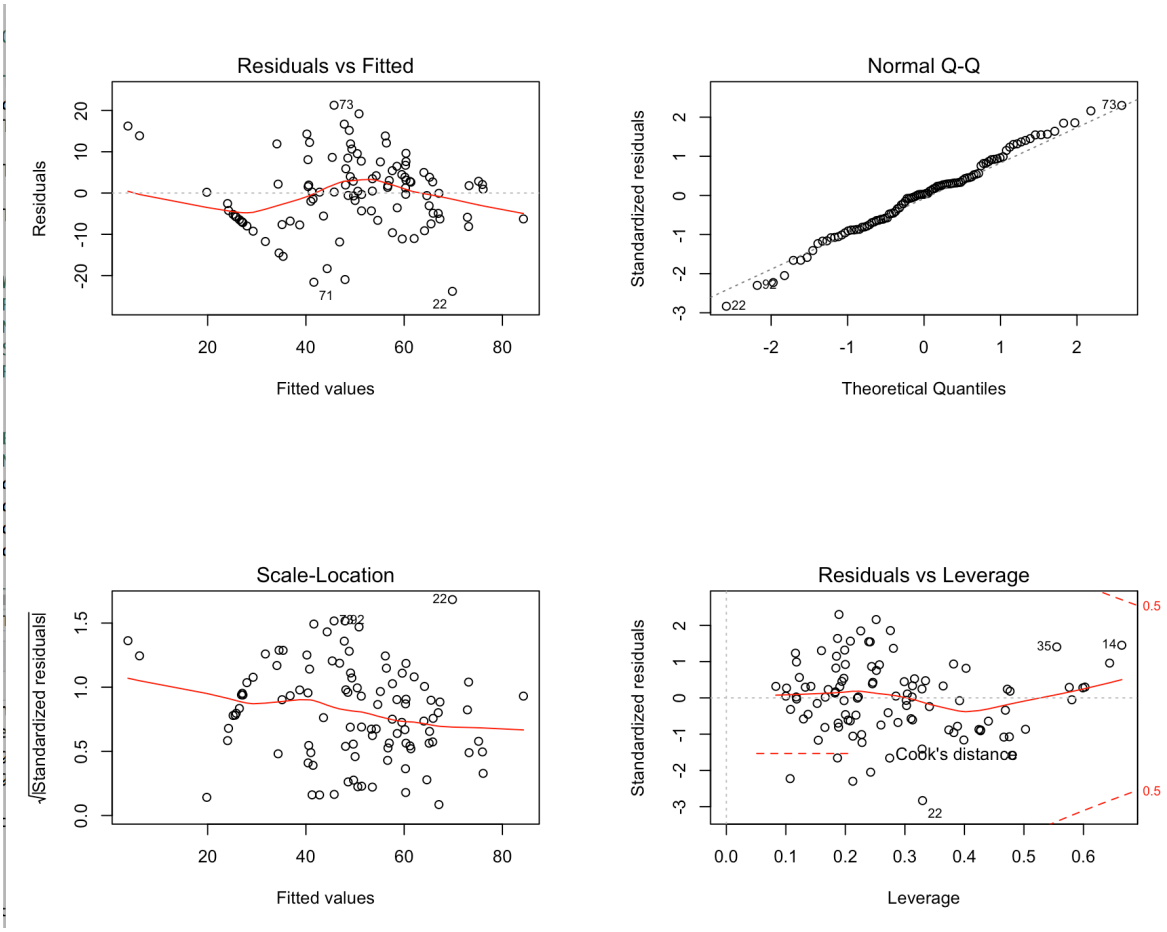

Residuals vs Fitted
Residuals
Fitted values
30
40
50
60
70
80
-30
-10
0
10
20
45
44
69


Normal Q-Q
Standardized residuals
Theoretical Quantiles
-2
-1
0
1
2
41
45
69


Scale-Location
√|Standardized residuals|
Fitted values
30
40
50
60
70
80
0.0
0.5
1.0
1.5
69
41
45


Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
-2
-1
0
1
2
14
8
69

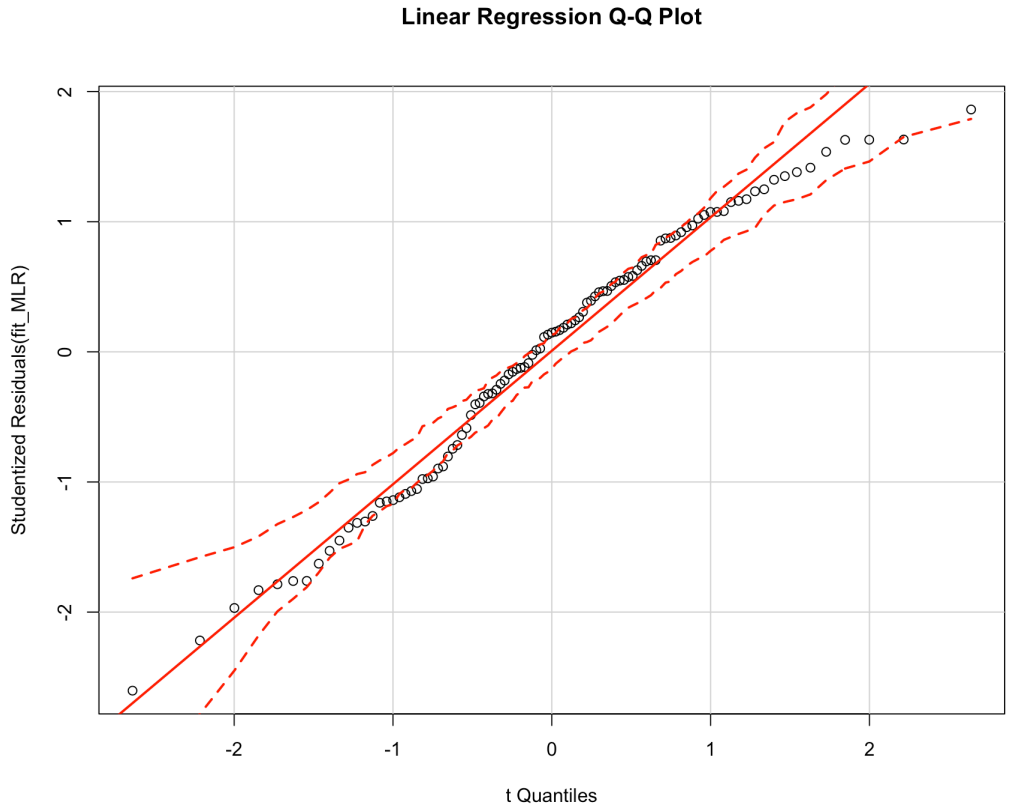
plot(fit_SLR)



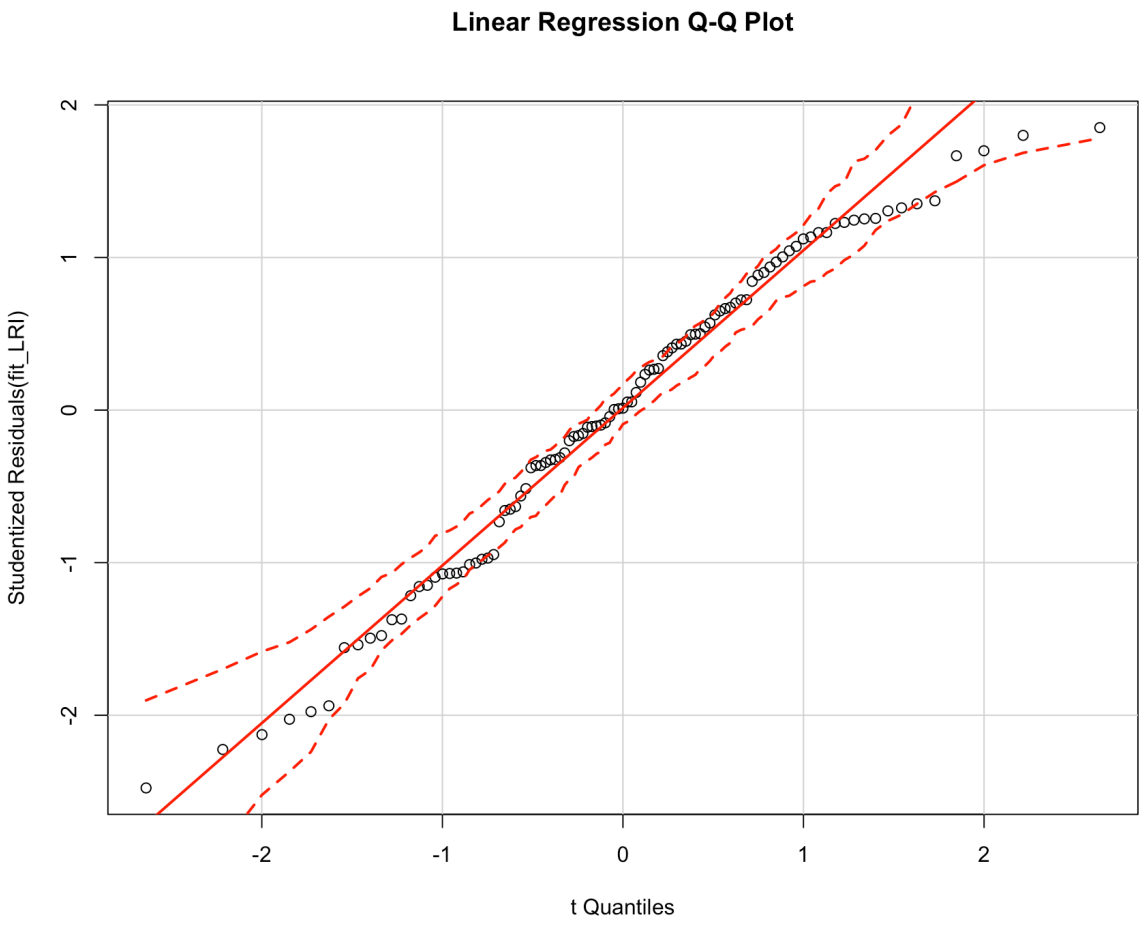
Interpretation:
We observe that the resulting graphs are generally similar for all the three cases. Residual-vs-Fitted represents linearity quality. There should be no systematic relationship between the residuals and the predicted. Normal Q-Q plot represents normality quality. The residual should be normally distributed with the mean “0”. The point on this graph falls on the straight 45-degree line. Scale-Location represents Homoscedasticity. There is constant variance assumption or random band around the horizontal line. Residual-vs-Leverage represents information on individual observations that need attention.

Enhanced approach:
Normality:
par(mfrow=c(1,1))

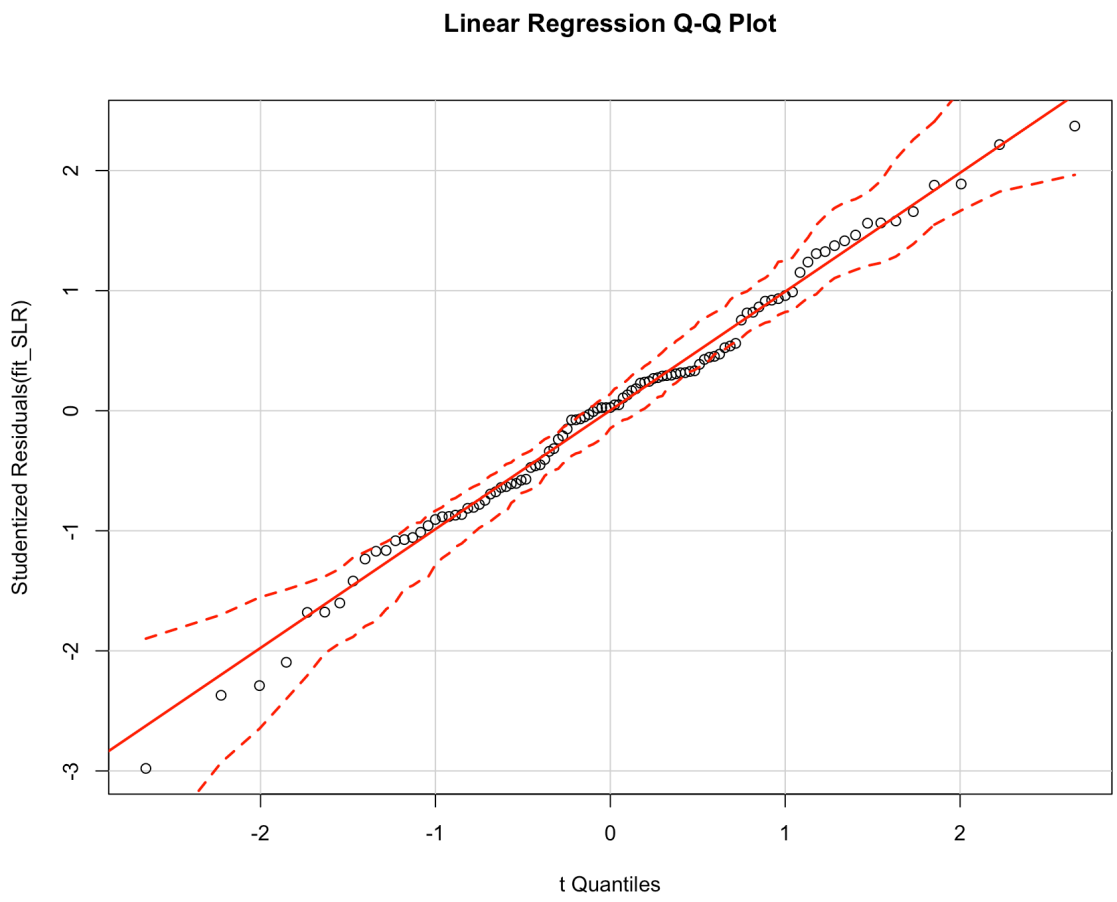
qqPlot(fit_MLR, labels=row.names(conc), id.method="identify", simulate=TRUE,main="Linear Regression Q-Q Plot")



qqPlot(fit_LRI, labels=row.names(conc), id.method="identify", simulate=TRUE,main="Linear Regression Q-Q Plot")



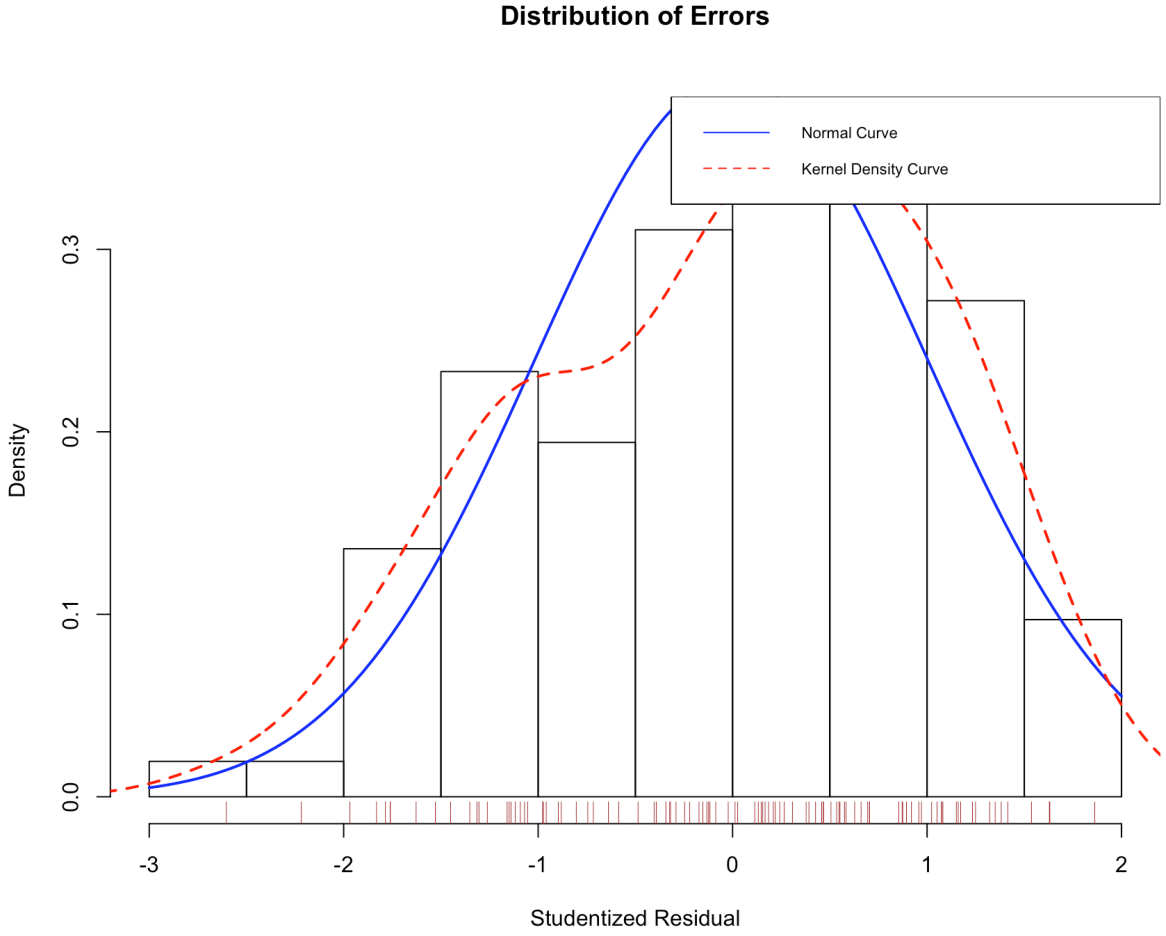
qqPlot(fit_SLR, labels=row.names(conc), id.method="identify", simulate=TRUE,main="Linear Regression Q-Q Plot")



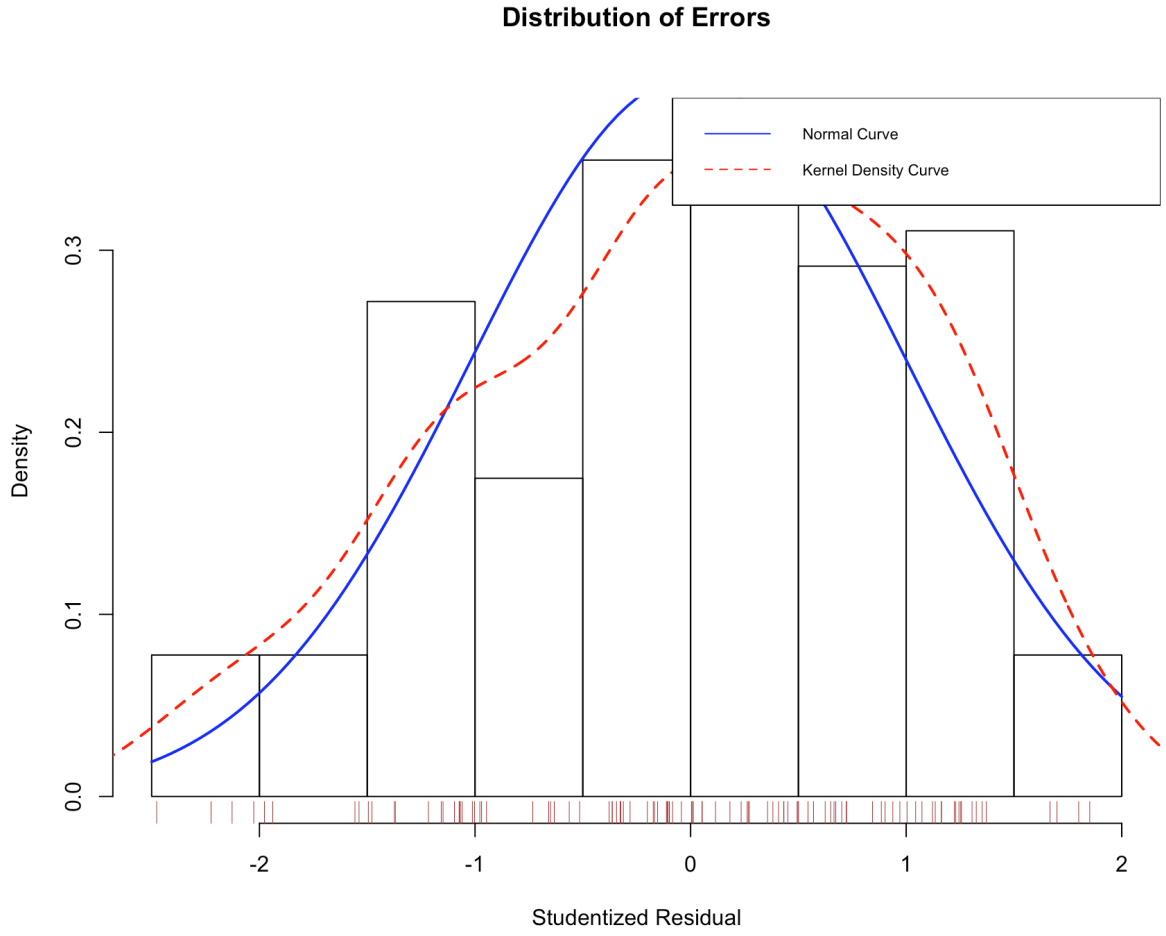
Interpretation:
The qqPlot() function provides a more accurate method for assessing the normality assumption. In the case of all 3 linear regression models, all points fall close to the line and are within the confidence envelope, suggesting that the normality condition has been met.

Plotting Studentized residuals:

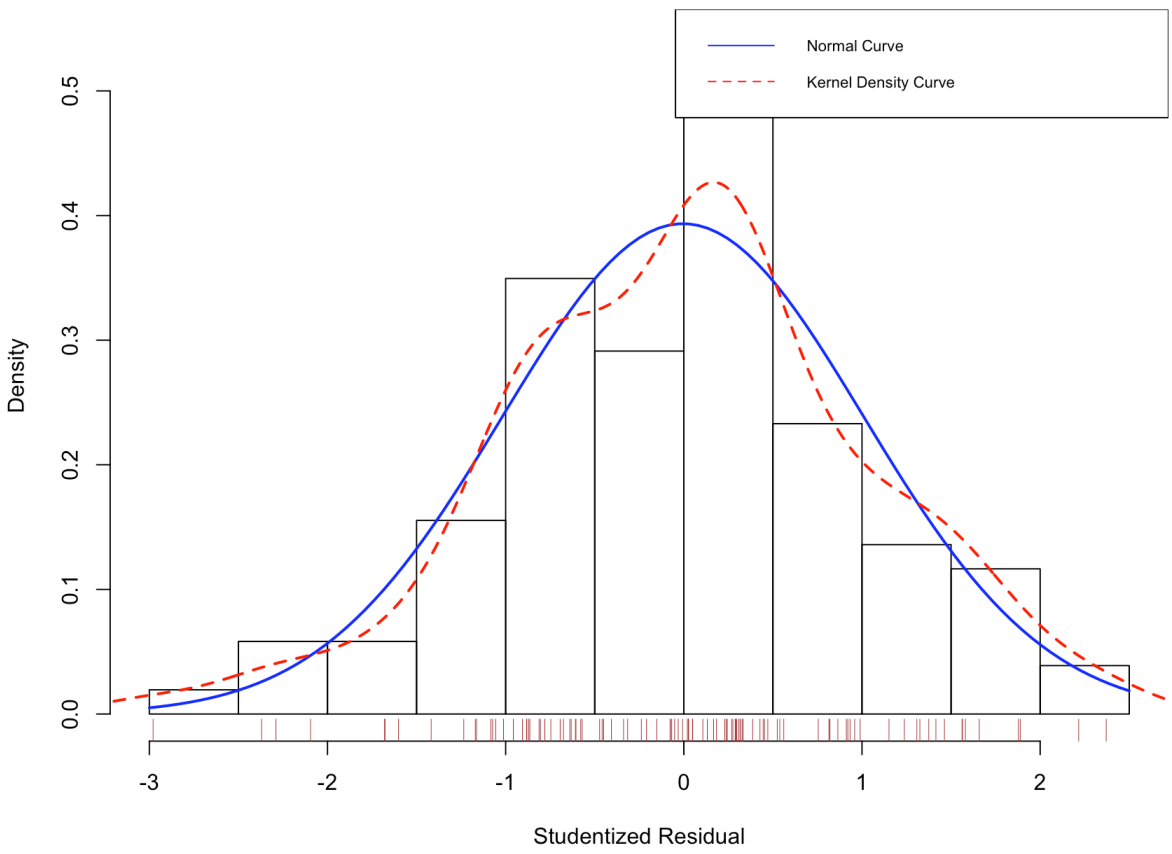
```
residualPlot = function(fit, nbreaks = 10){
  z = rstudent(fit)
  hist(z, breaks = nbreaks, freq = FALSE, xlab = "Studentized Residual", main = "Distribution of
Errors")
  rug(jitter(z), col="brown")
  curve(dnorm(x, mean=mean(z), sd=sd(z)), add=TRUE, col="blue", lwd=2)
  lines(density(z)$x, density(z)$y, col="red", lwd=2, lty=2)
  legend("topright", legend = c("Normal Curve", "Kernel Density Curve"), lty = 1:2, col=c("blue",
"red"), cex=.7 )
}

residualPlot(fit_MLR)
```



residualPlot(fit_LRI)



residualPlot(fit_SLR)



Interpretation:
All three models are consistent with respect to the normal distribution.

Independence of Errors- Durbin-Watson Test

durbinWatsonTest(fit_MLR)

```
 lag Autocorrelation D-W Statistic p-value
   1     -0.01249995      2.009189   0.858
 Alternative hypothesis: rho != 0
```

durbinWatsonTest(fit_LRI)

```
 lag Autocorrelation D-W Statistic p-value
   1     -0.01237993      2.005339   0.768
 Alternative hypothesis: rho != 0
```

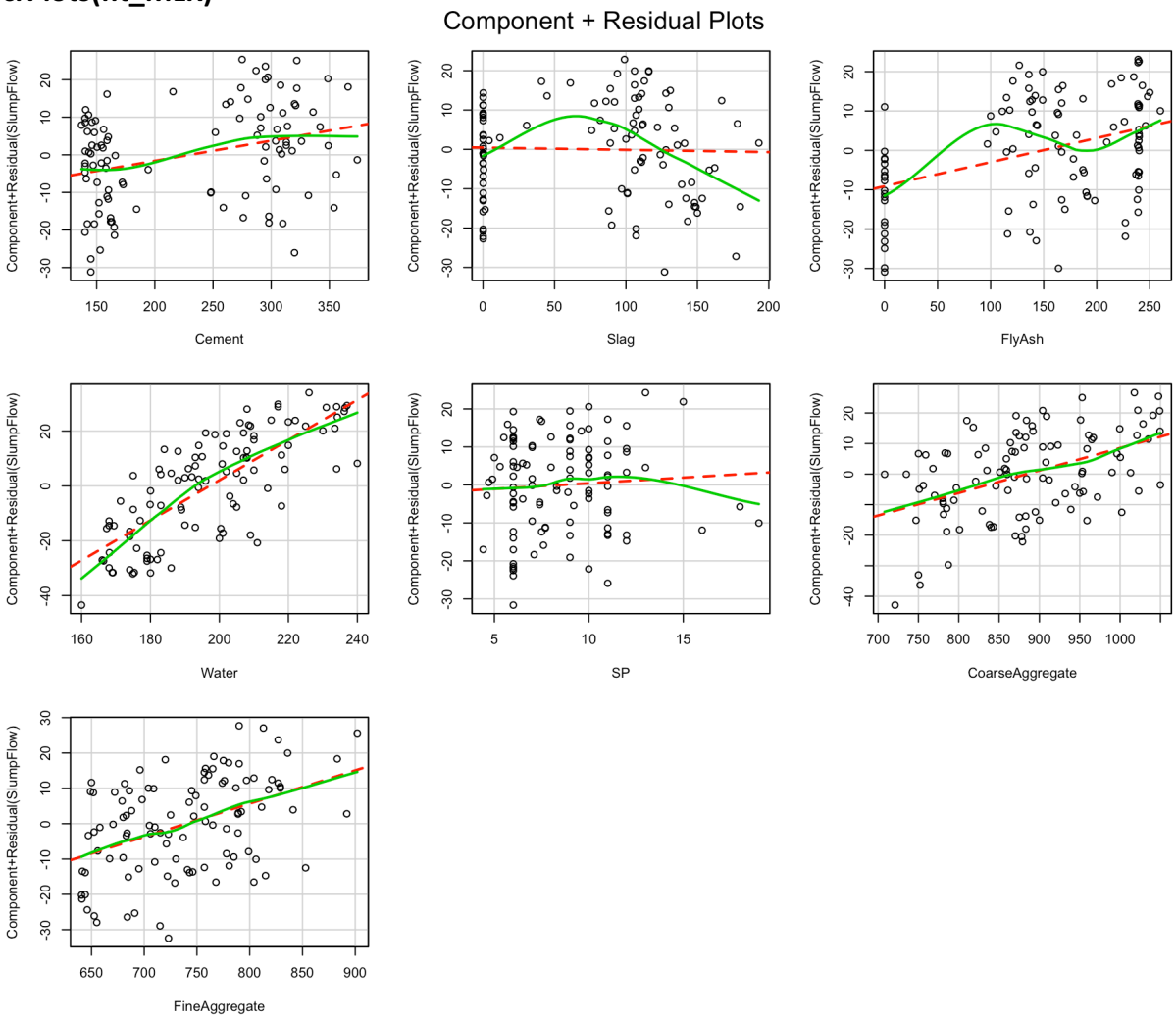
durbinWatsonTest(fit_SLR)

```
 lag Autocorrelation D-W Statistic p-value
   1     -0.03799301       2.06973   0.536
 Alternative hypothesis: rho != 0
```

Interpretation:
The durbinWatsonTest checks the residuals for autocorrelation. When the p-value is < 0.5, the residuals are significantly correlated whereas p > 0.05 provides no evidence of correlation. All 3 models are significantly correlated since p>0.5.

Linearity- Component plus residual plots

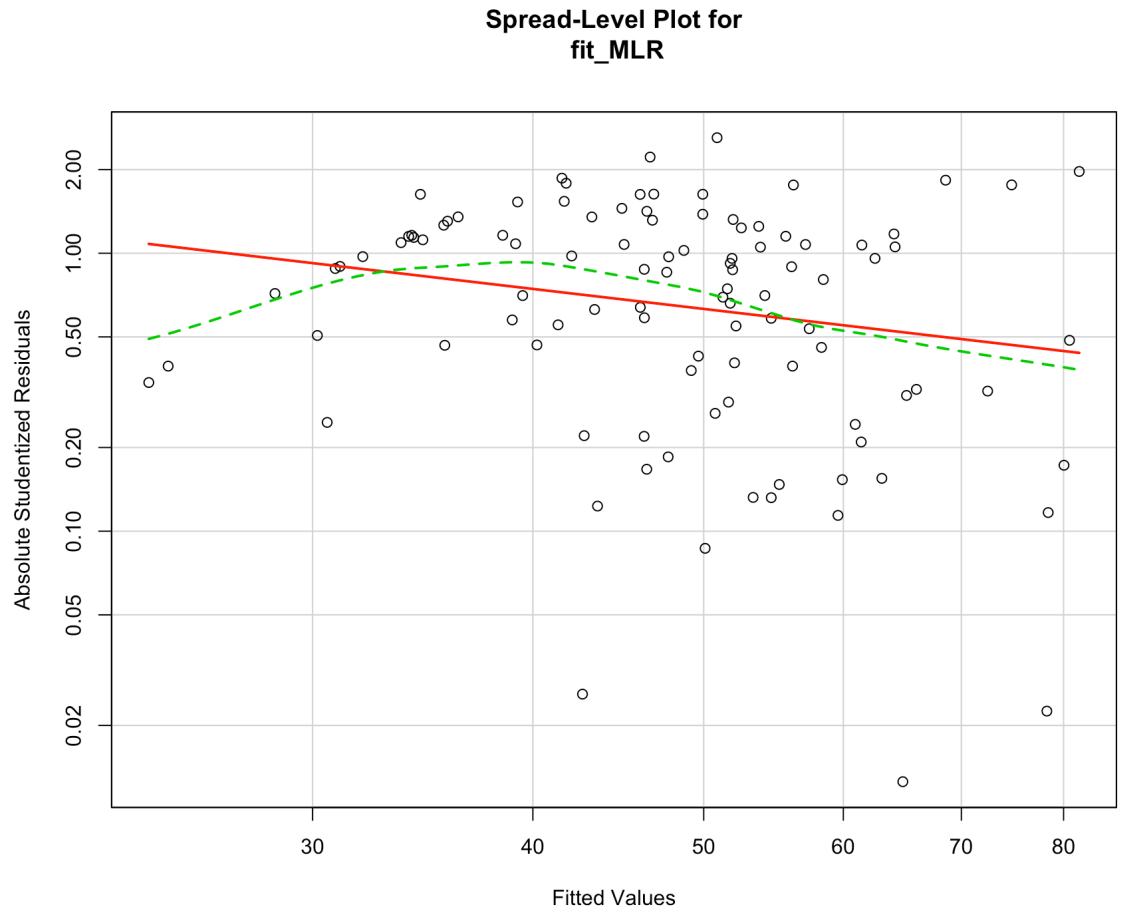
crPlots(fit_MLR)



Interpretation:
Any nonlinearity in any of the above graphs suggest that we may have not adequately modeled the functional form of that predictor in the regression. From above graphs, we can confirm that we have met the linearity assumption.

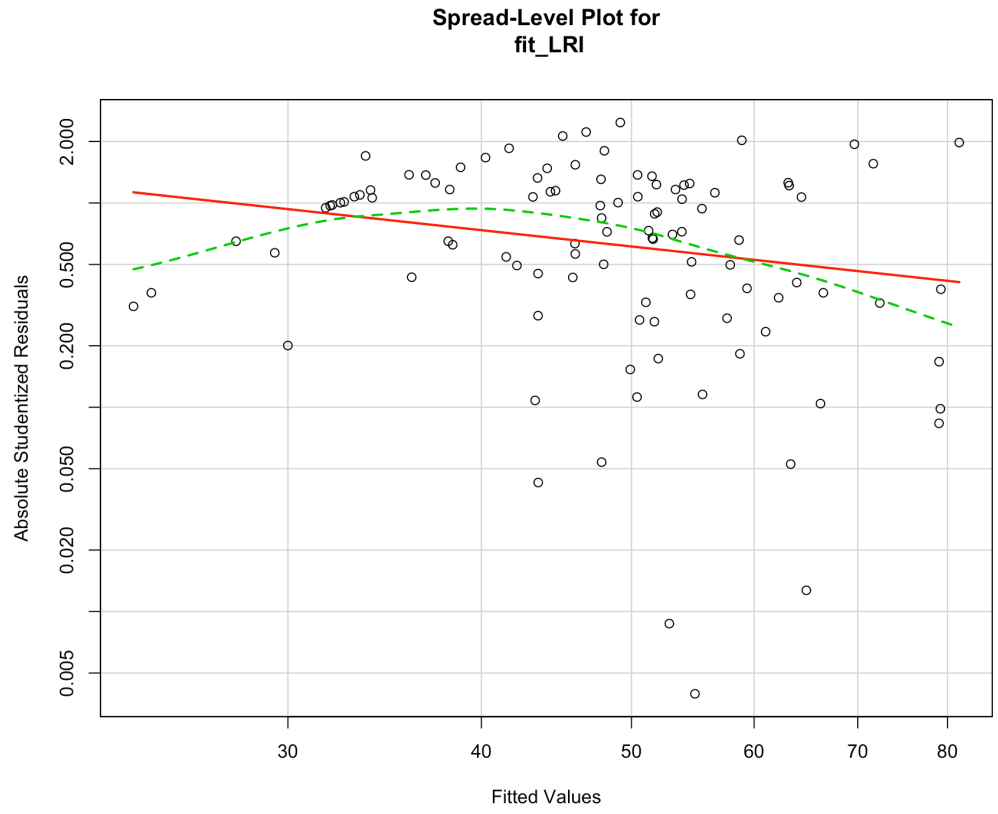
Homoscedasticity:
ncvTest(fit_MLR)
spreadLevelPlot(fit_MLR)

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.2327094    Df = 1     p = 0.6295221
```



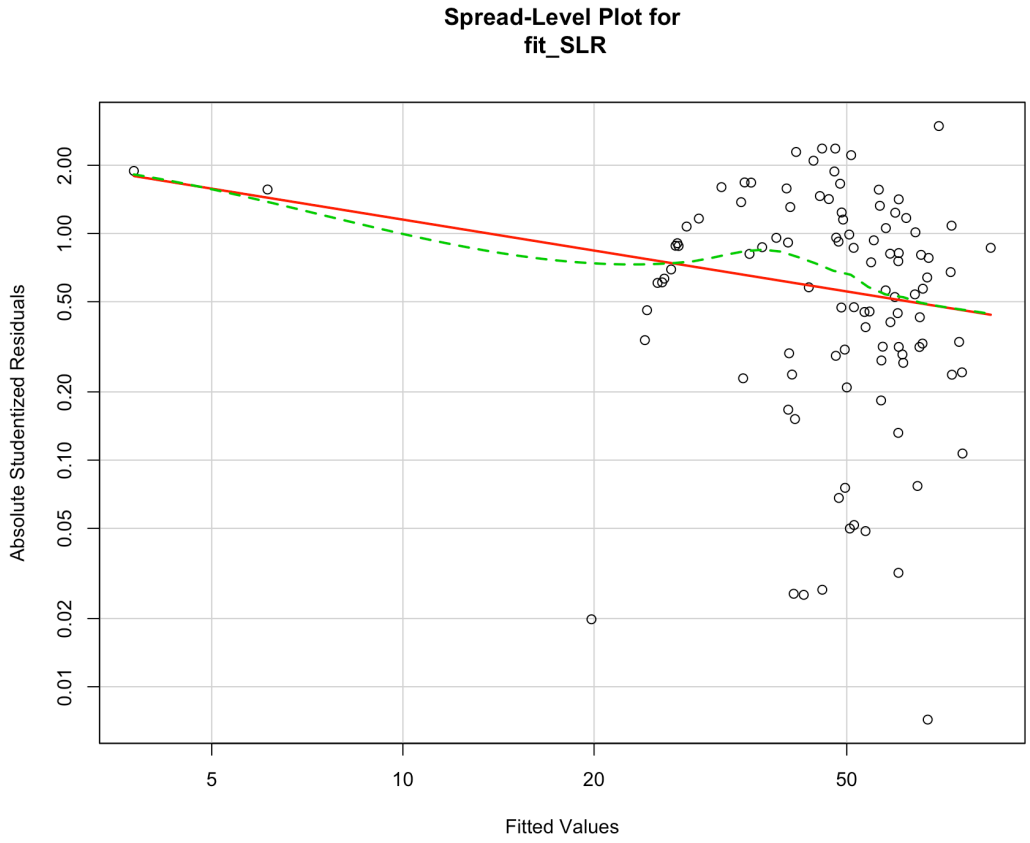
ncvTest(fit_LRI)
spreadLevelPlot(fit_LRI)

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.344344    Df = 1     p = 0.5573325
```



ncvTest(fit_SLR)
spreadLevelPlot(fit_SLR)

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.698219    Df = 1     p = 0.1004604
```



Interpretation:

All 3 models met the constant variance assumption with p = 0.6295221, 0.5573325, 0.1004604, respectively. Also, in all three graphs above, there are random points about the horizontal best fit line. If we had violated the assumption, we would see non-horizontal line.

Global validation of linear model assumption:

library(gvlma)
gvlma(fit_MLR)

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate, data = conc)

Coefficients:
    (Intercept)          Cement            Slag           FlyAsh           Water              SP  CoarseAggregate
     -252.87467         0.05364         -0.00569         0.06115         0.73180         0.29833         0.07366
  FineAggregate
        0.09402


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = fit_MLR)

                     Value   p-value                   Decision
Global Stat        21.919 2.080e-04 Assumptions NOT satisfied!
Skewness            1.703 1.919e-01    Assumptions acceptable.
Kurtosis            2.382 1.228e-01    Assumptions acceptable.
Link Function      16.433 5.041e-05 Assumptions NOT satisfied!
Heteroscedasticity  1.401 2.365e-01    Assumptions acceptable.
```

gvlma(fit_LRI)

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + FineAggregate:FlyAsh +
    Cement:Water, data = conc)

Coefficients:
         (Intercept)               Cement                 Slag               FlyAsh                Water
          -2.547e+02            8.718e-02           -4.822e-02            3.315e-01            7.657e-01
                  SP       CoarseAggregate         FineAggregate  FlyAsh:FineAggregate          Cement:Water
           3.797e-01            4.949e-02            1.313e-01           -3.964e-04           -2.912e-04


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = fit_LRI)

                     Value   p-value                   Decision
Global Stat        19.954 0.0005099 Assumptions NOT satisfied!
Skewness            1.550 0.2130830    Assumptions acceptable.
Kurtosis            1.938 0.1639018    Assumptions acceptable.
Link Function      15.123 0.0001008 Assumptions NOT satisfied!
Heteroscedasticity  1.343 0.2464232    Assumptions acceptable.
```

gvlma(fit_SLR)

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + FineAggregate:FlyAsh +
    Cement:Water, data = conc)

Coefficients:
         (Intercept)               Cement                 Slag               FlyAsh                Water
          -2.547e+02            8.718e-02           -4.822e-02            3.315e-01            7.657e-01
                  SP      CoarseAggregate         FineAggregate  FlyAsh:FineAggregate         Cement:Water
           3.797e-01            4.949e-02            1.313e-01           -3.964e-04           -2.912e-04


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = fit_LRI)

                     Value   p-value                   Decision
Global Stat         19.954 0.0005099 Assumptions NOT satisfied!
Skewness             1.550 0.2130830    Assumptions acceptable.
Kurtosis             1.938 0.1639018    Assumptions acceptable.
Link Function       15.123 0.0001008 Assumptions NOT satisfied!
Heteroscedasticity  1.343 0.2464232    Assumptions acceptable.
```

Interpretation:

The gvlma() function provides a go/no-go test of model assumption. Based on the result ,SecondOrderRegression passes with all test condition. Multiple linear regression with and without interaction did not pass on Link and Global Stat.

Multicollinearity:

vif(fit_MLR)
sqrt(vif(fit_MLR))>2

```
> vif(fit_MLR)
        Cement           Slag         FlyAsh          Water             SP CoarseAggregate   FineAggregate
     48.570807      55.276977      58.649500      31.431899       2.139998       88.171895      49.961057
> sqrt(vif(fit_MLR))>2
        Cement           Slag         FlyAsh          Water             SP CoarseAggregate   FineAggregate
          TRUE           TRUE           TRUE           TRUE          FALSE           TRUE           TRUE
>
```

vif(fit_LRI)
sqrt(vif(fit_LRI))>2

```
> vif(fit_LRI)
            Cement               Slag             FlyAsh              Water                 SP
        206.513249         62.436999        301.062015         51.895857          2.327243
     CoarseAggregate    FineAggregate FlyAsh:FineAggregate       Cement:Water
          97.499276         53.193326        283.166218        146.296537
> sqrt(vif(fit_LRI))>2
            Cement               Slag             FlyAsh              Water                 SP
              TRUE               TRUE               TRUE               TRUE              FALSE
     CoarseAggregate    FineAggregate FlyAsh:FineAggregate       Cement:Water
              TRUE               TRUE               TRUE               TRUE
```

vif(fit_SLR)
sqrt(vif(fit_SLR))>2

```
> vif(fit_SLR)
                     Cement                        Slag                     FlyAsh                       Water
                11565.00375                7893.58287                9518.23932                3046.19919
                         SP           CoarseAggregate              FineAggregate                Cement:Slag
                165534.94606                2697.83403                2420.83716                  58.18439
              Cement:FlyAsh              Cement:Water                 Cement:SP     Cement:CoarseAggregate
                   75.24369                1384.47211                1624.84232                1419.65109
       Cement:FineAggregate                Slag:FlyAsh                Slag:Water                   Slag:SP
                 1598.28606                  28.84437                 903.73609                2656.75400
        Slag:CoarseAggregate       Slag:FineAggregate             FlyAsh:Water                 FlyAsh:SP
                  779.91114                1002.20148                 848.82336                2255.92134
     FlyAsh:CoarseAggregate     FlyAsh:FineAggregate                  Water:SP      Water:CoarseAggregate
                 1422.46976                1184.11584                4891.35661                 339.69445
       Water:FineAggregate        SP:CoarseAggregate          SP:FineAggregate CoarseAggregate:FineAggregate
                 1371.85917                21780.82979               16411.23656                 647.72105
> sqrt(vif(fit_SLR))>2
                     Cement                        Slag                     FlyAsh                       Water
                       TRUE                        TRUE                        TRUE                        TRUE
                         SP           CoarseAggregate              FineAggregate                Cement:Slag
                       TRUE                        TRUE                        TRUE                        TRUE
              Cement:FlyAsh              Cement:Water                 Cement:SP     Cement:CoarseAggregate
                       TRUE                        TRUE                        TRUE                        TRUE
       Cement:FineAggregate                Slag:FlyAsh                Slag:Water                   Slag:SP
                       TRUE                        TRUE                        TRUE                        TRUE
        Slag:CoarseAggregate       Slag:FineAggregate             FlyAsh:Water                 FlyAsh:SP
                       TRUE                        TRUE                        TRUE                        TRUE
     FlyAsh:CoarseAggregate     FlyAsh:FineAggregate                  Water:SP      Water:CoarseAggregate
                       TRUE                        TRUE                        TRUE                        TRUE
       Water:FineAggregate        SP:CoarseAggregate          SP:FineAggregate CoarseAggregate:FineAggregate
                       TRUE                        TRUE                        TRUE                        TRUE
```

Interpretation:
The vif() function provides a check for multicollinearity condition using variance inflation factor. If SQRT(vif) >2 it indicates a multicollinearity problem. Based on the result , in all 3 models have multicollinearity problem. It doesn't make any sense to drop them all,so we would keep the model as what they were and not drop any variables.

Question 1.4:

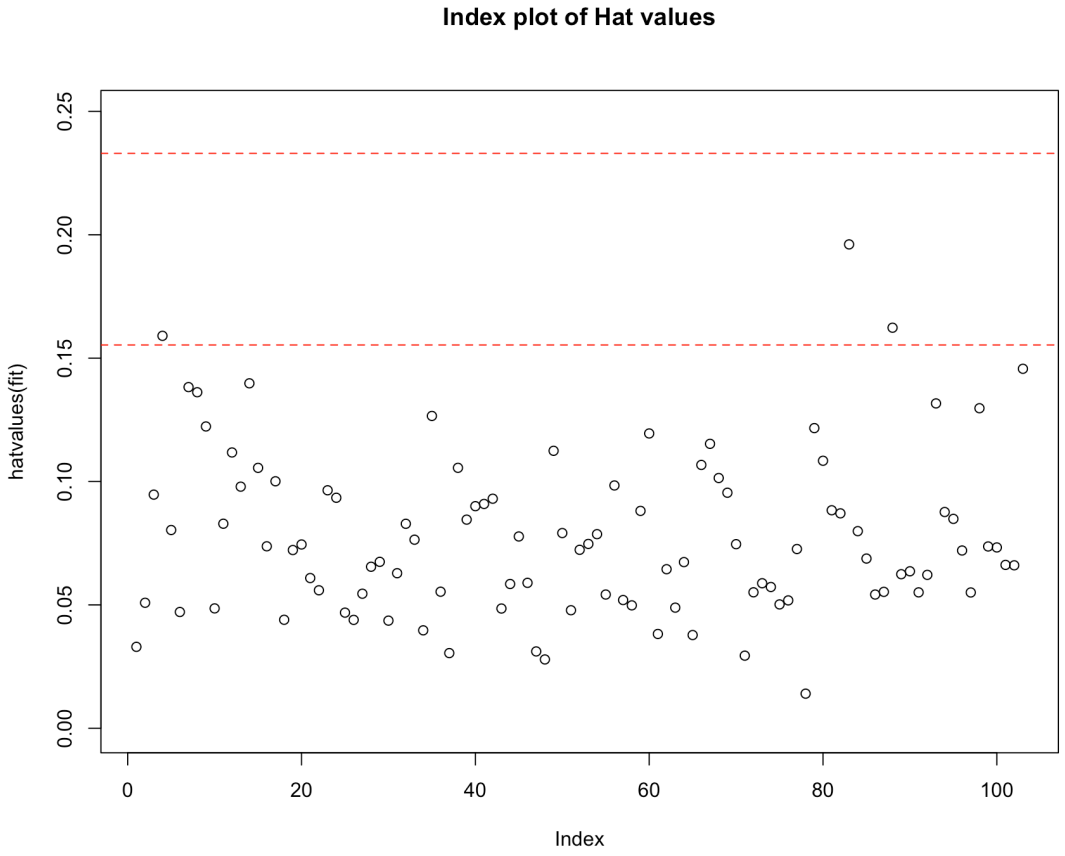
Unusual observations:
Outliers:
outlierTest(fit_MLR)
outlierTest(fit_LRI)
outlierTest(fit_SLR)

```
> outlierTest(fit_MLR)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferonni p
69 -2.603738           0.010717           NA
> outlierTest(fit_LRI)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferonni p
69 -2.476504           0.015095           NA
> outlierTest(fit_SLR)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferonni p
22 -2.978955          0.0039249      0.40426
```

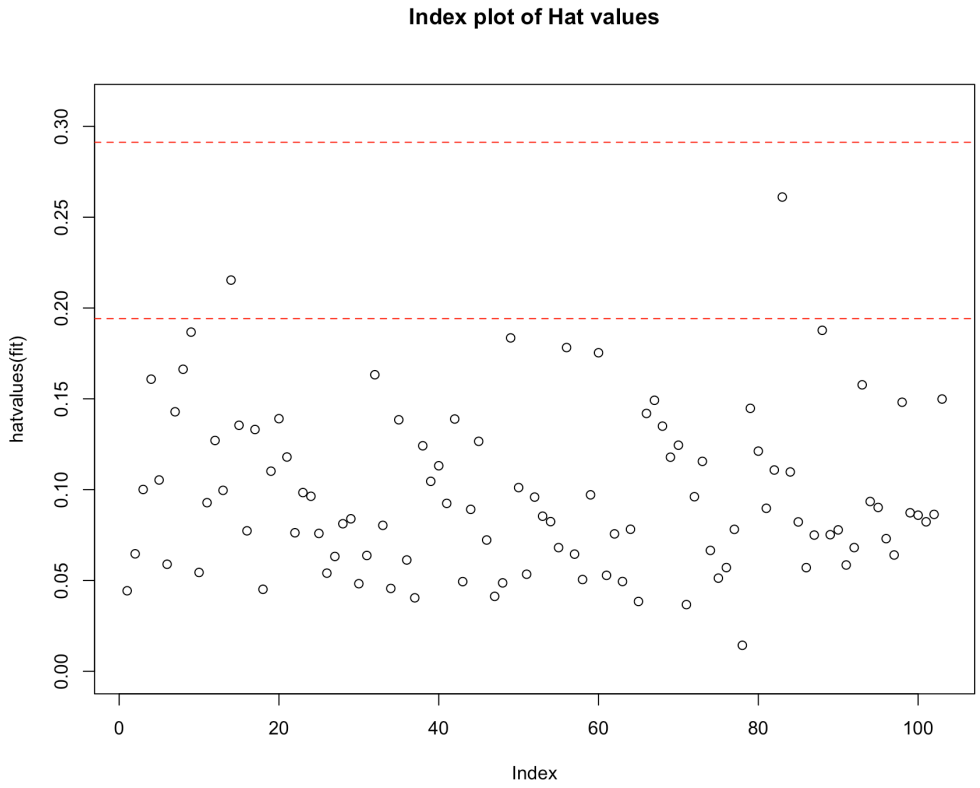
Interpretation:
The results of three models show no outliers. However, the points 69 and 22 have the largest studentized residual and should be considered to delete in the following corrective steps.

High Leverage points:

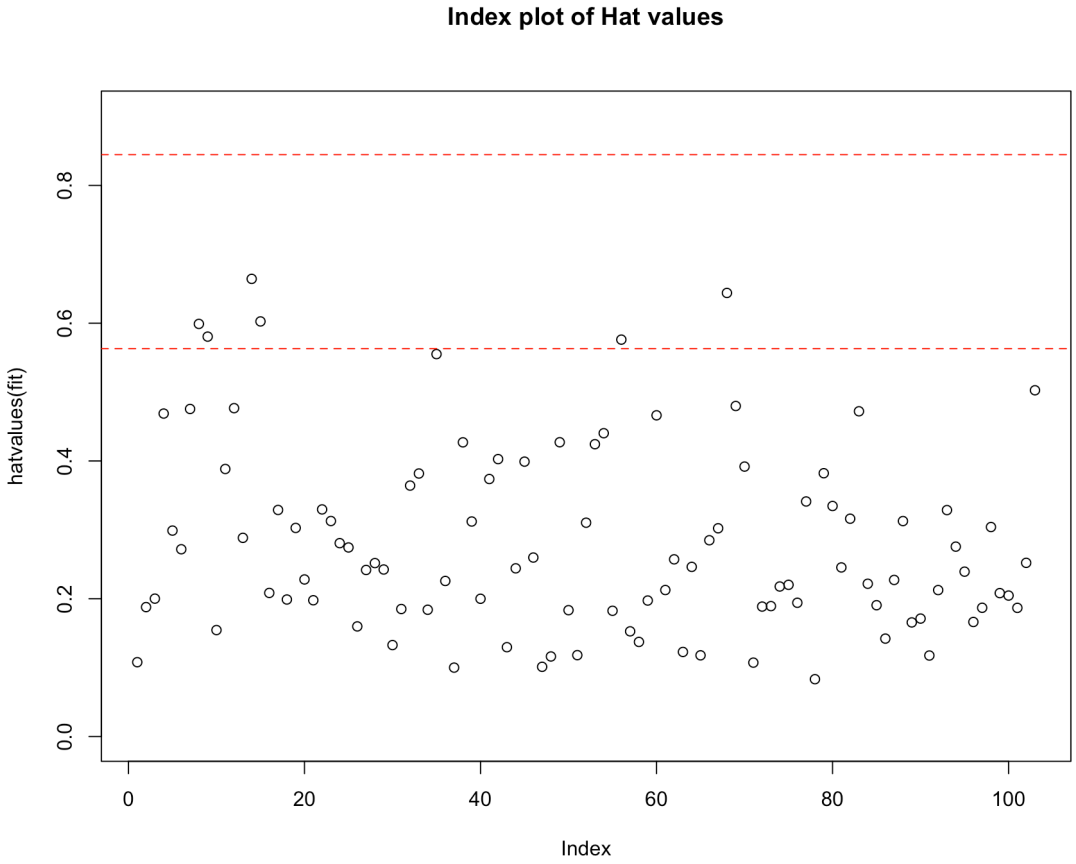
```
hat.plot<-function(fit){
  p<-length(coefficients(fit))
  n<-length(fitted(fit))
  plot(hatvalues(fit),ylim=c(0,3.2)*p/n,main="Index plot of Hat values")
  abline(h=c(2,3)*p/n,col="red",lty=2)
  identify(1:n,hatvalues(fit),names(hatvalues(fit)))
}

hat.plot(fit_MLR)
```



hat.plot(fit_LRI)



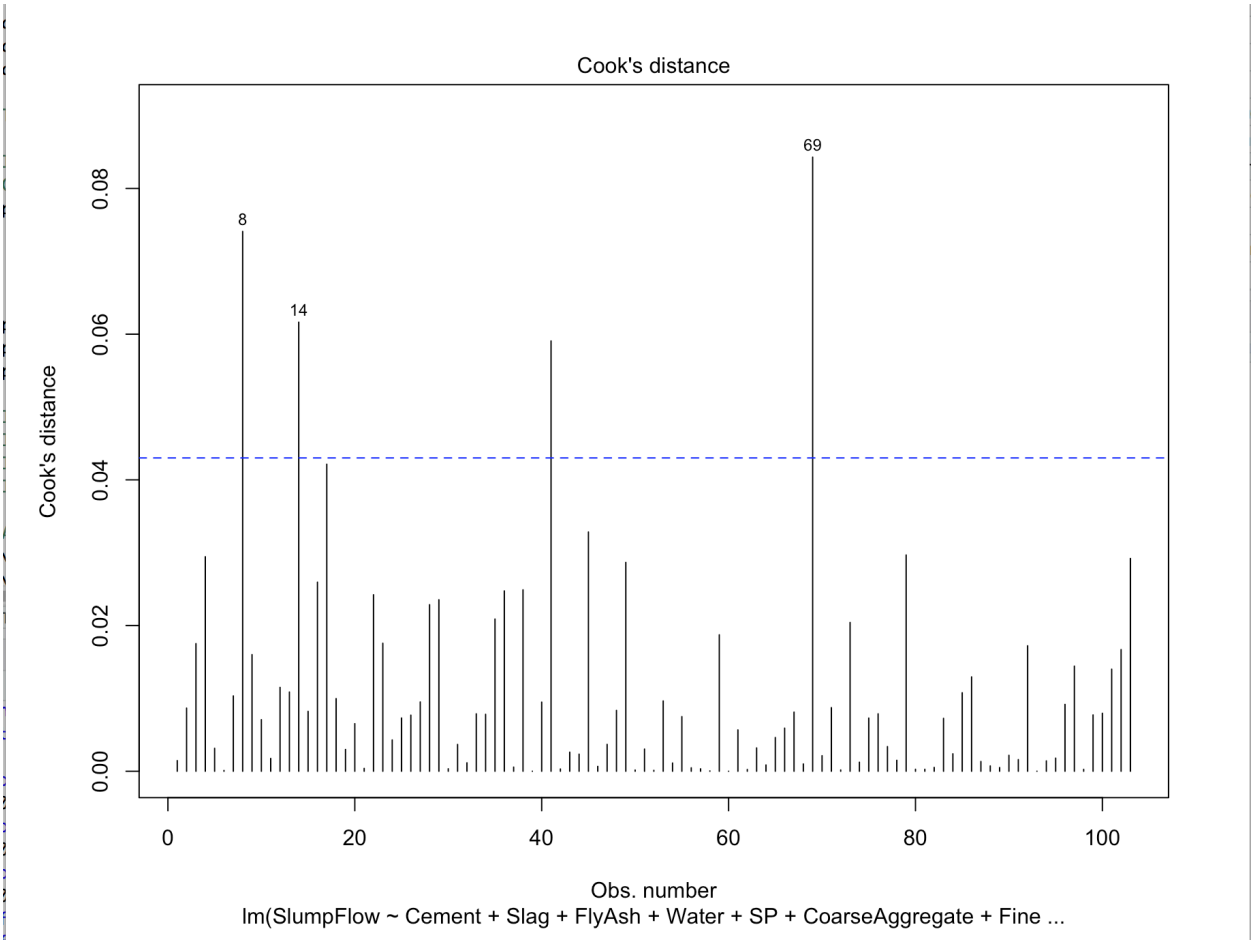
hat.plot(fit_SLR)



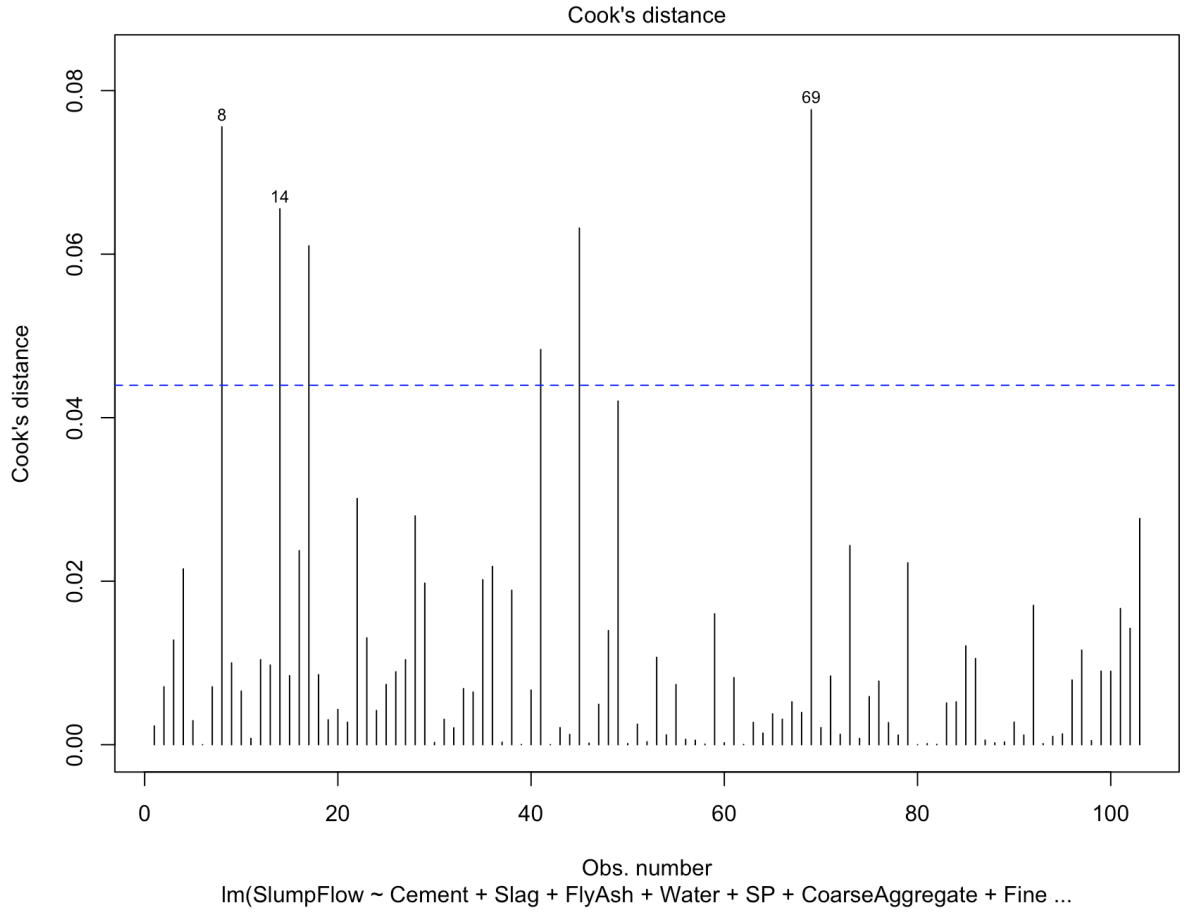
Interpretation:
There are no high leverage points greater than 3 times average hat value, which would facilitate the process of deleting outliers in the following steps.

Influential observations:
Cook's distance:

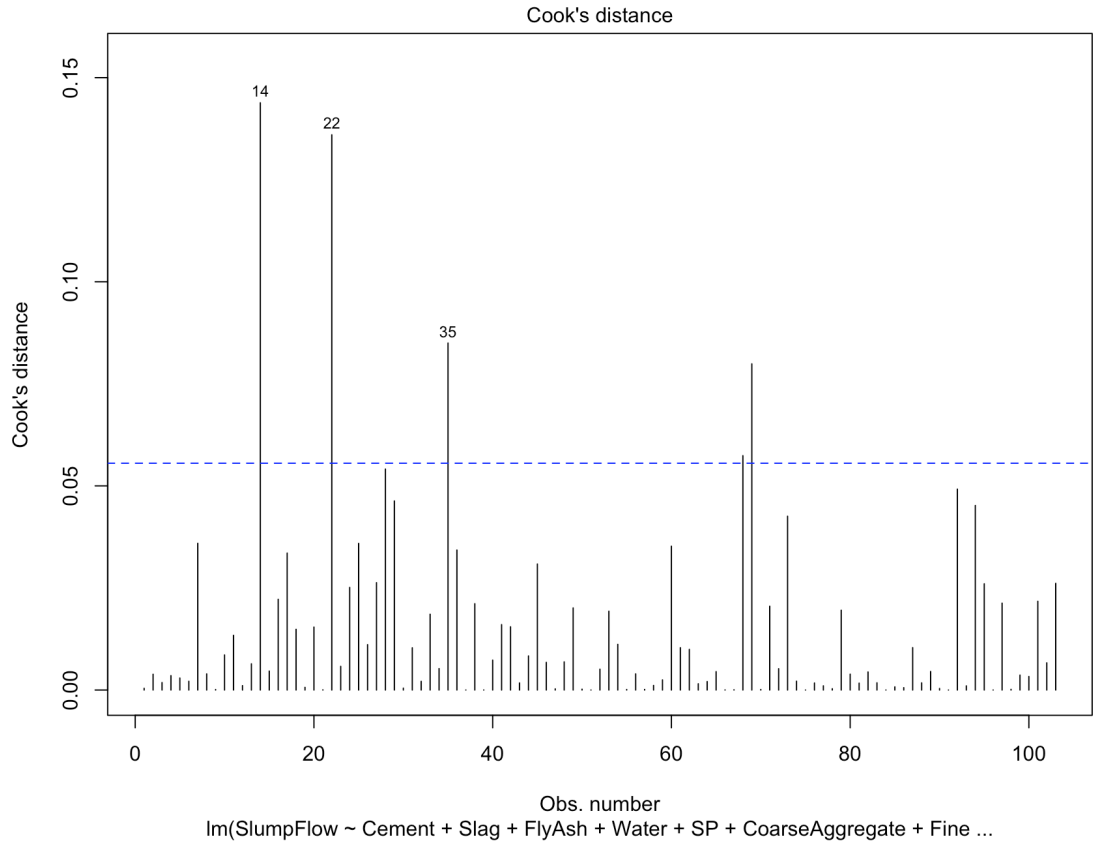
```
Dplot=function(fit,data){
  cutoff<-4/(nrow(data)-length(fit$coefficients)-2)
  plot(fit,which=4,cook.levels=cutoff)
  abline(h=cutoff,lty=2,col="blue")
}
Dplot(fit_MLR,conc)
```



Dplot(fit_LRI,conc)



Dplot(fit_SLR,conc)

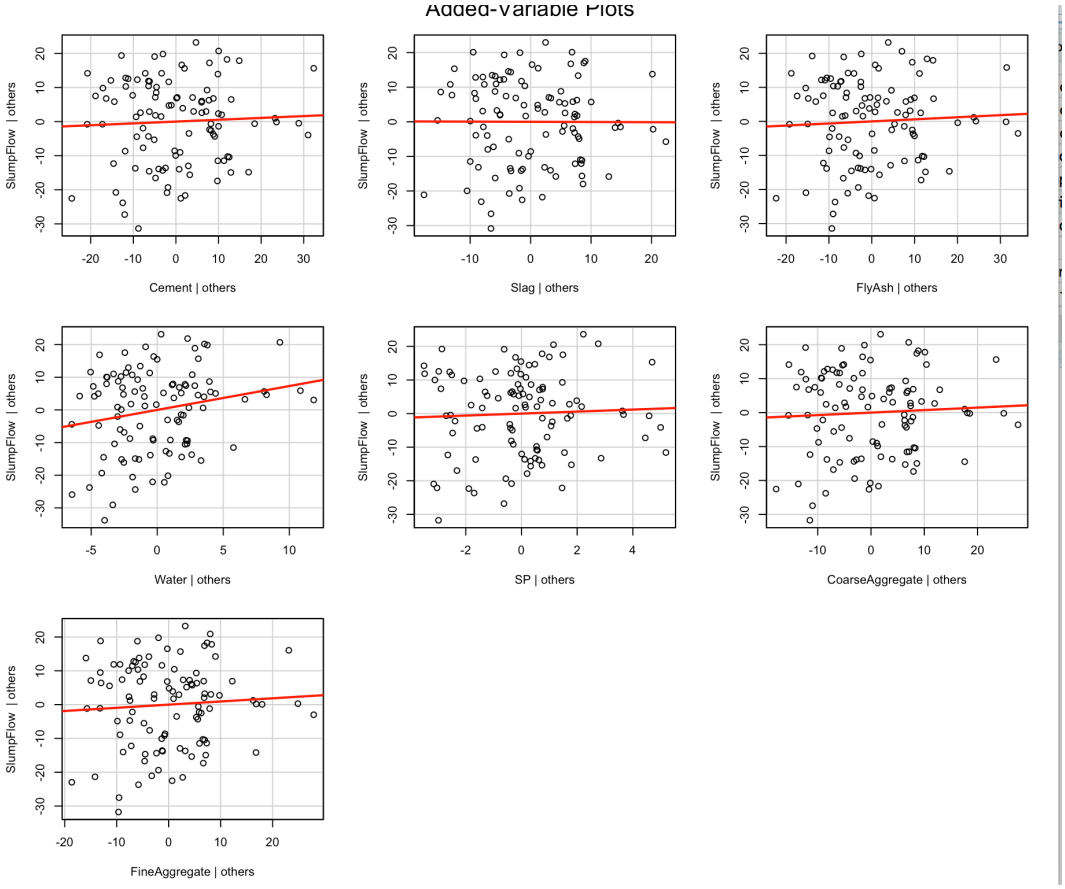


Interpretation:

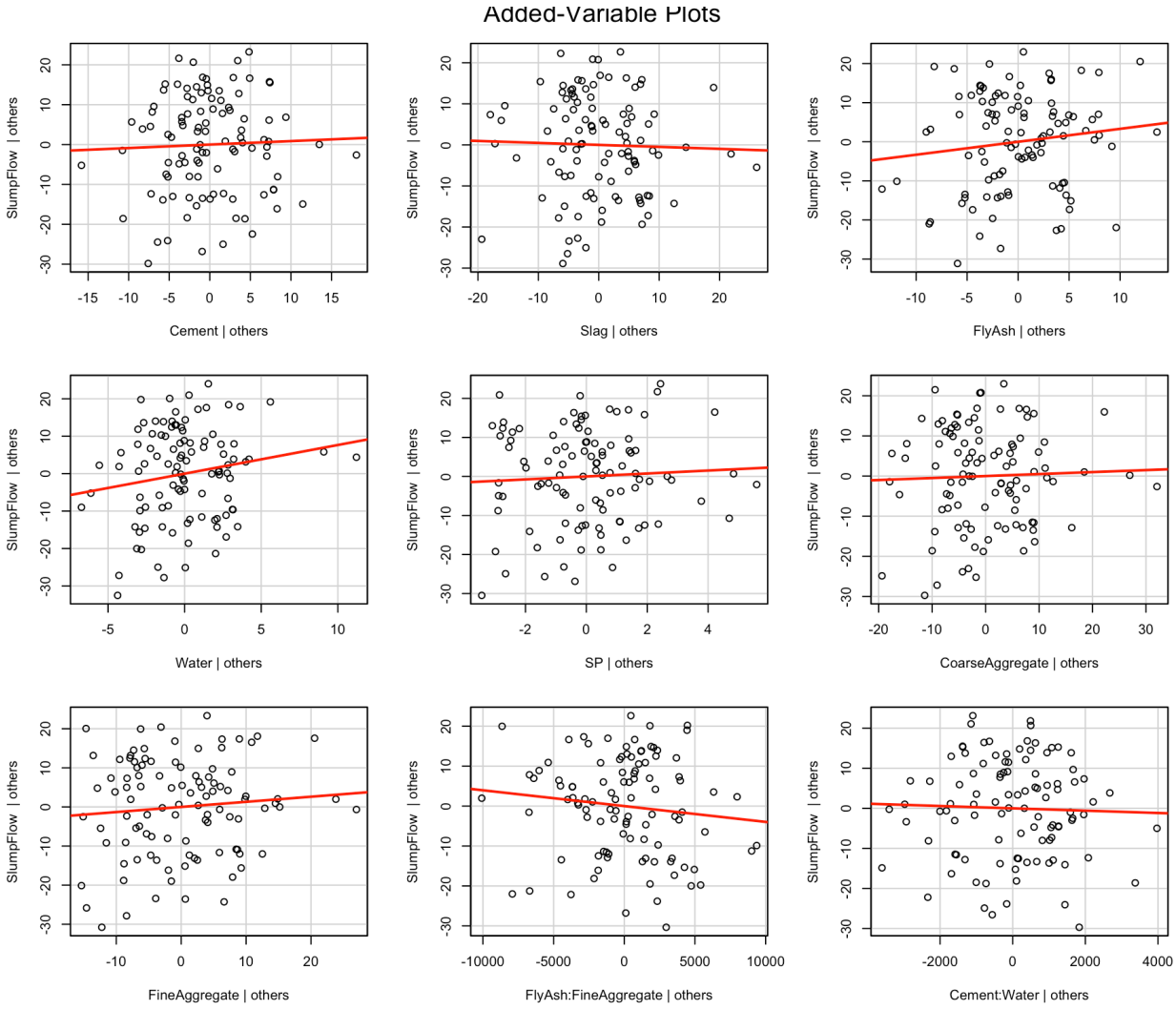
Influence observations are observations that have a disproportionate impact on the values of the model parameters. In LR, 69, 8, and 14, are those with most influence among the model parameters. In LR with Interactions, 69,49,41 are those with most influence. In Second Order Regression, 14, 22, 35 are those with most influence.

Added variable plots:

avPlots(fit_MLR, ask=FALSE, id.method ="identify",onepage=TRUE)



avPlots(fit_LRI, ask=FALSE, id.method ="identify",onepage=TRUE)



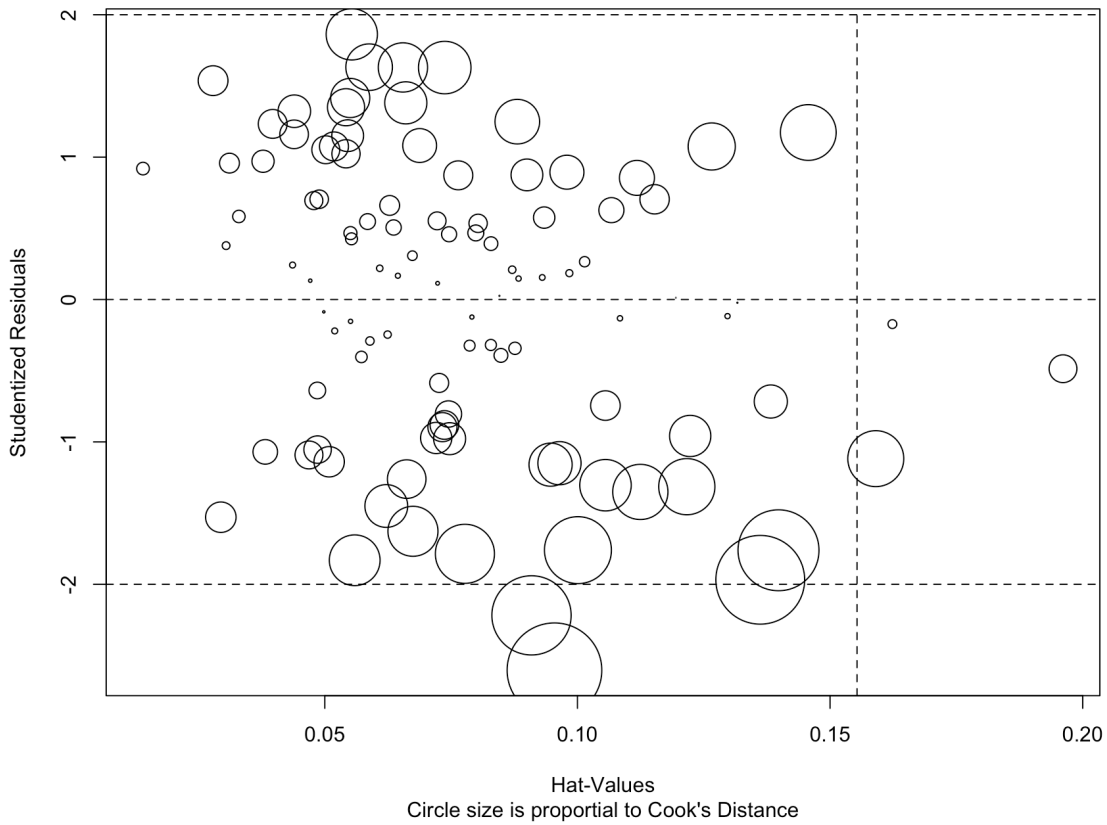
avPlots(fit_SLR, ask=FALSE,id.method ="identify", onepage=TRUE)

Interpretation:

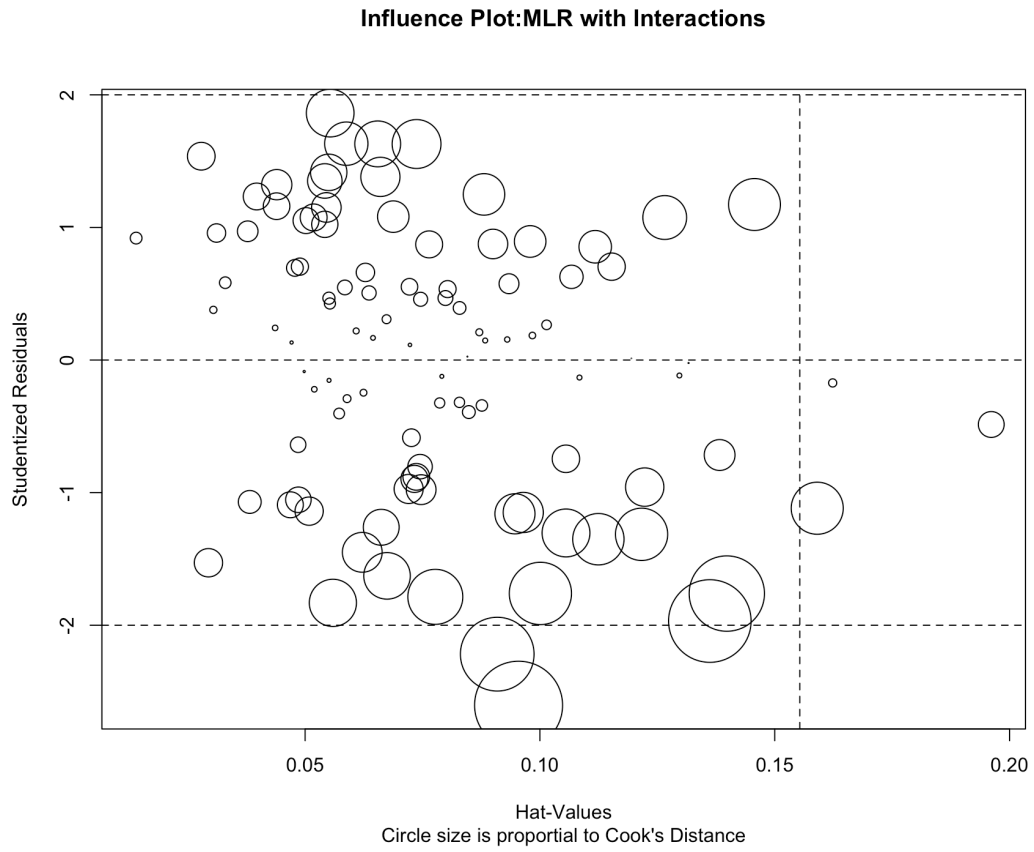
The Added-variable plots not only identify influential observations, they also explain the impact of various variables on response variable.

Combined information by influence plot:

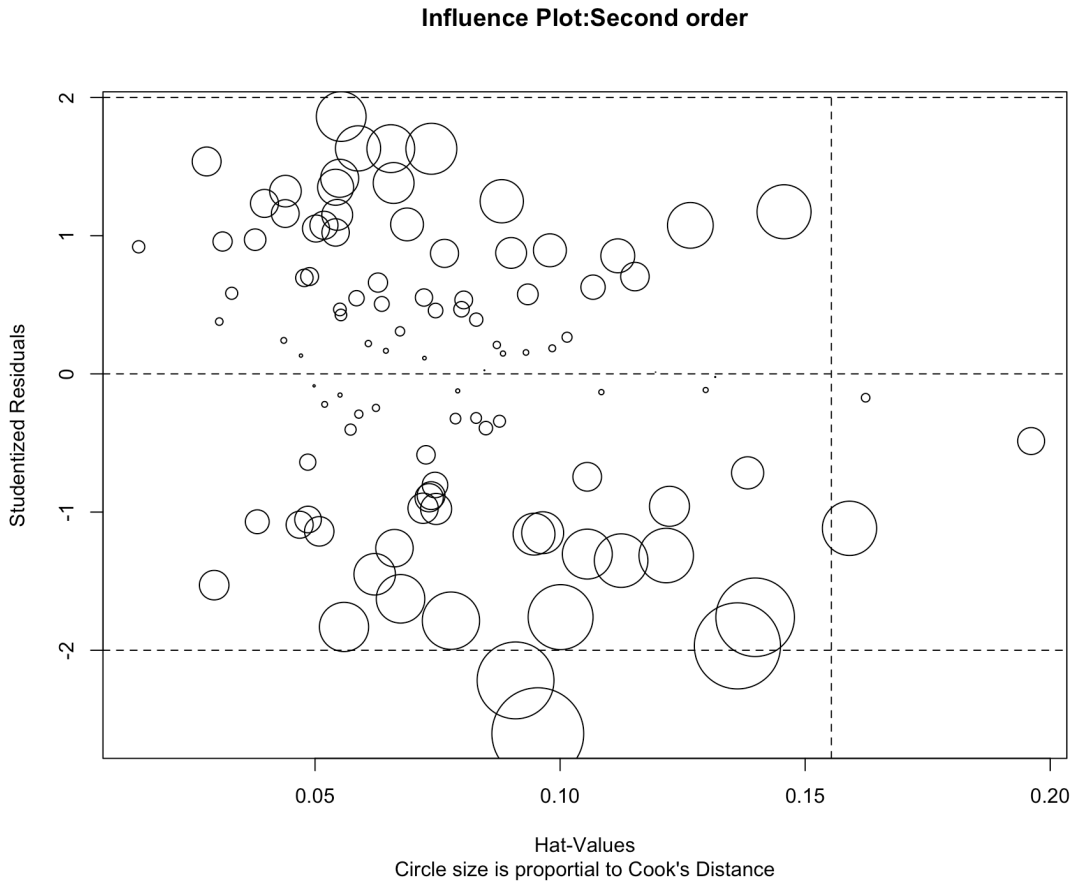
influencePlot(fit_MLR,id.method="identify", main="Influence Plot:MLR", sub="Circle size is proportial to Cook's Distance")



influencePlot(fit_MLR,id.method="identify", main="Influence Plot:MLR with Interactions", sub="Circle size is proportial to Cook's Distance")



influencePlot(fit_MLR,id.method="identify", main="Influence Plot:Second order", sub="Circle size is proportial to Cook's Distance")



Interpretation:
Items above +2 or below -2 on the vertical axis are considered outliers. Items above 0.5 on the horizontal axis have high leverage. Circle size is proportional to influence.

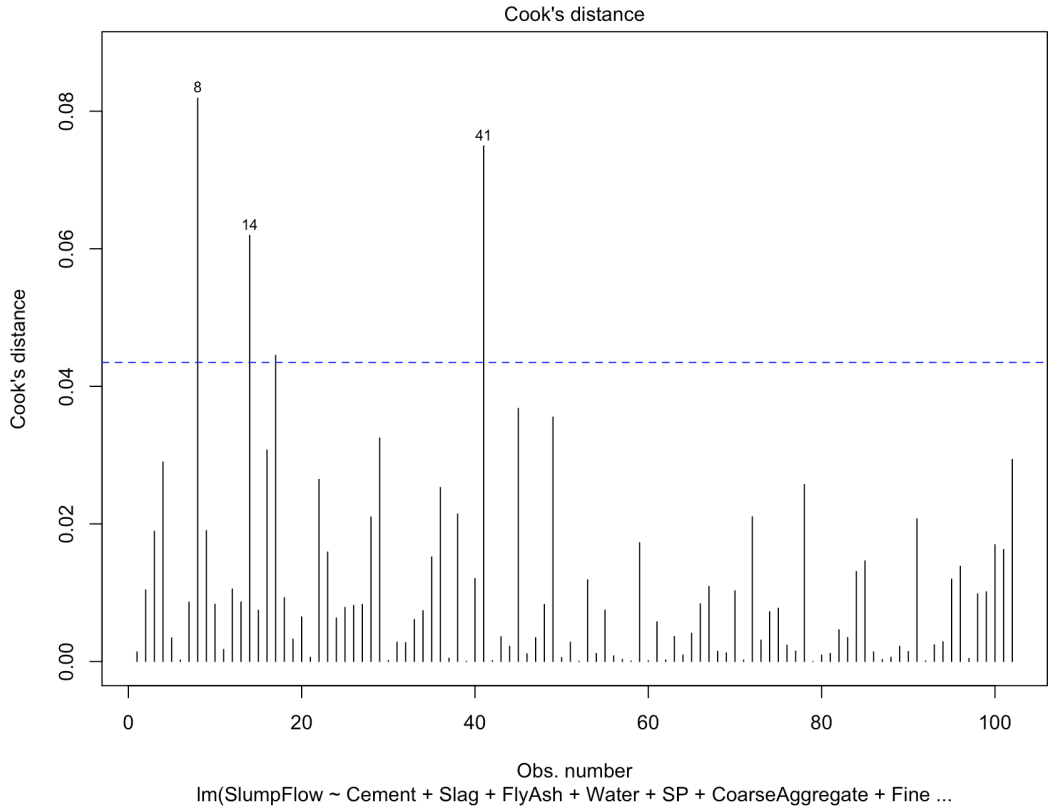
Corrective measures:
Deleting outliers:
Delete record 69 since it had the largest studentized residual and Cook's distance.
ConcreteData_deleted = conc[c(-69),]
ConcreteData_deleted

```
# A tibble: 102 x 10
   Cement  Slag FlyAsh Water    SP CoarseAggregate FineAggregate Slump SlumpFlow `28-dayCompressiveStrength`
    <dbl> <dbl>  <dbl> <dbl> <dbl>           <dbl>          <dbl> <dbl>      <dbl>                       <dbl>
 1    273  82.0    105   210  9.00             904            680 23.0       62.0                        35.0
 2    163 149      191   180 12.0              843            746  0         20.0                        41.1
 3    162 148      191   179 16.0              840            743  1.00      20.0                        41.8
 4    162 148      190   179 19.0              838            741  3.00      21.5                        42.1
 5    154 112      144   220 10.0              923            658 20.0       64.0                        26.8
 6    147  89.0    115   202  9.00             860            829 23.0       55.0                        25.2
 7    152 139      178   168 18.0              944            695  0         20.0                        38.9
 8    145   0      227   240  6.00             750            853 14.5       58.5                        36.6
 9    152   0      237   204  6.00             785            892 15.5       51.0                        32.7
10    304   0      140   214  6.00             895            722 19.0       51.0                        38.5
# ... with 92 more rows
```

```
fit_MLR_del<-lm(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate +
FineAggregate,data =ConcreteData_deleted)
summary(fit_MLR_del)
```

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate, data = ConcreteData_deleted)

Residuals:
    Min      1Q  Median      3Q     Max
-28.490  -9.472   1.955   9.265  22.481

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -150.28547  342.15673  -0.439   0.6615
Cement             0.03066    0.10945   0.280   0.7800
Slag              -0.03892    0.15237  -0.255   0.7990
FlyAsh             0.03609    0.11112   0.325   0.7461
Water              0.62914    0.34481   1.825   0.0712 .
SP                 0.02753    0.65170   0.042   0.9664
CoarseAggregate    0.03030    0.13222   0.229   0.8192
FineAggregate      0.05377    0.13865   0.388   0.6990
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.47 on 94 degrees of freedom
Multiple R-squared:  0.5223,    Adjusted R-squared:  0.4867
F-statistic: 14.68 on 7 and 94 DF,  p-value: 8.609e-13
```

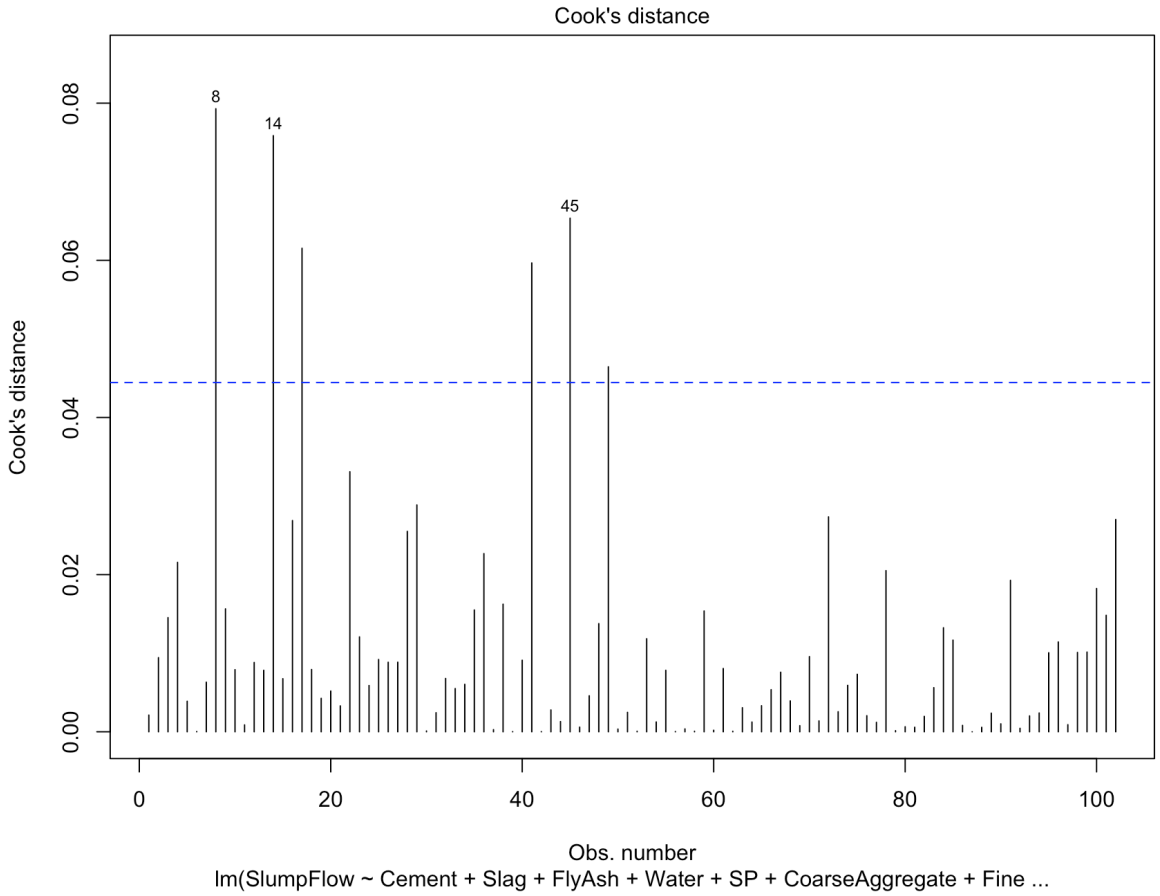
```
Dplot(fit_MLR_del, ConcreteData_deleted)
```




```
fit_LRI_del<-
lm(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate+FineAggregat
e:FlyAsh+Cement:Water,data =ConcreteData_deleted)
summary(fit_LRI_del)
```

```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + FineAggregate:FlyAsh +
    Cement:Water, data = ConcreteData_deleted)

Residuals:
     Min       1Q   Median       3Q      Max
-28.3977  -8.5529   0.9811   9.2948  22.6459

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)          -1.242e+02  3.652e+02  -0.340    0.734
Cement                5.429e-03  2.287e-01   0.024    0.981
Slag                 -8.116e-02  1.629e-01  -0.498    0.619
FlyAsh                2.521e-01  2.543e-01   0.991    0.324
Water                 5.855e-01  4.487e-01   1.305    0.195
SP                    5.060e-02  6.878e-01   0.074    0.942
CoarseAggregate       3.401e-03  1.400e-01   0.024    0.981
FineAggregate         7.868e-02  1.446e-01   0.544    0.588
FlyAsh:FineAggregate -3.245e-04  3.410e-04  -0.952    0.344
Cement:Water         -2.313e-06  8.741e-04  -0.003    0.998

Residual standard error: 12.54 on 92 degrees of freedom
Multiple R-squared:  0.5269,    Adjusted R-squared:  0.4807
F-statistic: 11.39 on 9 and 92 DF,  p-value: 8.475e-12
```

Dplot(fit_LRI_del, ConcreteData_deleted)



Interpretation:
The dataset without record 69 provides better results for both models.

Transforming variables:
Box-cox transformation to Normality:

summary(powerTransform(fit_MLR_del))

```
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
Y1    1.7071           2       1.1535       2.2607

Likelihood ratio tests about transformation parameters
                             LRT df         pval
LR test, lambda = (0) 40.164620  1 2.334379e-10
LR test, lambda = (1)  6.578362  1 1.032258e-02
```

We see that Y1 needs transformation.

```
Concrete_MLR_del_Trans = ConcreteData_deleted
Concrete_MLR_del_Trans[,1]=Concrete_MLR_del_Trans[,1]^1.707
fit_MLR_del_trans<-
lm(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate,data
=Concrete_MLR_del_Trans)
summary(powerTransform(fit_MLR_del_trans))
```

```
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
Y1    1.6851           2       1.1374       2.2329

Likelihood ratio tests about transformation parameters
                             LRT df         pval
LR test, lambda = (0) 39.822492  1 2.781237e-10
LR test, lambda = (1)  6.294054  1 1.211437e-02
```

```
summary(powerTransform(fit_SLR))
```

```
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
Y1    1.4235           1       0.8761       1.9709

Likelihood ratio tests about transformation parameters
                            LRT df         pval
LR test, lambda = (0) 29.032520  1 7.117341e-08
LR test, lambda = (1)  2.392242  1 1.219387e-01
```

Interpretation:
For lambda = 1, pval=0.12. Y1 conforms to Normality and doesn't need transformation

Adding or deleting variables
Based on the result of SQRT(vif) > 2 for most of the variables multicollinearity condition is not satisfied. However, we cannot deleting all variables or cannot add anything else.

Question 1.5:
Selecting the best regression model:
Comparing Mutiple linear regression, linear regression with interaction and Second order linear regression:

```
anova(fit_MLR, fit_LRI, fit_SLR)
```

```
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     95 15671.3
2     93 15439.5  2     231.8 1.0980   0.3389
3     74  7810.5 19    7629.0 3.8042 1.77e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Model with least RSS value is the best- Second order linear regression is the best model.

AIC(fit_MLR, fit_LRI, fit_SLR)

```
          df      AIC
fit_MLR    9 827.8614
fit_LRI   11 830.3267
fit_SLR   30 798.1361
```

Models with smaller AIC values are preferred- Second order linear regression is the best model.

Variable selection:
Backwards stepwise selection

```
library(MASS)
stepAIC(fit_SLR, direction = "backward")
```

```
Console  Terminal
~/
- FlyAsh:CoarseAggregate  1     825.1  9269.1 498.75
- FlyAsh:Water            1     928.9  9306.9 499.89
- Slag:FlyAsh             1    1435.8  9813.8 505.35
- Slag:FineAggregate      1    2897.8 11275.8 519.66
- Slag:Water              1    3572.7 11950.7 525.64
- Slag:CoarseAggregate    1    3892.7 12270.7 528.36

Step:  AIC=490.72
SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate +
    FineAggregate + Cement:Water + Slag:FlyAsh + Slag:Water +
    Slag:CoarseAggregate + Slag:FineAggregate + FlyAsh:Water +
    FlyAsh:CoarseAggregate + FlyAsh:FineAggregate + Water:CoarseAggregate +
    Water:FineAggregate

                         Df Sum of Sq     RSS    AIC
<none>                                  8514.5 490.72
- Cement:Water            1     415.5  8930.1 493.63
- Water:CoarseAggregate   1     423.5  8938.1 493.72
- Water:FineAggregate     1     778.8  9293.4 497.74
- FlyAsh:FineAggregate    1     784.8  9299.3 497.81
- SP                      1     946.2  9460.8 499.58
- FlyAsh:Water            1    1184.6  9699.1 502.14
- FlyAsh:CoarseAggregate  1    1333.5  9848.0 503.71
- Slag:FlyAsh             1    1620.7 10135.2 506.67
- Slag:FineAggregate      1    2936.8 11451.3 519.25
- Slag:Water              1    3806.9 12321.5 526.79
- Slag:CoarseAggregate    1    4022.3 12536.9 528.58

Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + Cement:Water + Slag:FlyAsh +
    Slag:Water + Slag:CoarseAggregate + Slag:FineAggregate +
    FlyAsh:Water + FlyAsh:CoarseAggregate + FlyAsh:FineAggregate +
    Water:CoarseAggregate + Water:FineAggregate, data = conc)

Coefficients:
          (Intercept)                 Cement                   Slag                  FlyAsh                  Water
            872.162339              -0.188274              -6.278219              -2.399647              -5.574350
                   SP         CoarseAggregate          FineAggregate         Cement:Water           Slag:FlyAsh
              1.867179              -0.337620              -0.768420               0.002577               0.001353
           Slag:Water  Slag:CoarseAggregate    Slag:FineAggregate         FlyAsh:Water  FlyAsh:CoarseAggregate
              0.012597               0.002325               0.002588               0.004615               0.001030
 FlyAsh:FineAggregate  Water:CoarseAggregate  Water:FineAggregate
              0.001121               0.002183               0.004231
```

```
fit_SLR_Back<-lm(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate +
                 FineAggregate + Cement:Water + Slag:FlyAsh + Slag:Water +
                 Slag:CoarseAggregate + Slag:FineAggregate + FlyAsh:Water +
                 FlyAsh:CoarseAggregate + FlyAsh:FineAggregate + Water:CoarseAggregate +
                 Water:FineAggregate,data=conc)
summary(fit_SLR_Back)
```

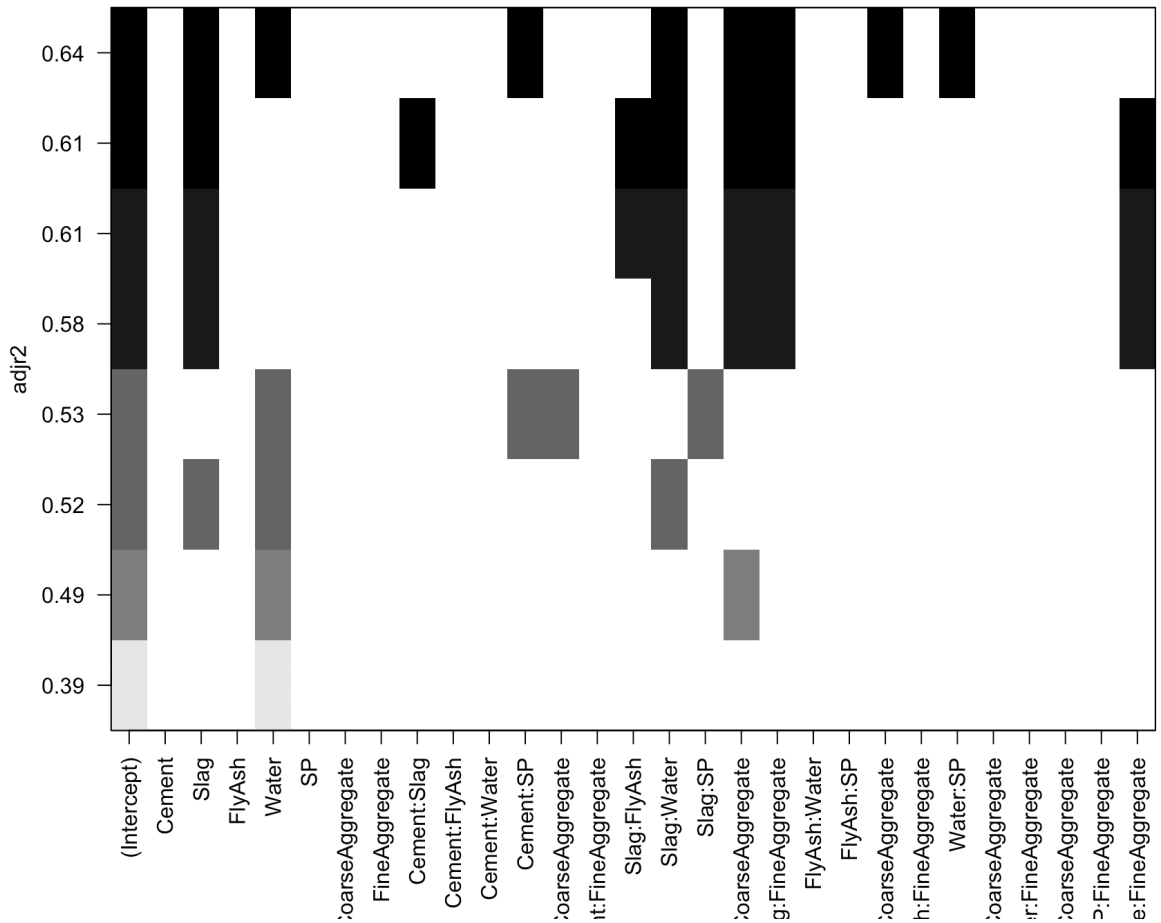
```
Call:
lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
    CoarseAggregate + FineAggregate + Cement:Water + Slag:FlyAsh +
    Slag:Water + Slag:CoarseAggregate + Slag:FineAggregate +
    FlyAsh:Water + FlyAsh:CoarseAggregate + FlyAsh:FineAggregate +
    Water:CoarseAggregate + Water:FineAggregate, data = conc)

Residuals:
     Min       1Q   Median       3Q      Max
-23.6002  -5.9627   0.6819   4.5737  20.6809

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             8.722e+02  5.373e+02   1.623 0.108242
Cement                 -1.883e-01  2.744e-01  -0.686 0.494429
Slag                   -6.278e+00  8.665e-01  -7.245 1.82e-10 ***
FlyAsh                 -2.400e+00  6.589e-01  -3.642 0.000465 ***
Water                  -5.574e+00  2.298e+00  -2.426 0.017384 *
SP                      1.867e+00  6.075e-01   3.073 0.002844 **
CoarseAggregate        -3.376e-01  2.321e-01  -1.455 0.149449
FineAggregate          -7.684e-01  3.317e-01  -2.317 0.022913 *
Cement:Water            2.577e-03  1.265e-03   2.037 0.044792 *
Slag:FlyAsh             1.353e-03  3.363e-04   4.022 0.000124 ***
Slag:Water              1.260e-02  2.044e-03   6.165 2.29e-08 ***
Slag:CoarseAggregate    2.325e-03  3.669e-04   6.337 1.08e-08 ***
Slag:FineAggregate      2.588e-03  4.779e-04   5.415 5.60e-07 ***
FlyAsh:Water            4.615e-03  1.342e-03   3.439 0.000907 ***
FlyAsh:CoarseAggregate  1.030e-03  2.824e-04   3.649 0.000454 ***
FlyAsh:FineAggregate    1.121e-03  4.006e-04   2.799 0.006342 **
Water:CoarseAggregate   2.183e-03  1.062e-03   2.056 0.042825 *
Water:FineAggregate     4.231e-03  1.517e-03   2.788 0.006535 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.01 on 85 degrees of freedom
Multiple R-squared:  0.7296,    Adjusted R-squared:  0.6755
F-statistic: 13.49 on 17 and 85 DF,  p-value: < 2.2e-16
```

summary(fit_SLR)

```
    FlyAsh:FineAggregate + Water:SP + Water:CoarseAggregate +
    Water:FineAggregate + SP:CoarseAggregate + SP:FineAggregate +
    CoarseAggregate:FineAggregate, data = conc)

Residuals:
     Min       1Q   Median       3Q      Max
-23.8222  -6.0751   0.2499   4.7302  21.2758

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    1.567e+03  1.277e+03   1.227 0.223715
Cement                        -1.638e+00  1.387e+00  -1.181 0.241227
Slag                          -5.560e+00  1.495e+00  -3.719 0.000386 ***
FlyAsh                        -3.498e+00  1.162e+00  -3.010 0.003568 **
Water                         -6.165e+00  2.778e+00  -2.219 0.029543 *
SP                            -9.203e+01  1.474e+02  -0.624 0.534359
CoarseAggregate               -5.943e-01  5.978e-01  -0.994 0.323325
FineAggregate                 -9.309e-01  7.902e-01  -1.178 0.242545
Cement:Slag                   -2.639e-04  5.594e-04  -0.472 0.638511
Cement:FlyAsh                  3.774e-04  4.528e-04   0.834 0.407198
Cement:Water                   4.472e-03  2.183e-03   2.049 0.044004 *
Cement:SP                      4.826e-02  5.069e-02   0.952 0.344250
Cement:CoarseAggregate         5.554e-04  5.822e-04   0.954 0.343217
Cement:FineAggregate           3.448e-04  6.659e-04   0.518 0.606098
Slag:FlyAsh                    9.259e-04  4.603e-04   2.011 0.047927 *
Slag:Water                     1.246e-02  2.541e-03   4.903 5.44e-06 ***
Slag:SP                        4.740e-02  7.788e-02   0.609 0.544640
Slag:CoarseAggregate           1.928e-03  5.389e-04   3.577 0.000618 ***
Slag:FineAggregate             1.972e-03  7.217e-04   2.732 0.007860 **
FlyAsh:Water                   5.582e-03  1.770e-03   3.153 0.002331 **
FlyAsh:SP                      4.320e-02  5.692e-02   0.759 0.450241
FlyAsh:CoarseAggregate         1.428e-03  4.753e-04   3.005 0.003624 **
FlyAsh:FineAggregate           1.433e-03  5.691e-04   2.519 0.013940 *
Water:SP                       5.024e-02  1.347e-01   0.373 0.710204
Water:CoarseAggregate          2.135e-03  1.191e-03   1.793 0.077110 .
Water:FineAggregate            4.104e-03  1.841e-03   2.229 0.028857 *
SP:CoarseAggregate             3.877e-02  5.893e-02   0.658 0.512625
SP:FineAggregate               3.905e-02  6.008e-02   0.650 0.517680
CoarseAggregate:FineAggregate -1.164e-04  4.208e-04  -0.276 0.782943
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.27 on 74 degrees of freedom
Multiple R-squared:  0.7519,    Adjusted R-squared:  0.658
F-statistic:  8.01 on 28 and 74 DF,  p-value: 3.907e-13
```

All subsets regression:

```
library(leaps)
leaps<-
regsubsets(SlumpFlow~Cement+Slag+FlyAsh+Water+SP+CoarseAggregate+FineAggregate+

Cement:Slag+Cement:FlyAsh+Cement:Water+Cement:SP+Cement:CoarseAggregate+Cement:Fi
neAggregate+ Slag:FlyAsh+Slag:Water+Slag:SP+Slag:CoarseAggregate+Slag:FineAggregate+
FlyAsh:Water+FlyAsh:SP+FlyAsh:CoarseAggregate+FlyAsh:FineAggregate+
Water:SP+Water:CoarseAggregate+Water:FineAggregate+
            SP:CoarseAggregate+SP:FineAggregate+
            CoarseAggregate:FineAggregate, data = conc, nbest=1)
plot(leaps, scale="adjr2")
```



Interpreting the prediction results:

The model needs to be refined by reducing redundant variables from a larger group. Based on the Backwards stepwise selection, the model has been refined into a simpler one and its AIC drops from 503.83 to 490.72. Another approach to reselect the variables is all subset regression, either based on adjusted R-squared or on Mallow's Cp value. Comparing adjusted R-Squared of the refined model and original one, the original model is around 0.66, and the refined model has 0.675. Also, the residual error is lesser for the original model. Therefore, it is the better model.

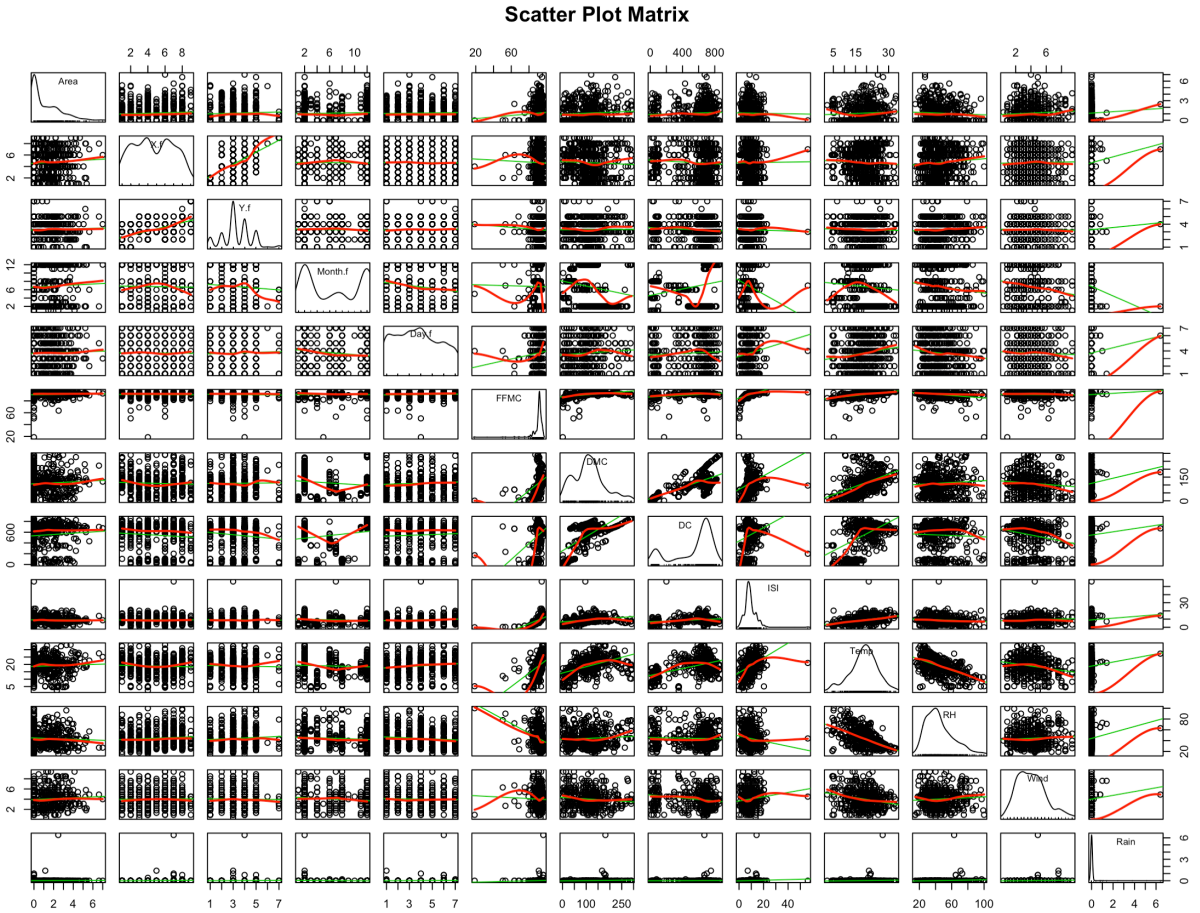
Task 4:

```
Forest_Fires_Data <- read_excel("Forest Fires Data.xlsx")
View(Forest_Fires_Data)
Forest_Fires_Data
```

```
# A tibble: 517 x 13
       X     Y Month Day    FFMC   DMC    DC   ISI  Temp    RH  Wind  Rain  Area
   <dbl> <dbl> <chr> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1  7.00  5.00 mar   fri     86.2  26.2  94.3  5.10  8.20  51.0 6.70  0         0
 2  7.00  4.00 oct   tue     90.6  35.4 669    6.70 18.0   33.0 0.900 0         0
 3  7.00  4.00 oct   sat     90.6  43.7 687    6.70 14.6   33.0 1.30  0         0
 4  8.00  6.00 mar   fri     91.7  33.3  77.5  9.00  8.30  97.0 4.00  0.200     0
 5  8.00  6.00 mar   sun     89.3  51.3 102    9.60 11.4   99.0 1.80  0         0
 6  8.00  6.00 aug   sun     92.3  85.3 488   14.7  22.2   29.0 5.40  0         0
 7  8.00  6.00 aug   mon     92.3  88.9 496    8.50 24.1   27.0 3.10  0         0
 8  8.00  6.00 aug   mon     91.5 145   608   10.7   8.00  86.0 2.20  0         0
 9  8.00  6.00 sep   tue     91.0 130   693    7.00 13.1   63.0 5.40  0         0
10  7.00  5.00 sep   sat     92.5  88.0 699    7.10 22.8   40.0 4.00  0         0
# ... with 507 more rows
>
```

```
#Question 2.1
Forest_Fires_Data$X.f<-factor(Forest_Fires_Data$X)
Forest_Fires_Data$Y.f<-factor(Forest_Fires_Data$Y)
Forest_Fires_Data$Month.f<-factor(Forest_Fires_Data$Month)
Forest_Fires_Data$Day.f<-factor(Forest_Fires_Data$Day)
Forest_Fires_Data$Area<-log1p(Forest_Fires_Data$Area)
ForestfiresData<-Forest_Fires_Data[,c(13,14:17,5:12)]
ForestfiresData
scatterplotMatrix(ForestfiresData,spread = FALSE,lty.smooth=2,main="Scatter Plot Matrix")
# X-Y coordinates and Month and day information have been transformed into factors
```

Area has been transformed into LN(Area+1)

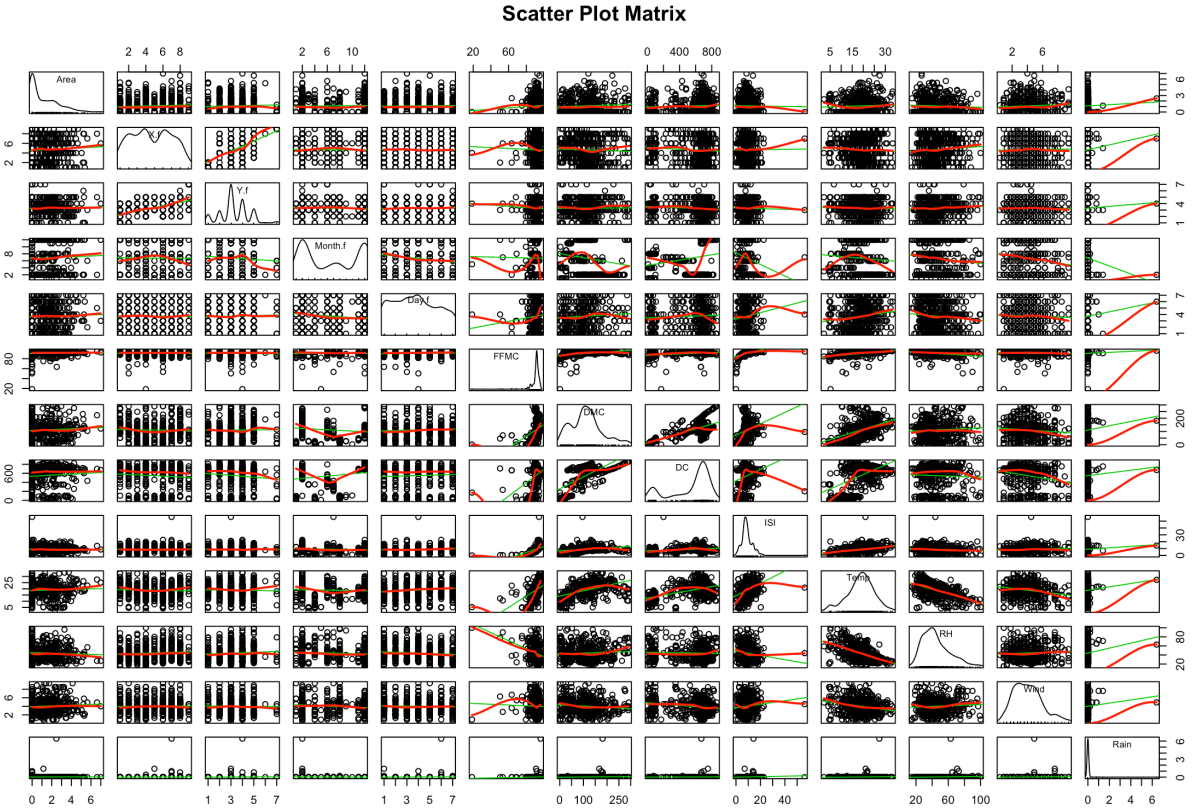


#The 4 categorical variables have been transformed to dummy variables by factor() method.
Area has been transformed into Ln(Area+1).
#The output data frame includes one response and twelve independent variables.
#In the scatter plot matrix, the points in four categorical variables are vertically arranged and don't show much correlation with the outcome variable.

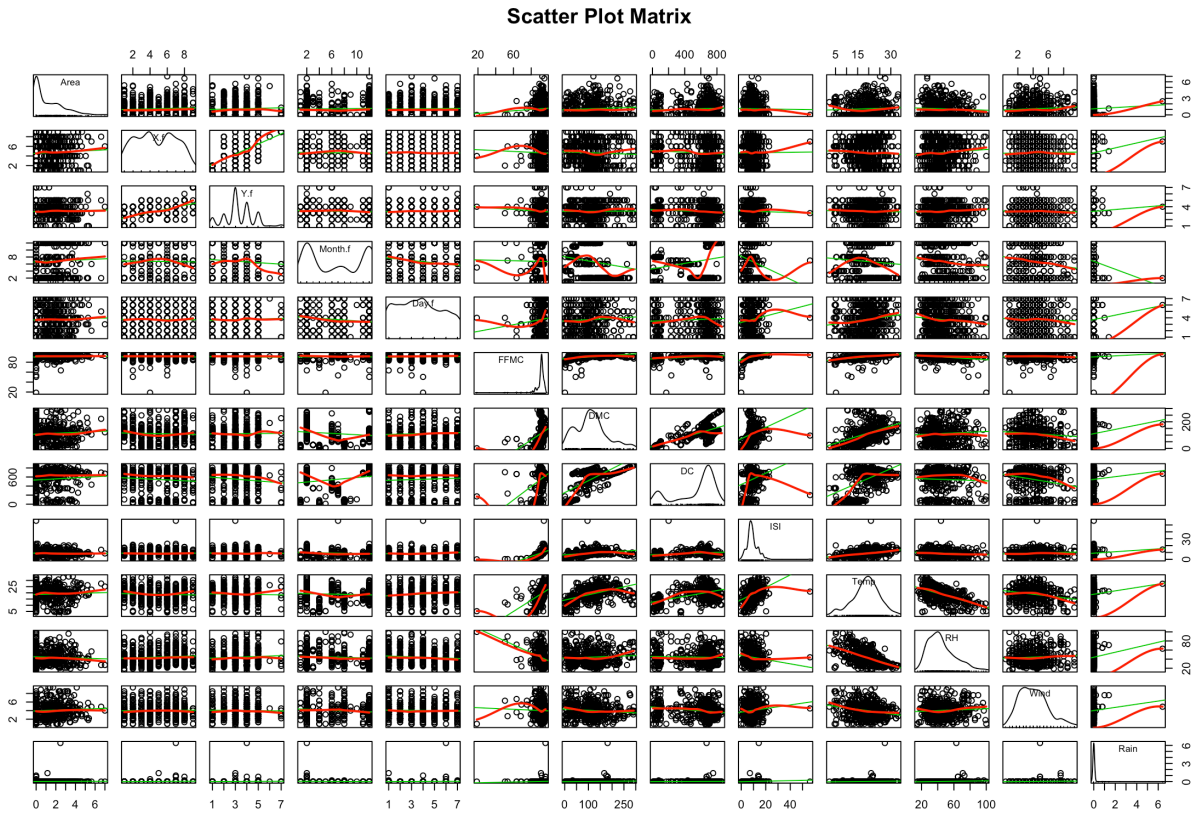
#Question 2.2:
#Use regression models:

#STFWI - Spacial, temporal and WI attributes
FF_fit_STFWI = lm(Area ~ X.f + Y.f + Month.f + Day.f + FFMC + DMC + DC + ISI,
data=ForestfiresData)

summary(FF_fit_STFWI)



#STM- Spacial, temporal and M(weather conditions)
FF_fit_STM = lm(Area ~ X.f + Y.f + Month.f + Day.f + Temp + RH + Wind + Rain,
data=ForestfiresData)
summary(FF_fit_STM)



#FWI- Fire Weather Index

```
FF_fit_FWI = lm(Area ~ FFMC + DMC + DC + ISI, data=ForestfiresData)
summary(FF_fit_FWI)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.3703 -1.1242 -0.6145  0.8882  5.8198

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0851719  1.1513950  -0.074    0.941
FFMC         0.0126619  0.0137577   0.920    0.358
DMC          0.0009234  0.0013587   0.680    0.497
DC           0.0001928  0.0003412   0.565    0.572
ISI         -0.0176878  0.0160811  -1.100    0.272

Residual standard error: 1.398 on 512 degrees of freedom
Multiple R-squared:  0.008046,  Adjusted R-squared:  0.0002959
F-statistic: 1.038 on 4 and 512 DF,  p-value: 0.3869
```

#Model M- uses the four weather conditions

```
FF_fit_M = lm(Area ~ Temp + RH + Wind + Rain, data=ForestfiresData)
summary(FF_fit_M)
```

#Interpretation:

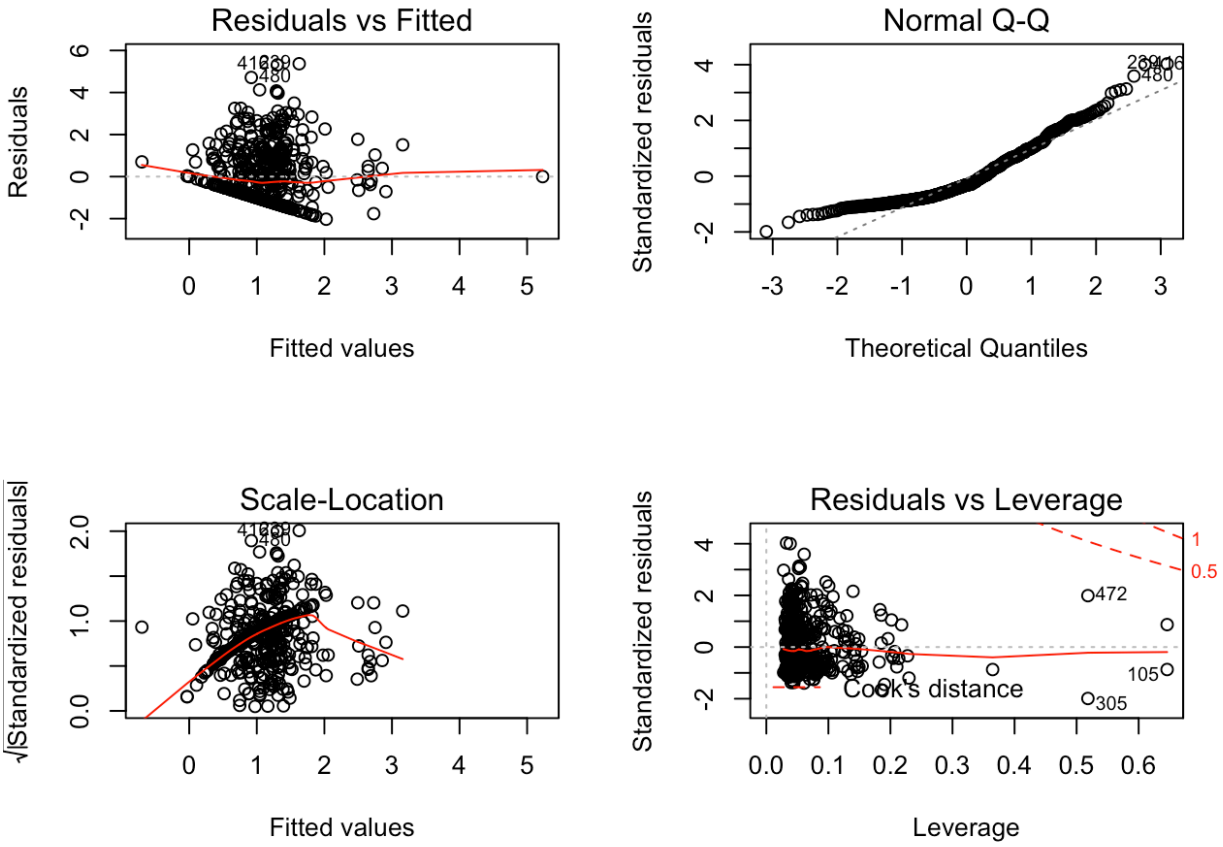
```
    Min      1Q  Median      3Q     Max
-1.3993 -1.0978 -0.7081  0.9121  5.7593

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.742041   0.443148   1.674   0.0946 .
Temp         0.012766   0.012958   0.985   0.3250
RH          -0.002834   0.004506  -0.629   0.5296
Wind         0.062603   0.035446   1.766   0.0780 .
Rain         0.085167   0.211852   0.402   0.6878
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.397 on 512 degrees of freedom
Multiple R-squared:  0.0104,    Adjusted R-squared:  0.002671
F-statistic: 1.345 on 4 and 512 DF,  p-value: 0.2519
```

#There are four algorithms mentioned in the paper- STFWI, STM, FWI, M that have been used.
#In terms of spatial and temporal variables lack of linearity as shown

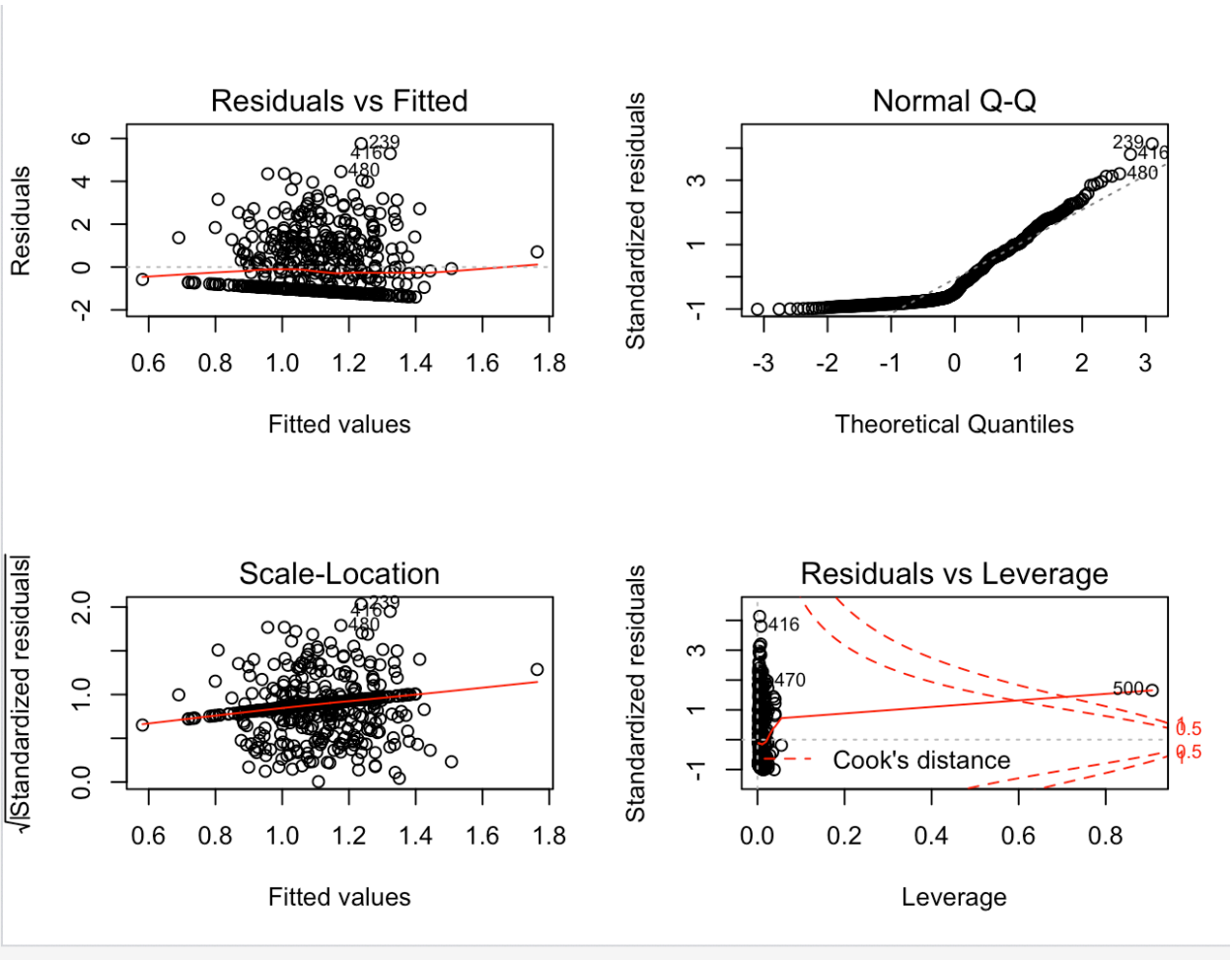
#Question 3.1:
#Regression diagnostics:
#Typical Approach

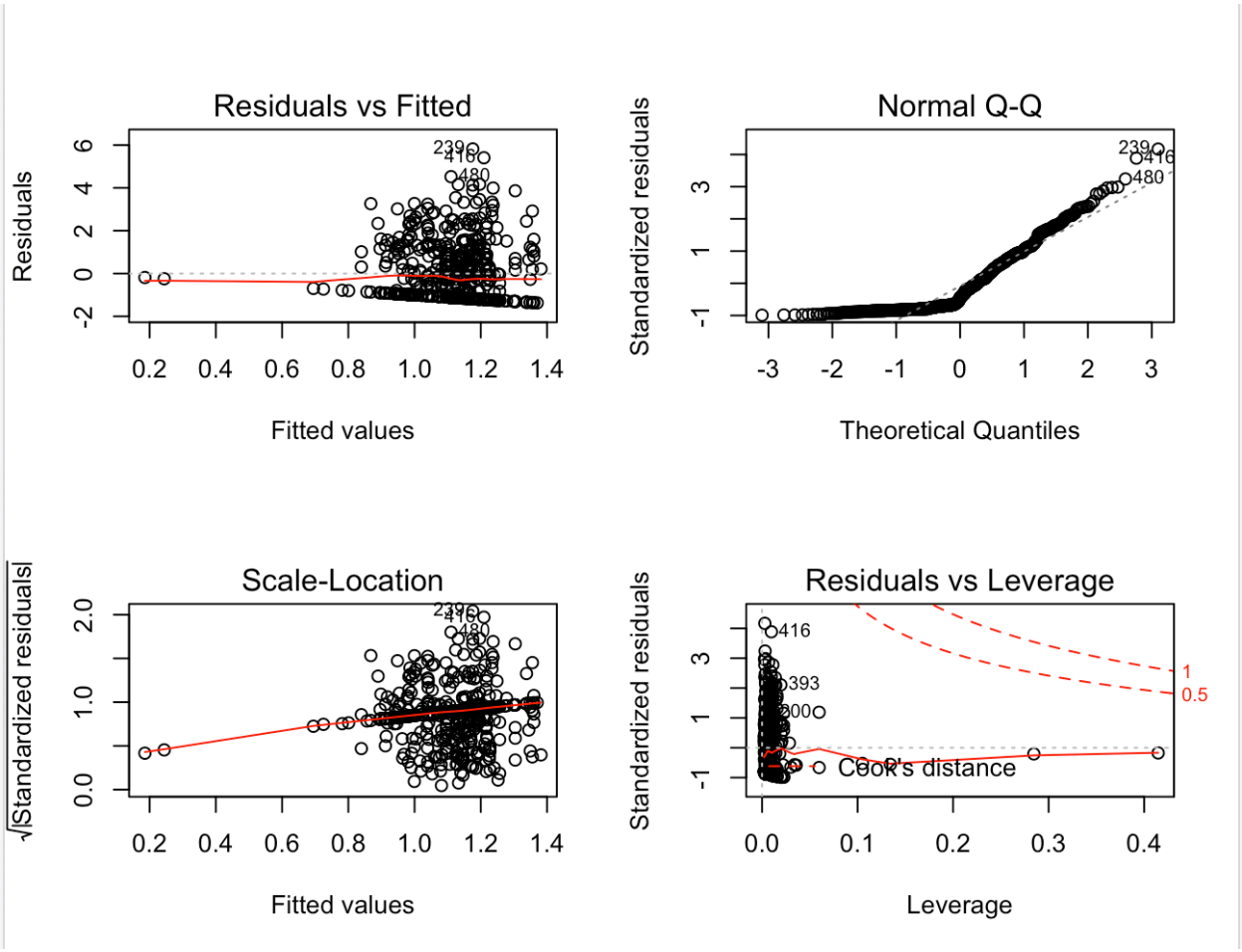
```
par(mfrow=c(2,2))
plot(FF_fit_STFWI)
```

plot(FF_fit_STM)



# Point 421 and 517 are highly leveraged that should be considered for deletion in the corrective step #
plot(FF_fit_FWI)
plot(FF_fit_M)

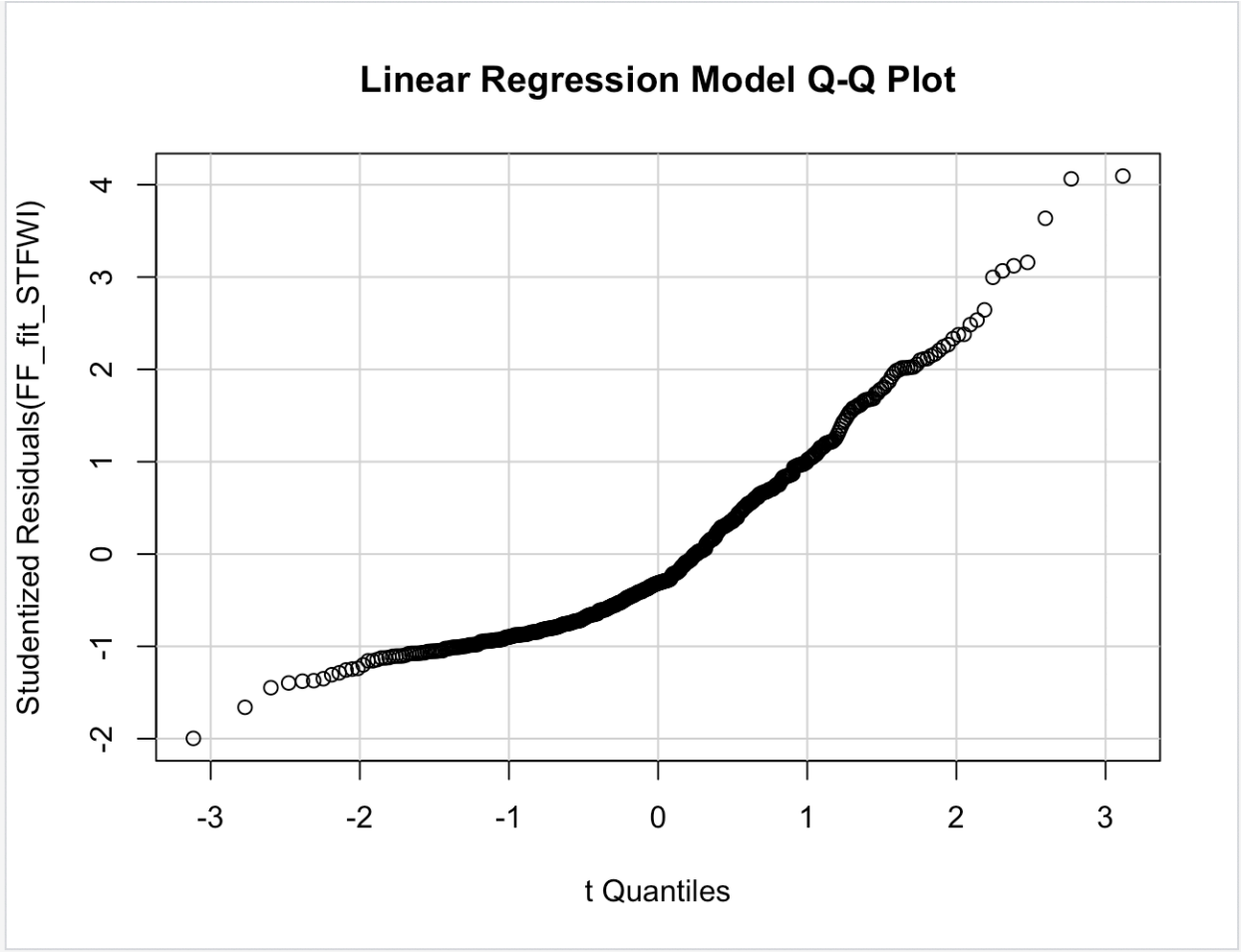

Residuals vs Fitted
Residuals
Fitted values
239
416
480
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
470
500
0.5



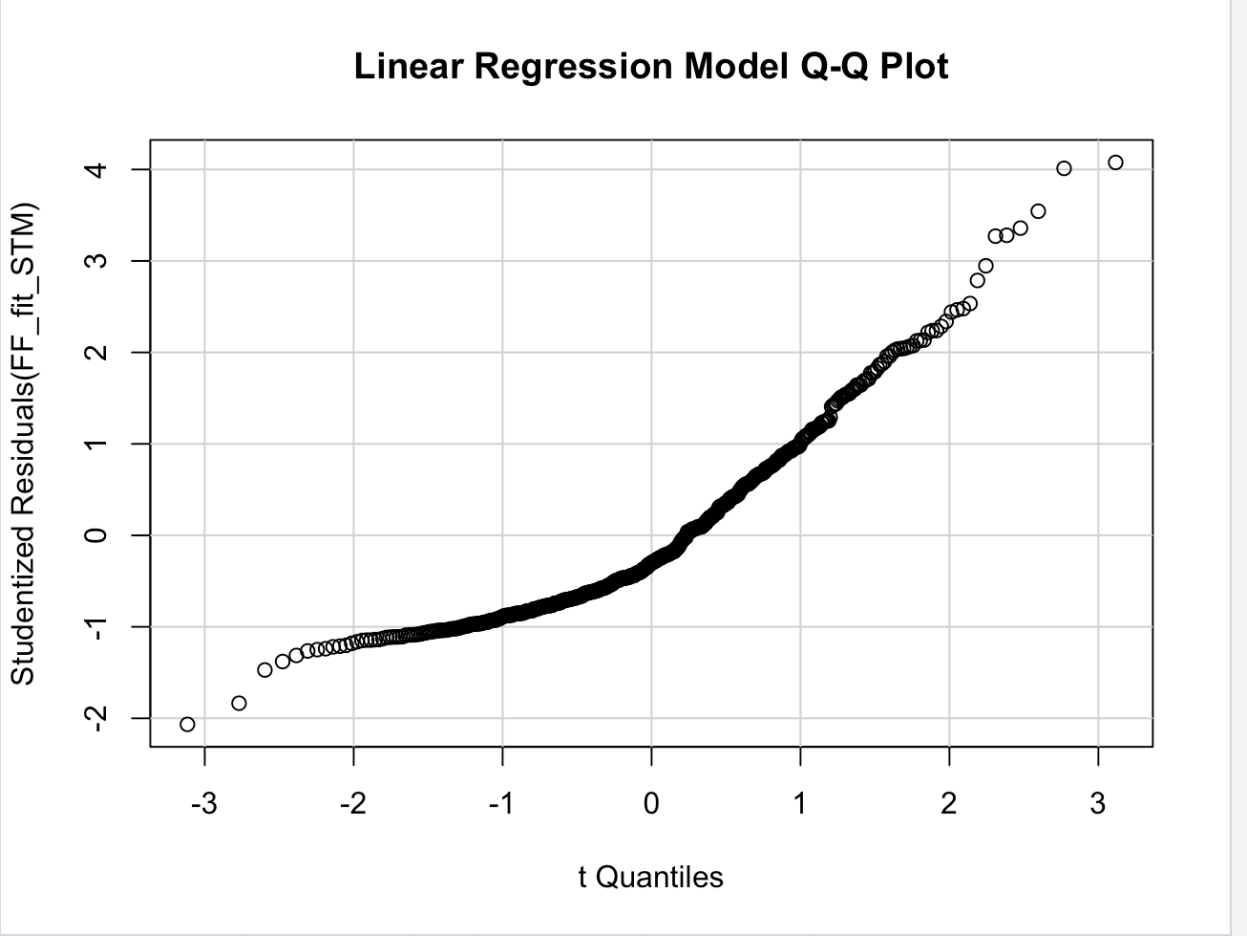
#Interpretation:
#In the normal Q-Q plot, there are too many points that deviate from the 45 degree diagonal- the dataset violates the normality assumption.
#In the residual VS fitted plot, the regression curve is flat.
#In the Scale-Location plot, data points are not randomly around the horizontal line, but align along a convex curve that suggests the dataset doesnt have homoscedasticity.
#The Residuals VS Leverage plot identifies several high-leveraged points.

#Enhanced Approach
#Normality
#Enhanced Q-Q Plot
par(mfrow=c(1,1))
qqPlot(FF_fit_STFWI,labels=row.names(ForestfiresData), id.method="identify", simulate=TRUE,

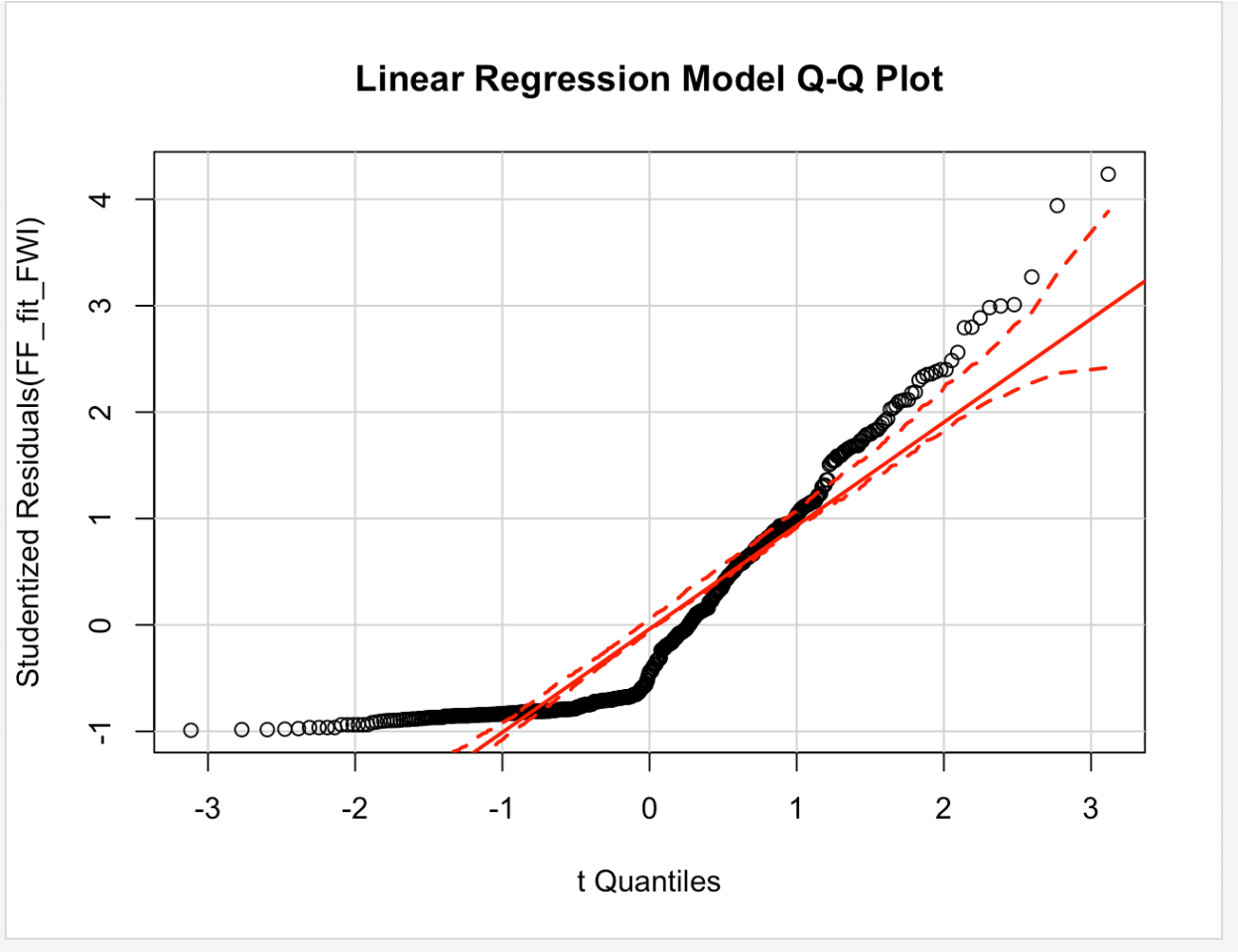
main="Linear Regression Model Q-Q Plot")



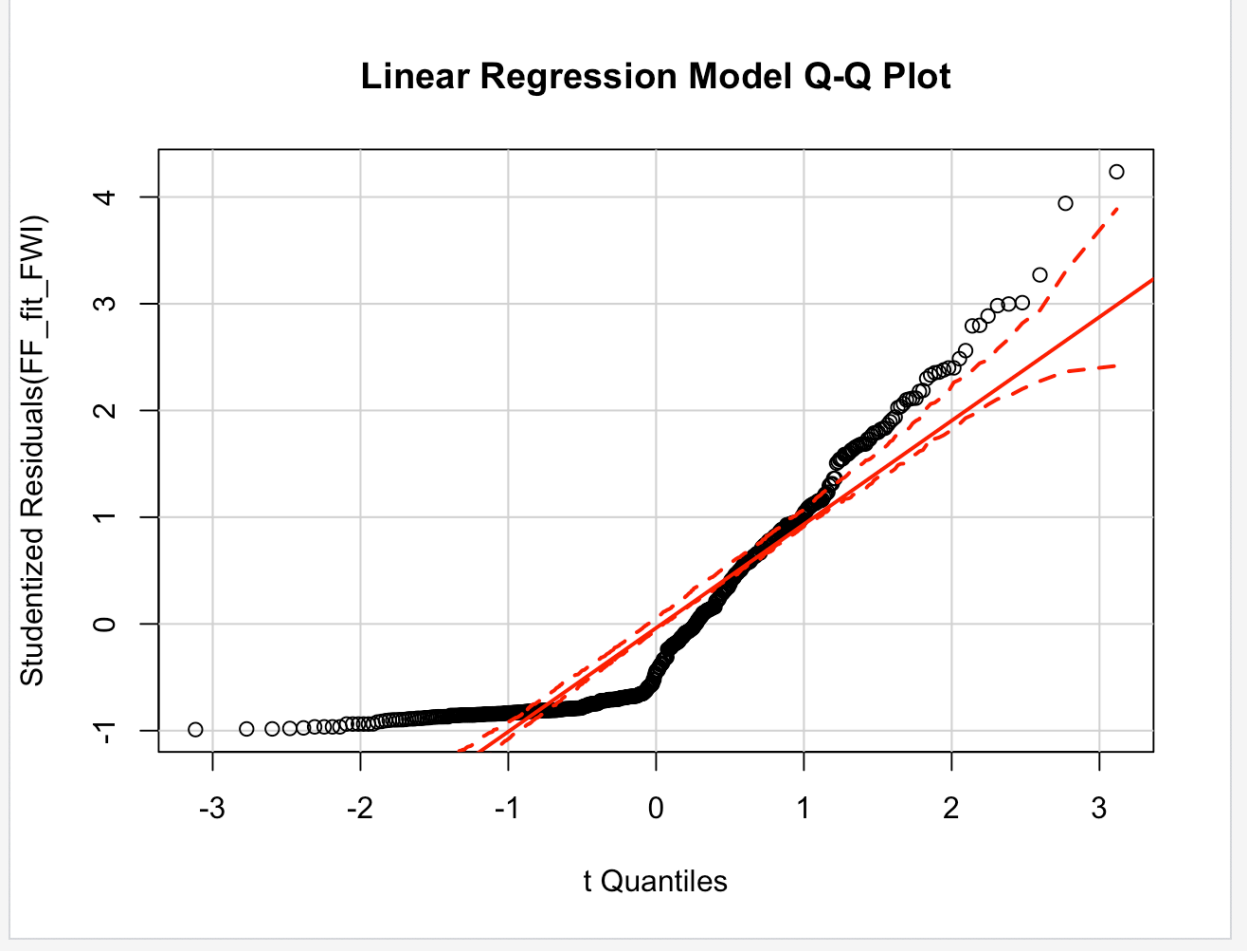
qqPlot(FF_fit_STM,labels=row.names(ForestfiresData), id.method="identify", simulate=TRUE,
main="Linear Regression Model Q-Q Plot")




```
qqPlot(FF_fit_FWI,labels=row.names(ForestfiresData), id.method="identify", simulate=TRUE,
      main="Linear Regression Model Q-Q Plot")
```




```
qqPlot(FF_fit_M,labels=row.names(ForestfiresData), id.method="identify", simulate=TRUE,
      main="Linear Regression Model Q-Q Plot")
```

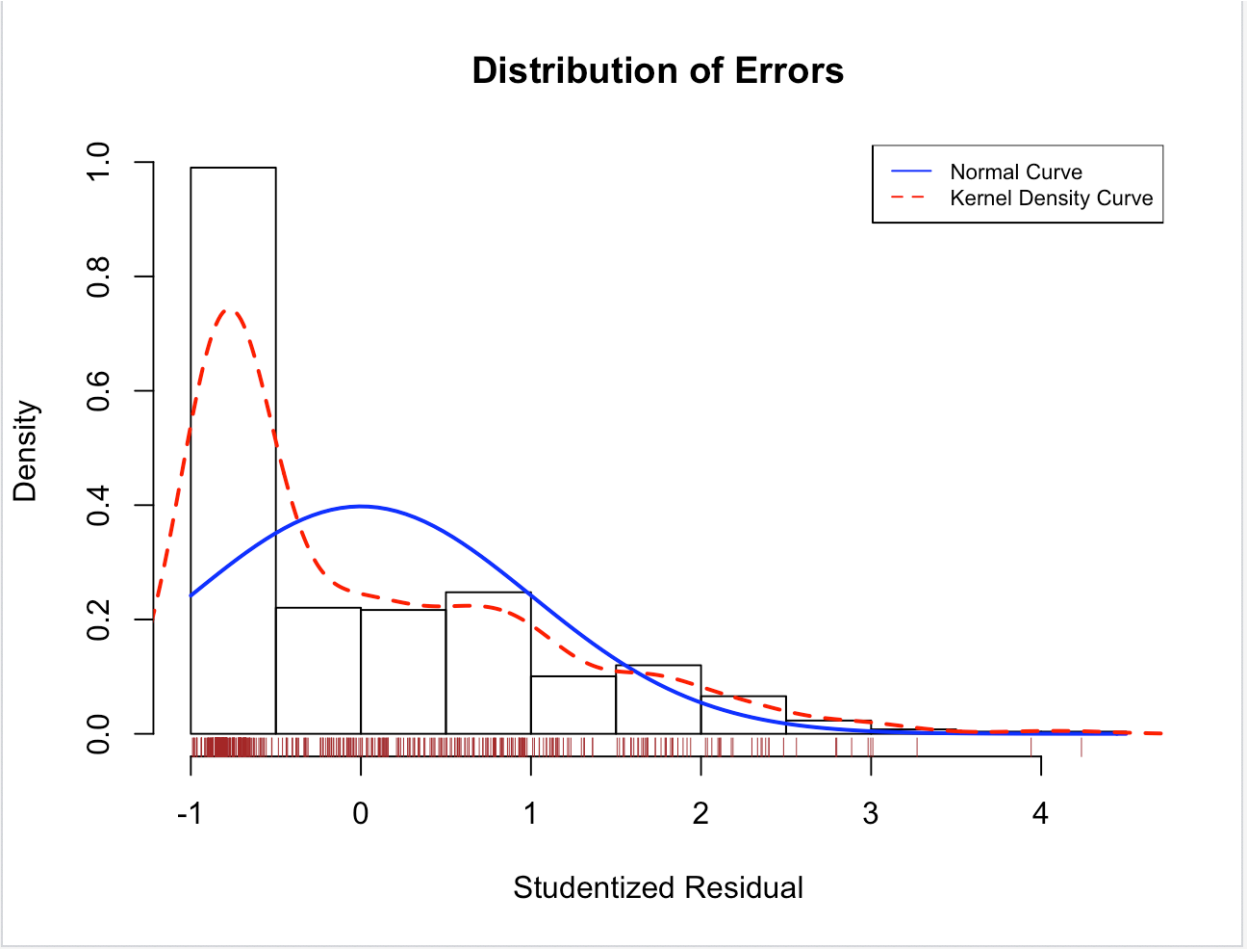


#Interpretation:
#Besides the four dummy variables, ISI and FFMC don't have linearity with respect to the area.

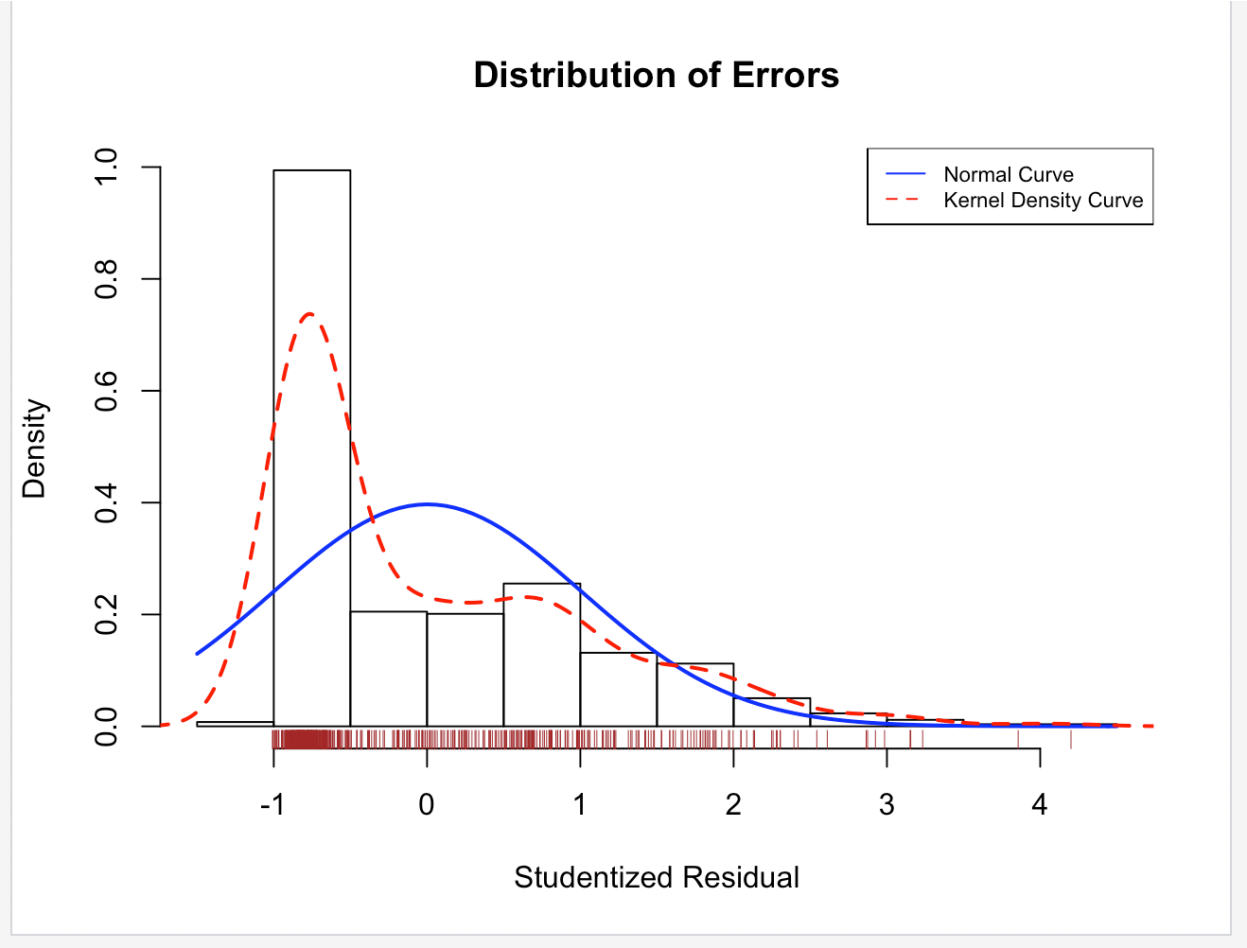
```
# Studentized Residuals Distribution
residualPlot = function(fit, nbreaks = 10){
  z = rstudent(fit)
  hist(z, breaks = nbreaks, freq = FALSE, xlab = "Studentized Residual", main = "Distribution of
Errors")
  rug(jitter(z), col="brown")
  curve(dnorm(x, mean=mean(z), sd=sd(z)), add=TRUE, col="blue", lwd=2)
  lines(density(z)$x, density(z)$y, col="red", lwd=2, lty=2)
  legend("topright", legend = c("Normal Curve", "Kernel Density Curve"), lty = 1:2, col=c("blue",
"red"), cex=.7 )
}
```

#Interpretation:
#we can clearly see the dependent variable is very much left skewed, though it has been already transformed by log Area.

# Since STFWI and STM contain dummy variables, the normal and Kernel Density curve don't apply to them.
residualPlot(FF_fit_FWI)



residualPlot(FF_fit_M)



#The two models are consistent with respect to the normal distribution

#Indepedence of Errors
#Durbin-Watson Test
durbinWatsonTest(FF_fit_STFWI)
durbinWatsonTest(FF_fit_STM)
durbinWatsonTest(FF_fit_FWI)

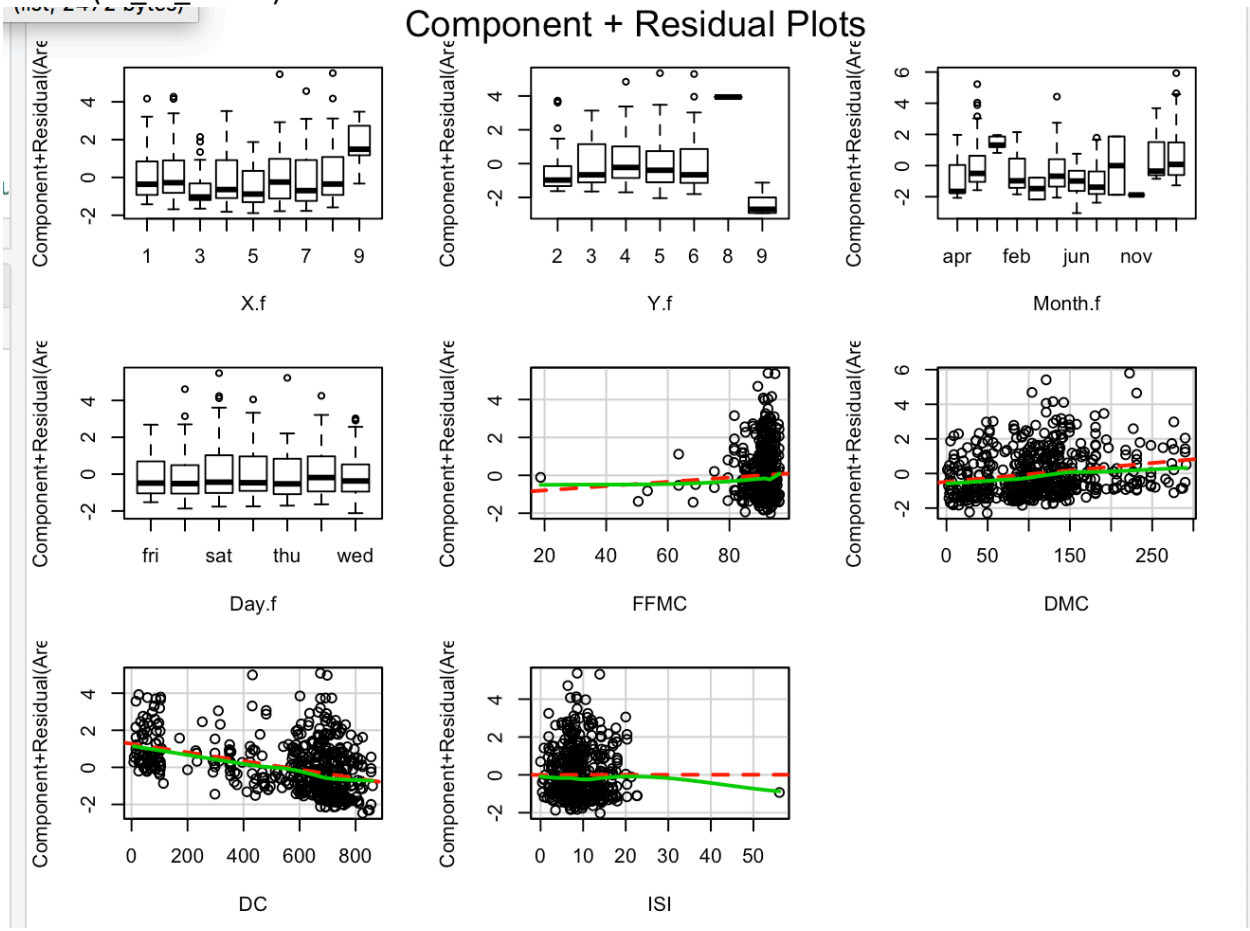
durbinWatsonTest(FF_fit_M)

```
> #Durbin-Watson Test
> durbinWatsonTest(FF_fit_STFWI)
 lag Autocorrelation D-W Statistic p-value
   1       0.5164365     0.9668493       0
 Alternative hypothesis: rho != 0
> durbinWatsonTest(FF_fit_STM)
 lag Autocorrelation D-W Statistic p-value
   1       0.5205424     0.9584846       0
 Alternative hypothesis: rho != 0
> durbinWatsonTest(FF_fit_FWI)
 lag Autocorrelation D-W Statistic p-value
   1       0.5397034       0.91882       0
 Alternative hypothesis: rho != 0
> durbinWatsonTest(FF_fit_M)
 lag Autocorrelation D-W Statistic p-value
   1       0.5365163     0.9245255       0
 Alternative hypothesis: rho != 0
>
```

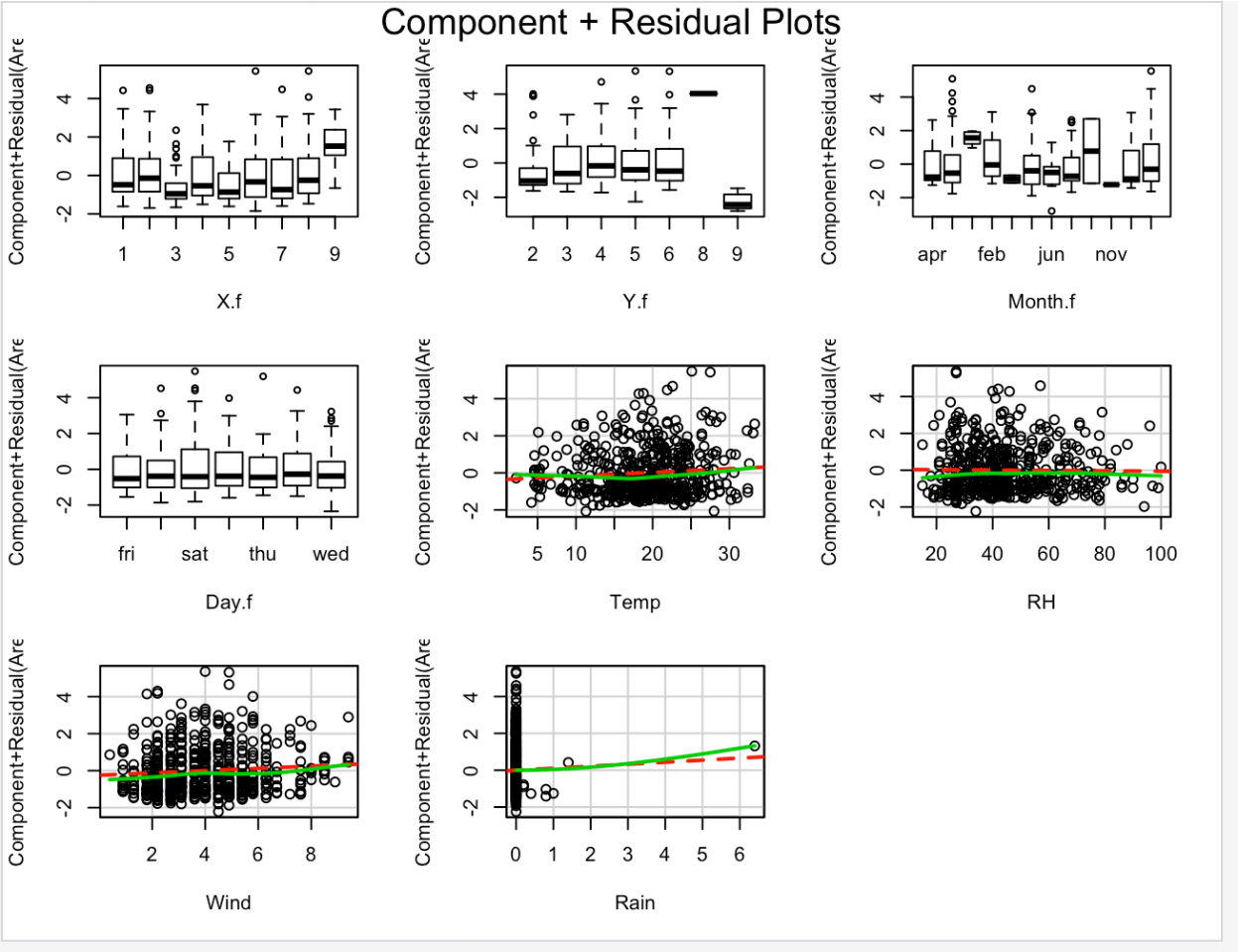
The durbinWatsonTest checks the residuals for autocorrelation. When the p-value is < 0.5, the residuals are significantly correlated whereas p > 0.05 provides no evidence of correlation.
All 4 models have autocorrelation feature.

#Linearity- Component plus residual plots

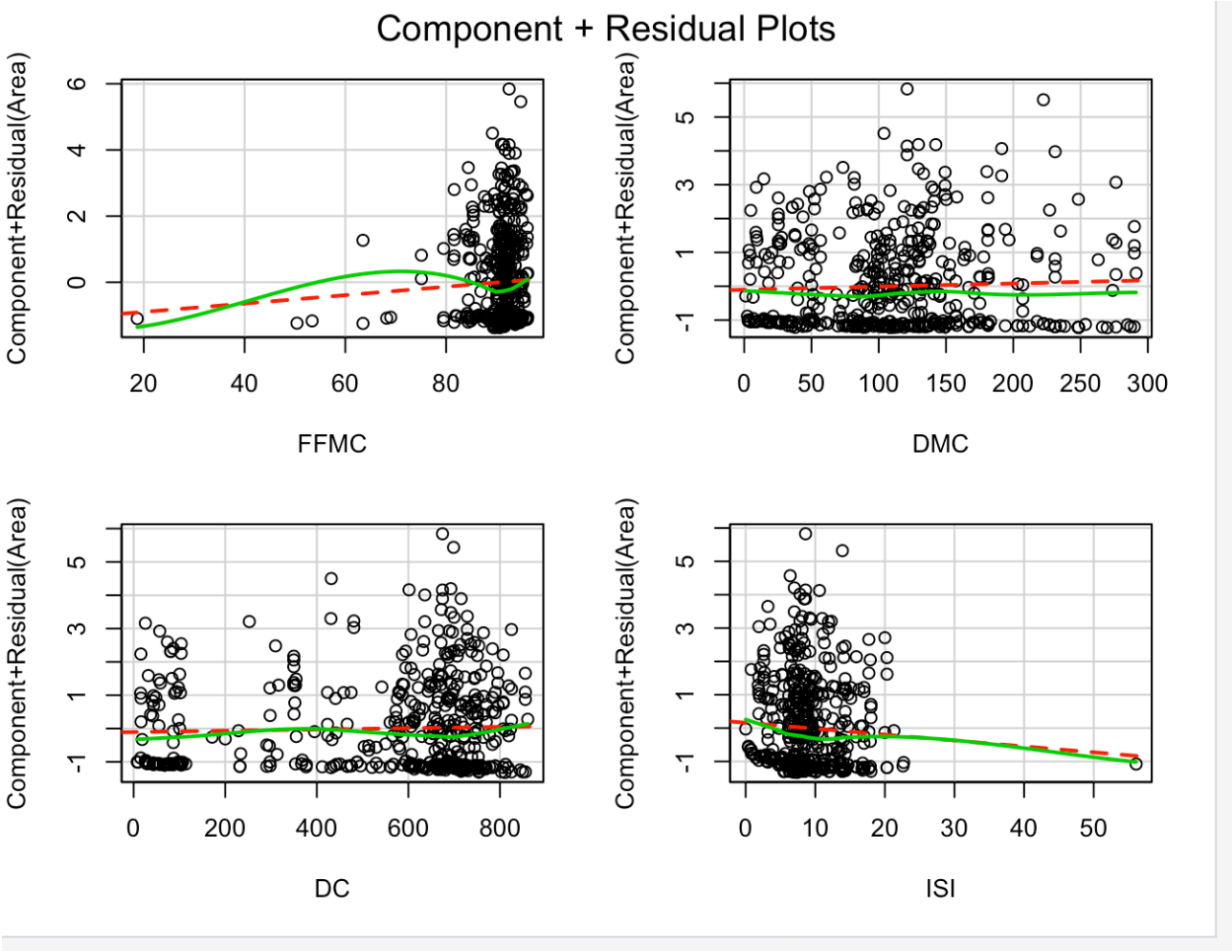
crPlots(FF_fit_STFWI)



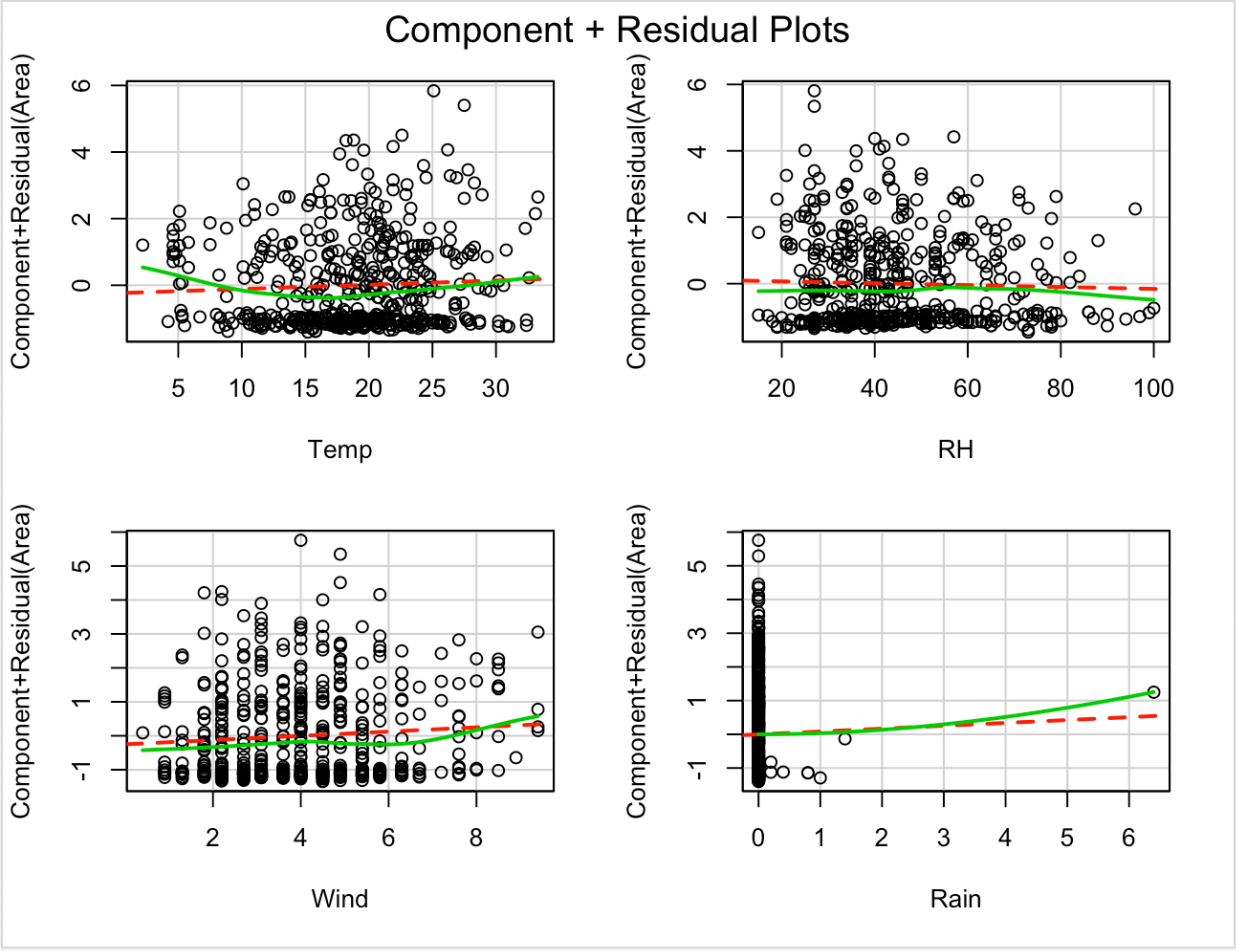
crPlots(FF_fit_STM)



crPlots(FF_fit_FWI)

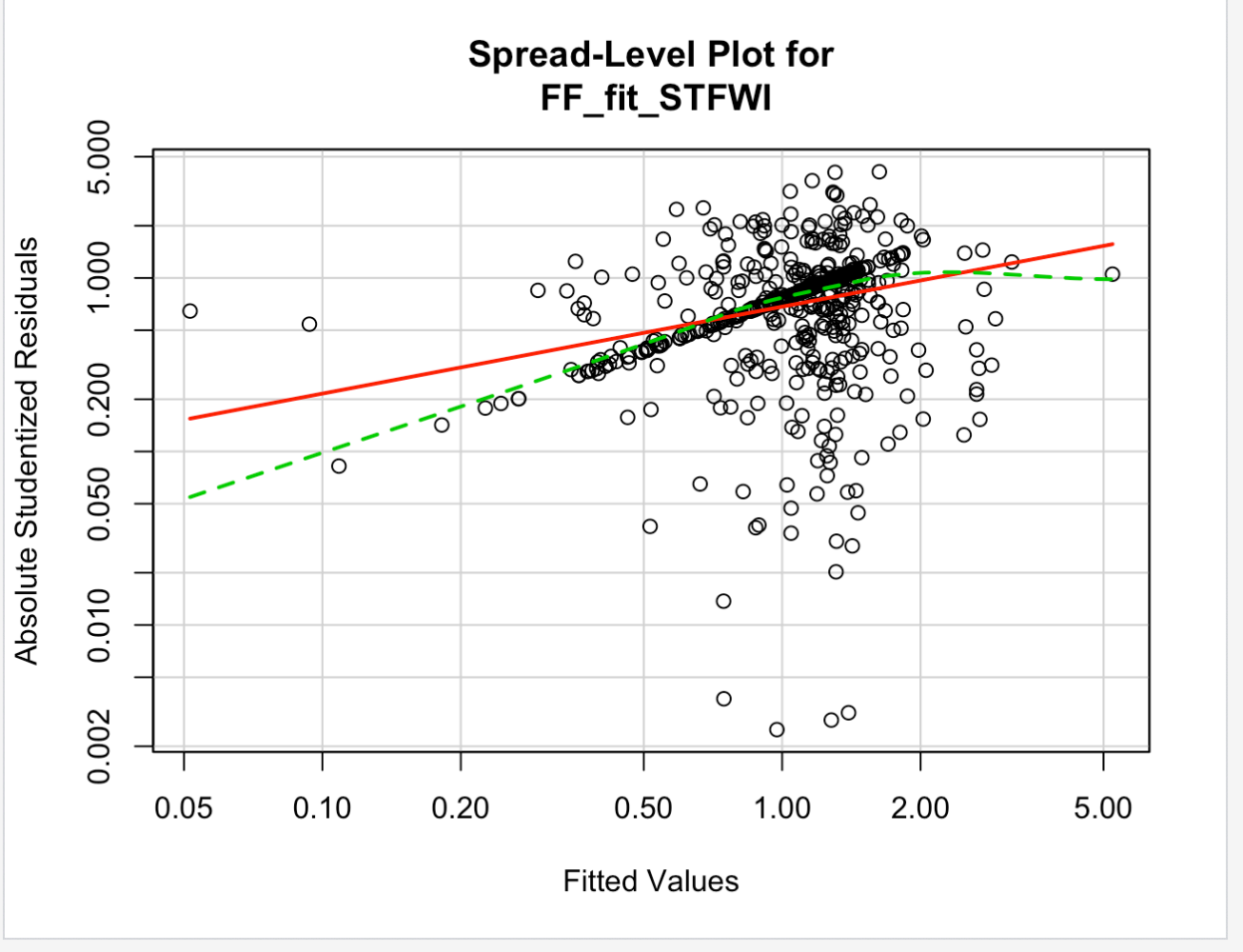


crPlots(FF_fit_M)

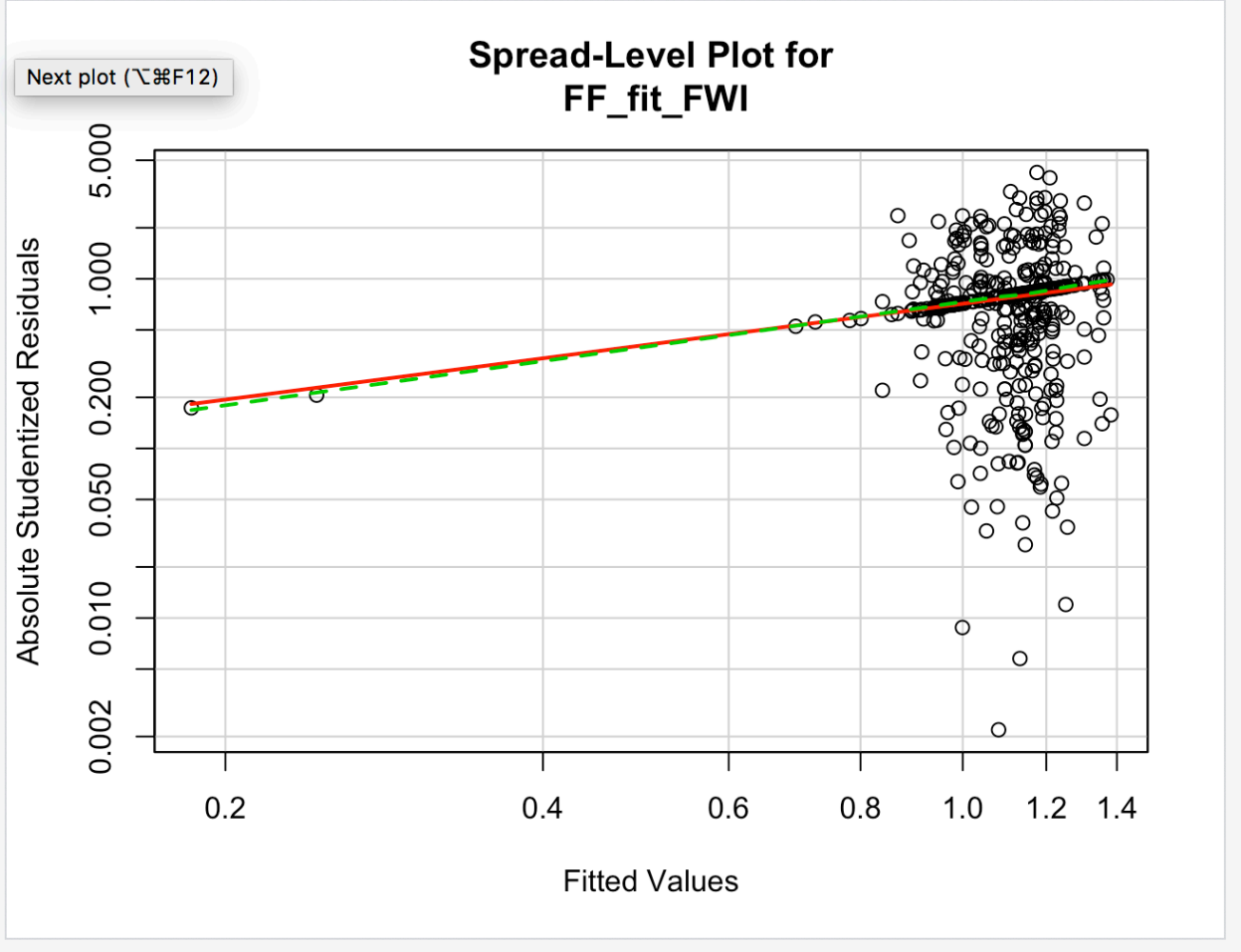


#Any nonlinearity in any of the above graphs suggest that we may have not adequately modeled the functional form of that predictor in the regression.
#From above graphs, we can confirm that we have met the linearity assumption.

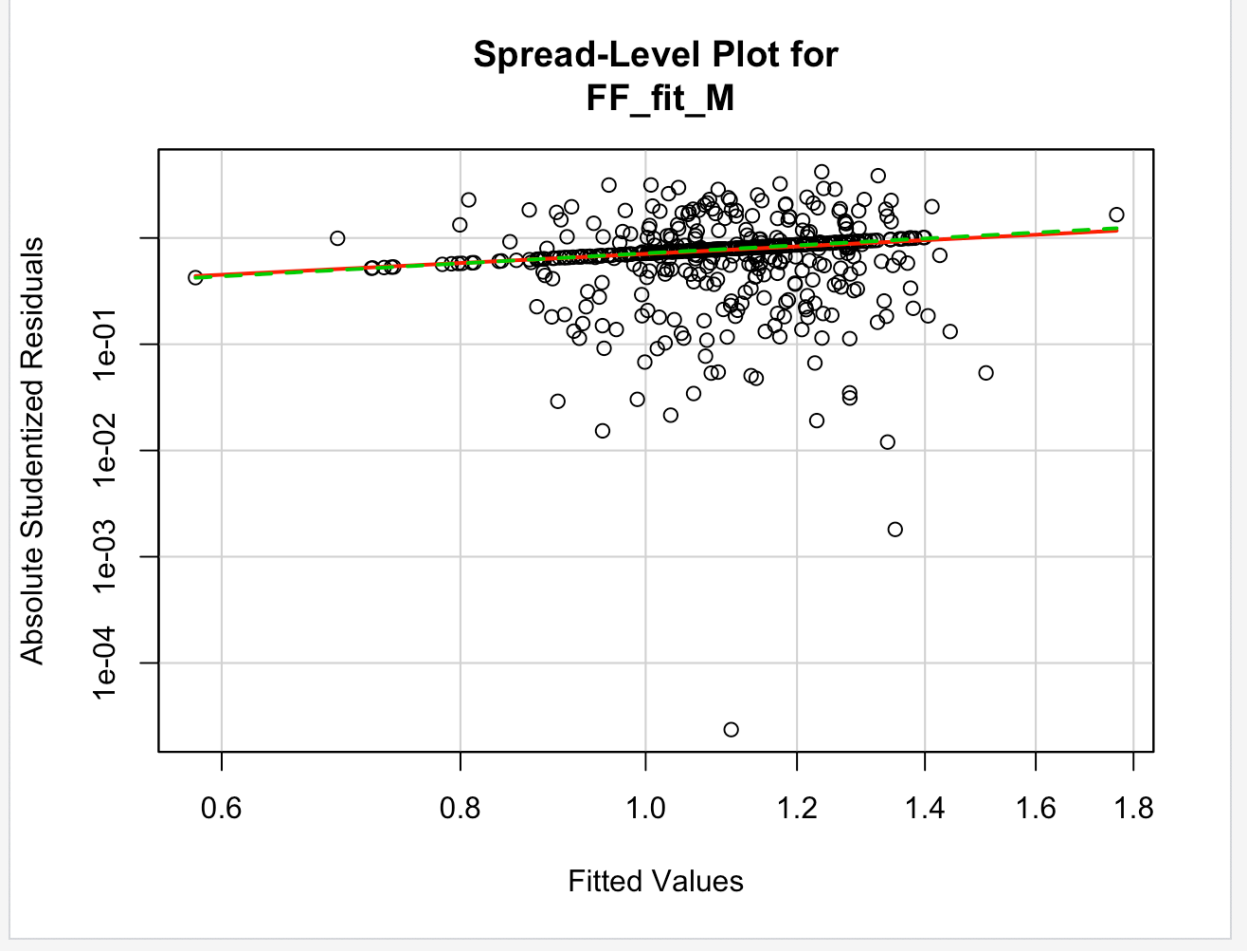
#Homoscedasticity
ncvTest(FF_fit_STFWI)
spreadLevelPlot(FF_fit_STFWI)




```
ncvTest(FF_fit_STM)
spreadLevelPlot(FF_fit_STM)

ncvTest(FF_fit_FWI)
spreadLevelPlot(FF_fit_FWI)
```




```
ncvTest(FF_fit_M)
spreadLevelPlot(FF_fit_M)
```



#In all the graphs, there are random points about the horizontal best fit line.
#If we had violated the assumption, we would see non-horizontal line.

```
# Global validation of linear model assumption
library(gvlma)
gvlma(FF_fit_STFWI)
gvlma(FF_fit_STM)
gvlma(FF_fit_FWI)
gvlma(FF_fit_M)
```

```
        ISI
  0.0001668


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = FF_fit_STFWI)

                     Value   p-value                   Decision
Global Stat        140.113 0.000e+00 Assumptions NOT satisfied!
Skewness           107.023 0.000e+00 Assumptions NOT satisfied!
Kurtosis            22.333 2.292e-06 Assumptions NOT satisfied!
Link Function        1.667 1.966e-01    Assumptions acceptable.
Heteroscedasticity   9.089 2.571e-03 Assumptions NOT satisfied!
>
```

#The spread level plot has a positively correlated curve.
which means the variances increases as the variables increase.

# Multicollinearity
vif(FF_fit_STFWI)
sqrt(vif(FF_fit_STFWI))>2

```
> # Multicollinearity
> vif(FF_fit_STFWI)
             GVIF Df GVIF^(1/(2*Df))
X.f     14.298251  8        1.180878
Y.f     12.094143  6        1.230877
Month.f 92.319311 11        1.228376
Day.f    1.432947  6        1.030432
FFMC     2.156833  1        1.468616
DMC      3.857272  1        1.963994
DC      27.382654  1        5.232844
ISI      1.795823  1        1.340083
> sqrt(vif(FF_fit_STFWI))>2
         GVIF    Df GVIF^(1/(2*Df))
X.f      TRUE  TRUE           FALSE
Y.f      TRUE  TRUE           FALSE
Month.f  TRUE  TRUE           FALSE
Day.f   FALSE  TRUE           FALSE
FFMC    FALSE FALSE           FALSE
```

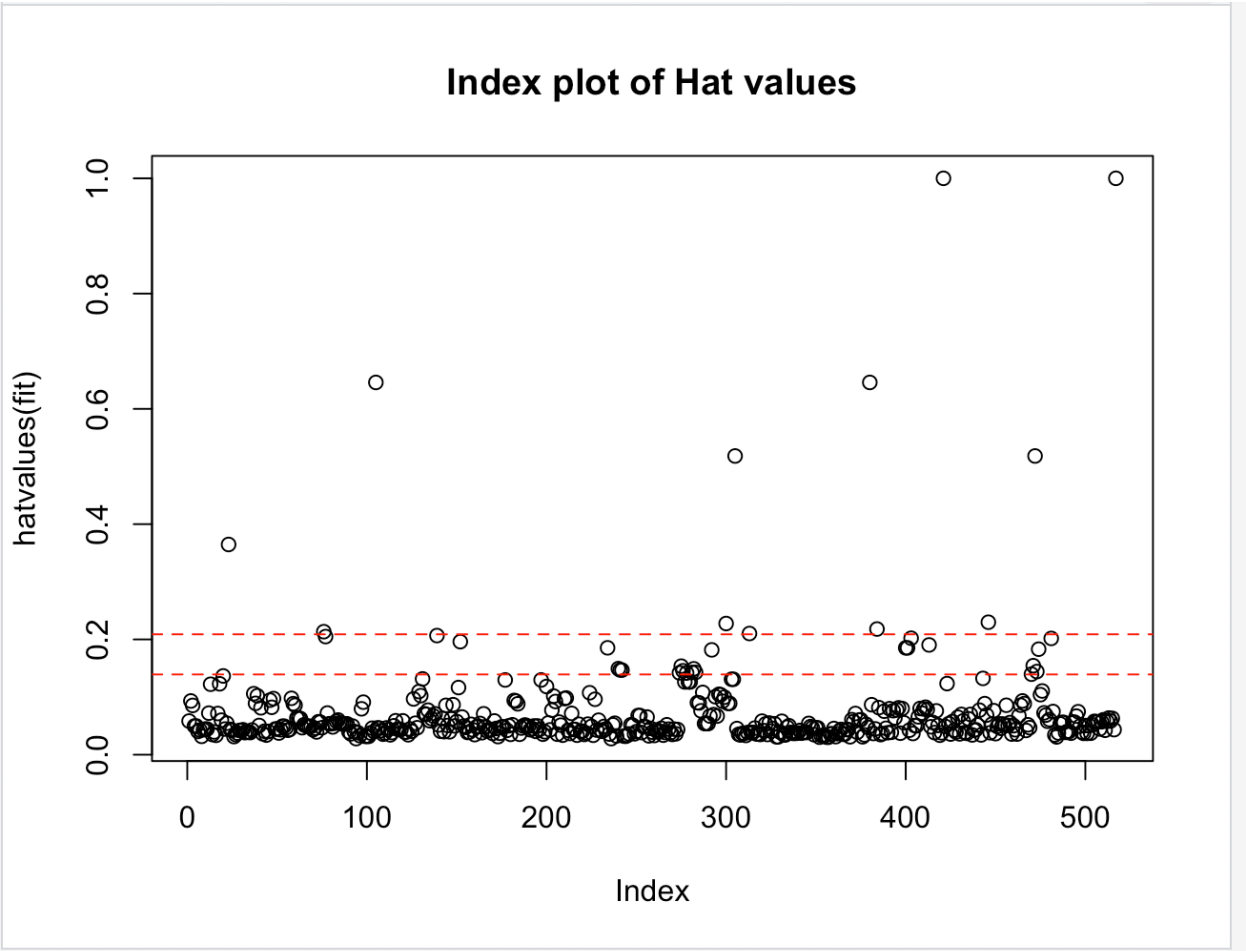
vif(FF_fit_STM)
sqrt(vif(FF_fit_STM))>2

```
vif(FF_fit_FWI)
sqrt(vif(FF_fit_FWI))>2

vif(FF_fit_M)
sqrt(vif(FF_fit_M))>2


#Only the DC variable has the problem of Multicolinearity


#Question 2.4:
# Identify unusual observations and take corrective measures

#Outlier Test

outlierTest(FF_fit_STFWI)
outlierTest(FF_fit_STM)
outlierTest(FF_fit_FWI)
outlierTest(FF_fit_M)
```

```
DC        TRUE  FALSE            TRUE
ISI      FALSE  FALSE           FALSE
> outlierTest(FF_fit_STFWI)
    rstudent unadjusted p-value Bonferonni p
239 4.094275         4.9690e-05     0.025590
416 4.063408         5.6491e-05     0.029093
> outlierTest(FF_fit_STM)
    rstudent unadjusted p-value Bonferonni p
239 4.076742         5.3451e-05     0.027527
416 4.011896         6.9850e-05     0.035973
> outlierTest(FF_fit_FWI)
    rstudent unadjusted p-value Bonferonni p
239 4.236704         2.6911e-05     0.013913
416 3.940781         9.2520e-05     0.047833
> outlierTest(FF_fit_M)
    rstudent unadjusted p-value Bonferonni p
239  4.20067         3.1404e-05     0.016236
>
```

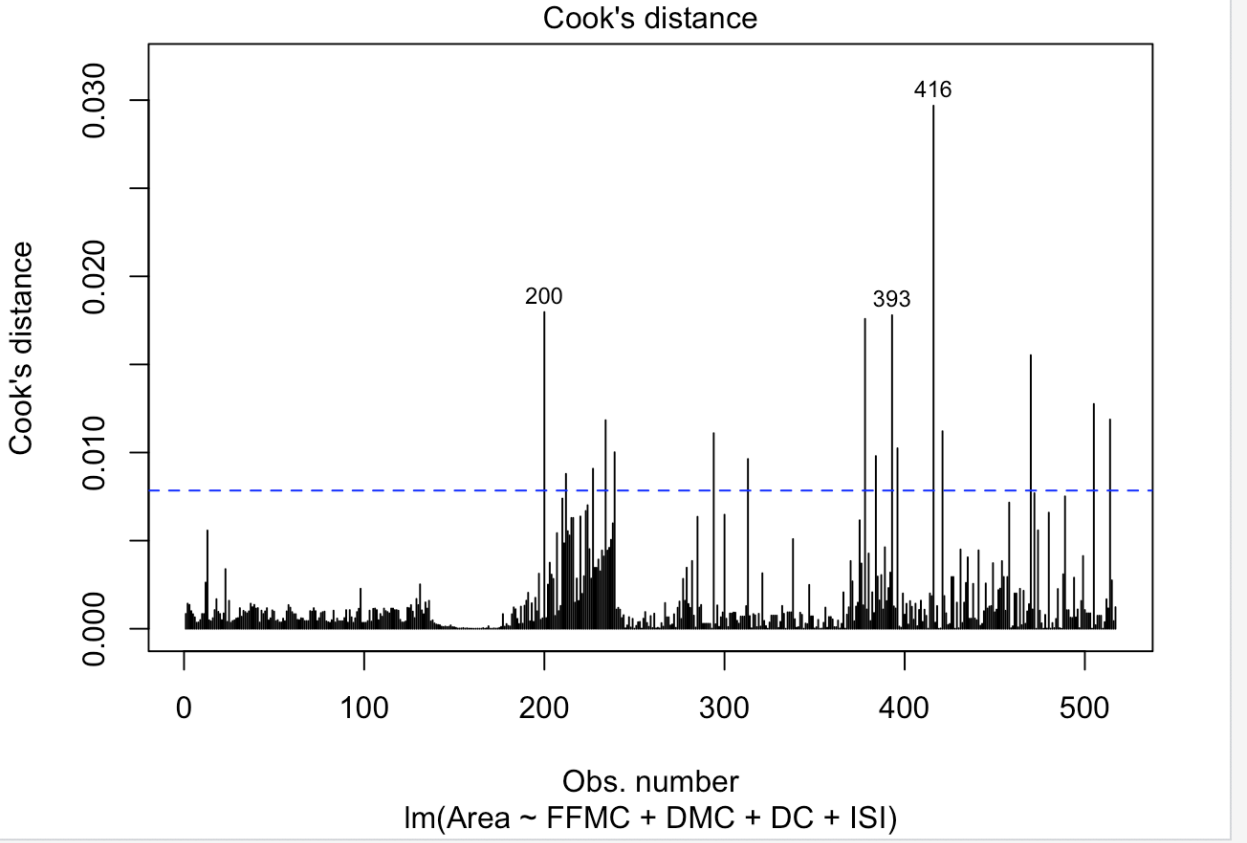
#The results of these models show that points 239 and 416 are outliers.

```
hat.plot<-function(fit){
  p<-length(coefficients(fit))
  n<-length(fitted(fit))
  plot(hatvalues(fit),main="Index plot of Hat values")
  abline(h=c(2,3)*p/n,col="red",lty=2)
  identify(1:n,hatvalues(fit),names(hatvalues(fit)))
}
hat.plot(FF_fit_STFWI)
hat.plot(FF_fit_STM)
hat.plot(FF_fit_FWI)
hat.plot(FF_fit_M)
```

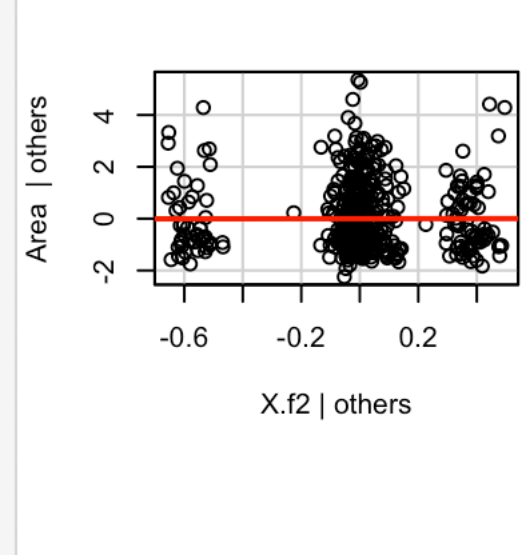



```
#Influential Observations
#Cook's distance
Dplot=function(fit,data){
  cutoff<-4/(nrow(data)-length(fit$coefficients)-2)
  plot(fit,which=4,cook.levels=cutoff)
  abline(h=cutoff,lty=2,col="blue")
}
```

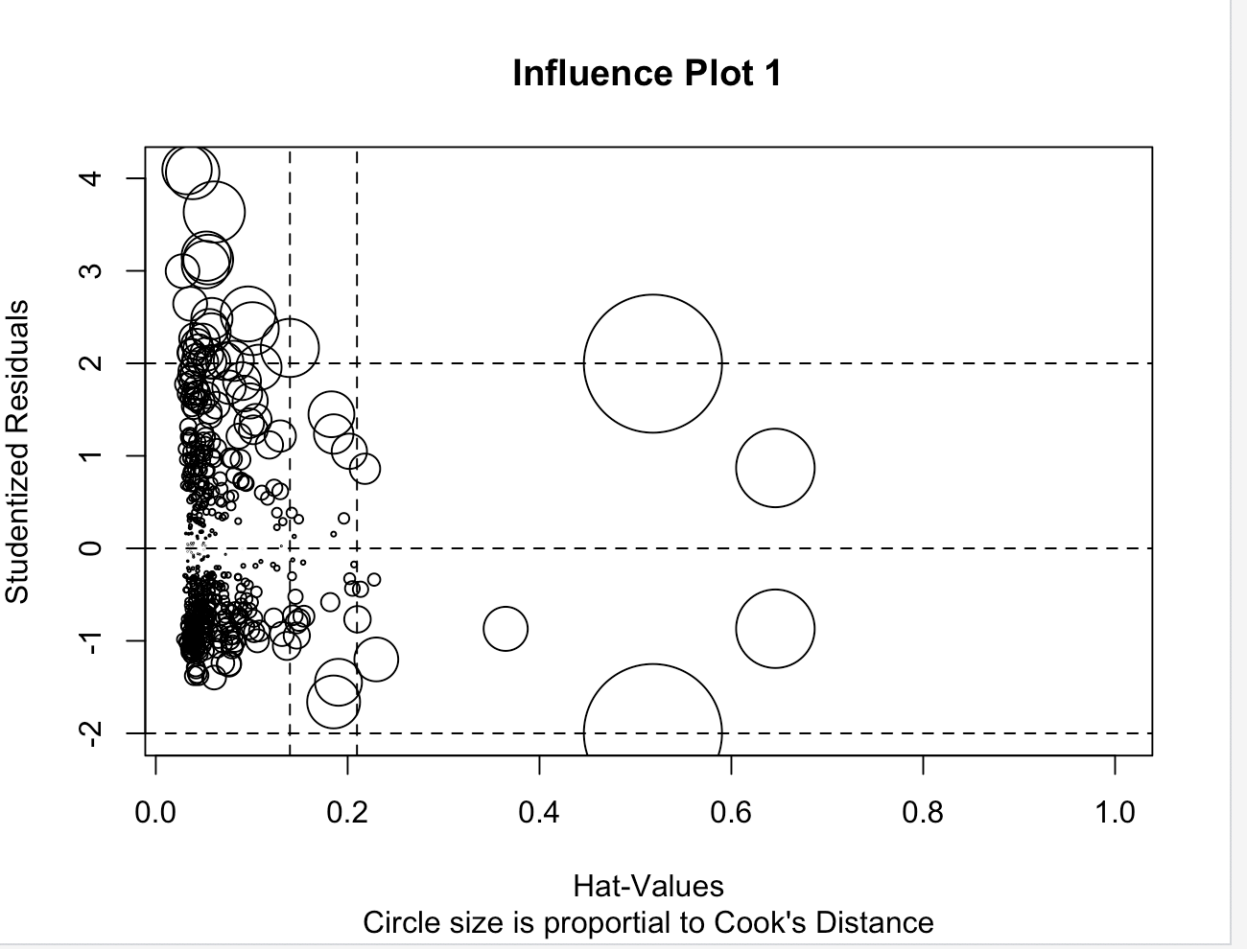
```
Dplot(FF_fit_STFWI,Forest_Fires_Data)
Dplot(FF_fit_STM,Forest_Fires_Data)
Dplot(FF_fit_FWI,Forest_Fires_Data)
Dplot(FF_fit_M,Forest_Fires_Data)
```



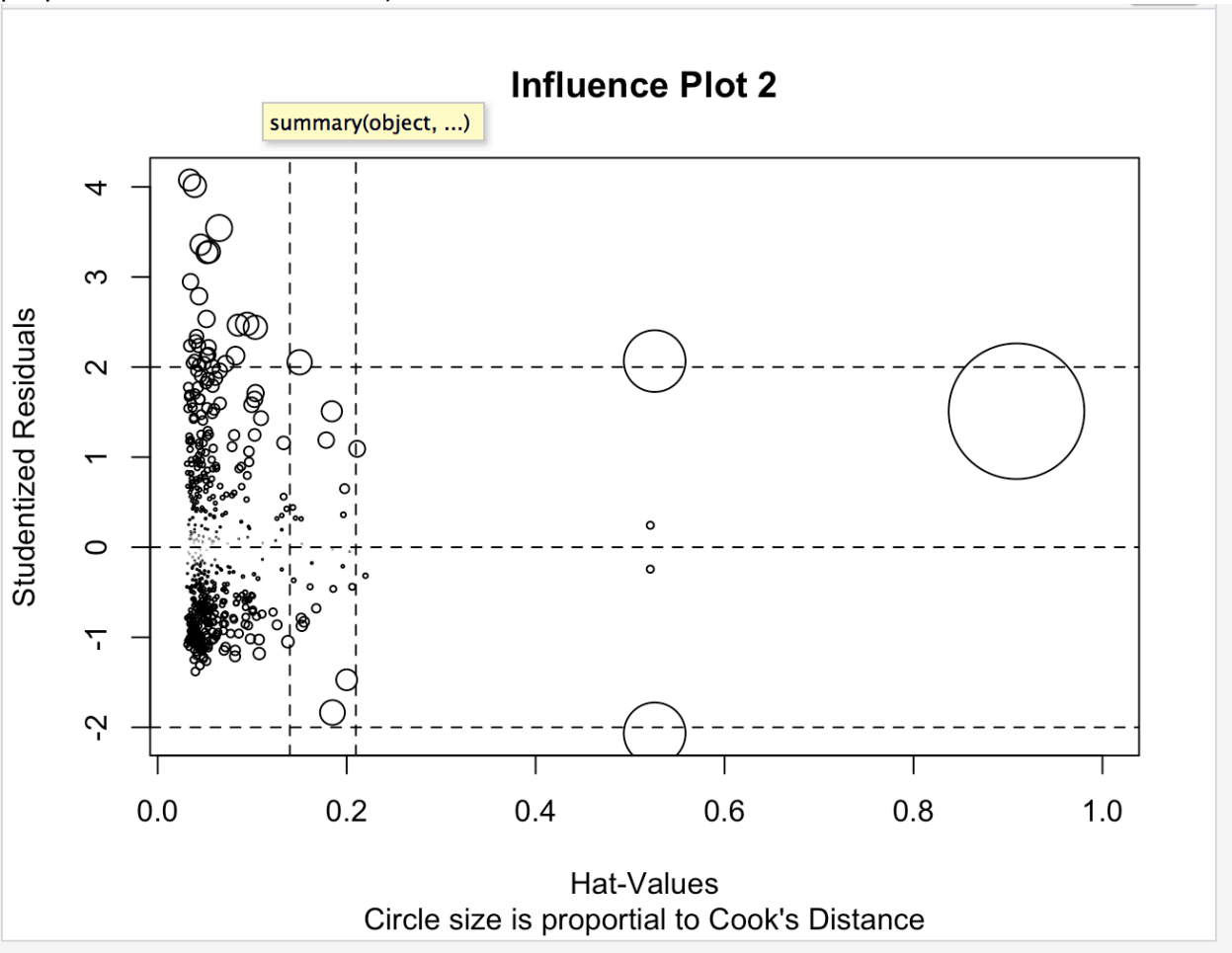

```
#Added variable plots
avPlots(FF_fit_STFWI, ask=FALSE, id.method ="identify",onepage=TRUE )
avPlots(FF_fit_STM, ask=FALSE, id.method ="identify",onepage=TRUE )
avPlots(FF_fit_FWI, ask=FALSE,id.method ="identify", onepage=TRUE )
avPlots(FF_fit_M, ask=FALSE,id.method ="identify", onepage=TRUE )
```



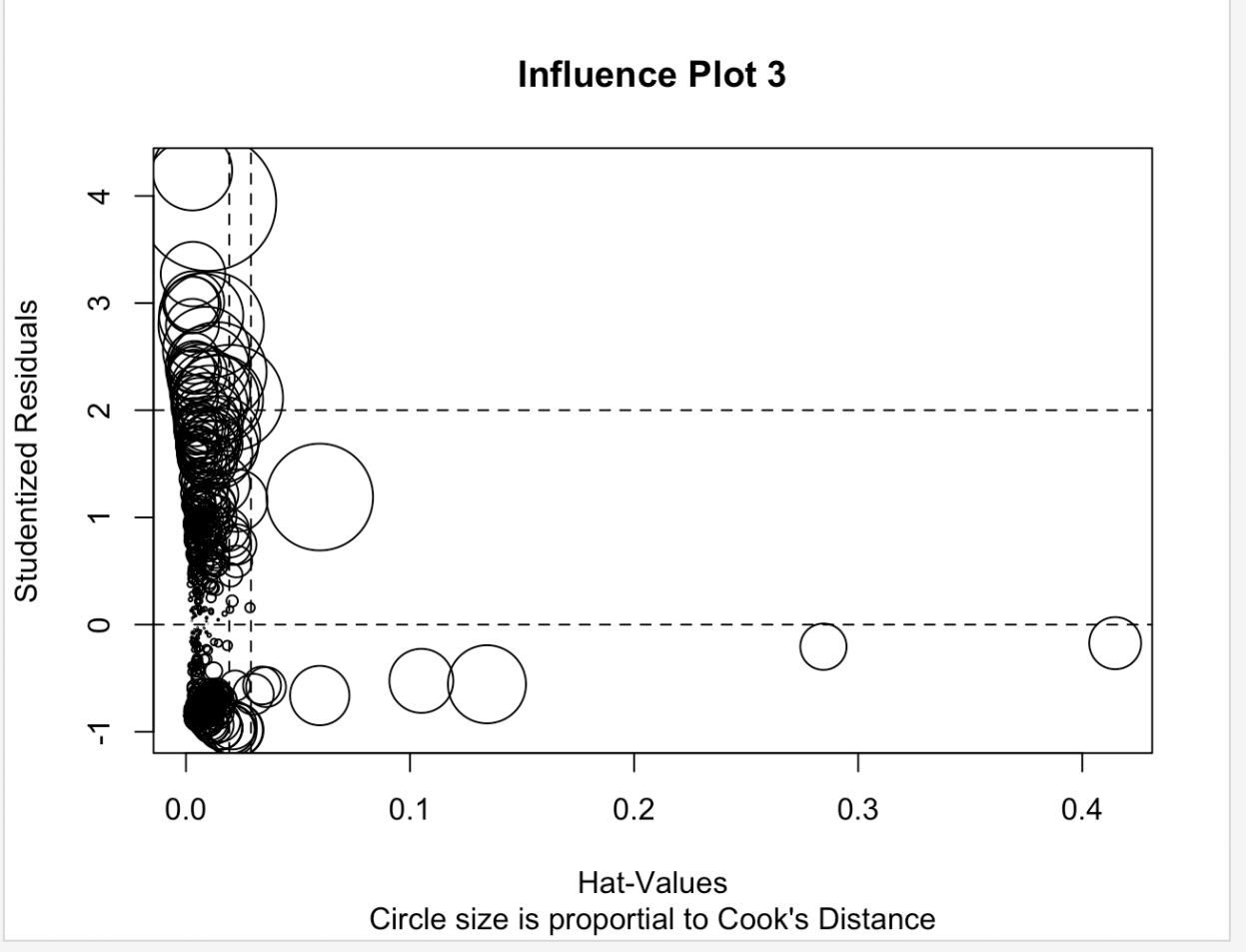
#Combined information by influence plot
influencePlot(FF_fit_STFWI,id.method="identify", main="Influence Plot 1", sub="Circle size is proportial to Cook's Distance")



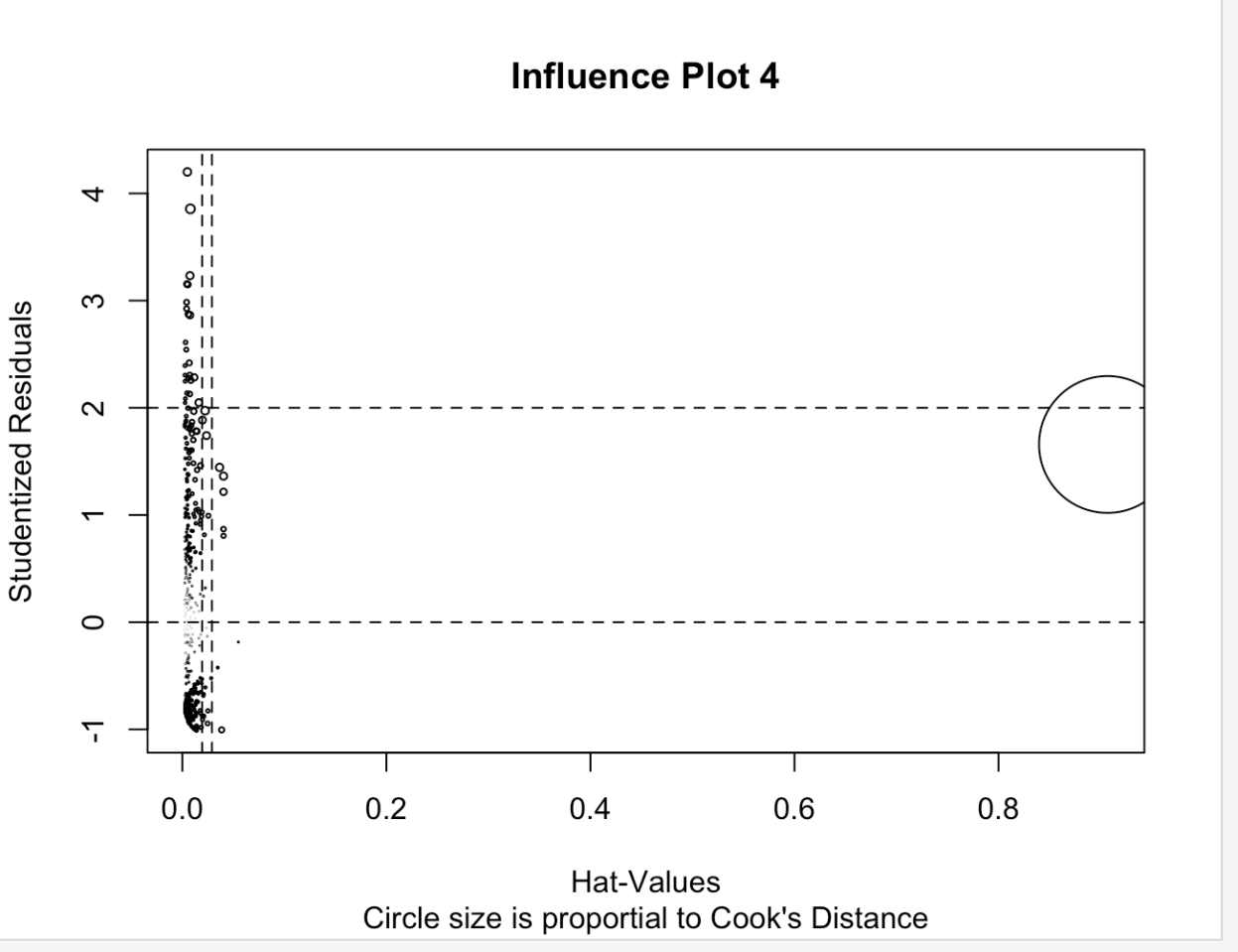
influencePlot(FF_fit_STM,id.method="identify", main="Influence Plot 2", sub="Circle size is proportial to Cook's Distance")



```
influencePlot(FF_fit_FWI,id.method="identify", main="Influence Plot 3", sub="Circle size is proportial to Cook's Distance")
```



influencePlot(FF_fit_M,id.method="identify", main="Influence Plot 4", sub="Circle size is proportial to Cook's Distance")

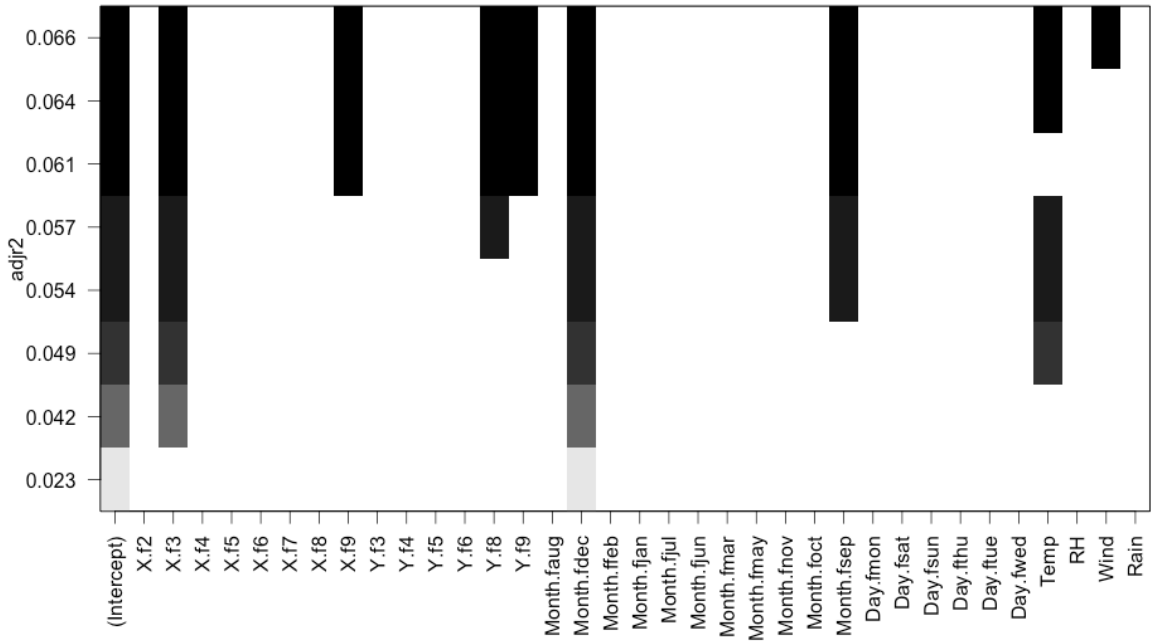


#Corrective measures:
#Deleting outliers
Forest_Fires_Data_Deleted = Forest_Fires_Data[c(-105, -239, -305, -416, -421, -517, -472, -480, -500),]
Forest_Fires_Data_Deleted

```
# A tibble: 508 x 17
      X     Y Month Day    FFMC   DMC    DC   ISI  Temp    RH  Wind  Rain  Area X.f   Y.f
  <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct>
1  7.00  5.00 mar   fri    86.2  26.2  94.3  5.10  8.20  51.0 6.70  0         0 7     5
2  7.00  4.00 oct   tue    90.6  35.4 669    6.70 18.0   33.0 0.900 0         0 7     4
3  7.00  4.00 oct   sat    90.6  43.7 687    6.70 14.6   33.0 1.30  0         0 7     4
4  8.00  6.00 mar   fri    91.7  33.3  77.5  9.00  8.30  97.0 4.00  0.200     0 8     6
5  8.00  6.00 mar   sun    89.3  51.3 102    9.60 11.4   99.0 1.80  0         0 8     6
6  8.00  6.00 aug   sun    92.3  85.3 488   14.7  22.2   29.0 5.40  0         0 8     6
7  8.00  6.00 aug   mon    92.3  88.9 496    8.50 24.1   27.0 3.10  0         0 8     6
8  8.00  6.00 aug   mon    91.5 145   608   10.7   8.00  86.0 2.20  0         0 8     6
9  8.00  6.00 sep   tue    91.0 130   693    7.00 13.1   63.0 5.40  0         0 8     6
10 7.00  5.00 sep   sat    92.5  88.0 699    7.10 22.8   40.0 4.00  0         0 7     5
#     with 498 more rows, and 2 more variables: Month_f <fct>, Day_f <fct>
```

```
FF_fit_STFWI_Deleted = lm(Area ~ X.f + Y.f + Month.f + Day.f + FFMC + DMC + DC + ISI,
data=Forest_Fires_Data_Deleted) summary(FF_fit_STFWI_Deleted)
D.plot(Model_STFWI_Deleted, Model_STFWI_Deleted)

FF_fit_STM_Deleted = lm(Area ~ X.f + Y.f + Month.f + Day.f + Temp + RH + Wind + Rain,
data=Forest_Fires_Data_Deleted) summary(FF_fit_STM_Deleted)

FF_fit_FWI_Deleted = lm(Area ~ FFMC + DMC + DC + ISI, data=Forest_Fires_Data_Deleted)
summary(FF_fit_FWI_Deleted)

FF_fit_M_Deleted = lm(Area ~ Temp + RH + Wind + Rain, data=Forest_Fires_Data_Deleted)
summary(FF_fit_M_Deleted)



Forest_Fires_Data_Transformed = Forest_Fires_Data_Transformed$Area+1
summary(powerTransform(ForestfireData_Trans$Area))
Forest_Fires_Data_Transformed[,1]=Forest_Fires_Data_Transformed[,1]^(-0.7143)
FF_fit_STFWI_Trans = lm(Area ~ X.f + Y.f + Month.f + Day.f + FFMC + DMC + DC + ISI,
data=Forest_Fires_Data_Transformed)
summary(powerTransform(FF_fit_STFWI_Trans))



FF_fit_STM_Trans = lm(Area ~ X.f + Y.f + Month.f + Day.f + Temp + RH + Wind + Rain,
data=Forest_Fires_Data_Transformed)
summary(powerTransform(FF_fit_STM_Trans))


FF_fit_FWI_Trans = lm(Area ~ FFMC + DMC + DC + ISI, data=Forest_Fires_Data_Transformed)
summary(powerTransform(FF_fit_FWI_Trans))


FF_fit_M_Trans = lm(Area ~ Temp + RH + Wind + Rain, data=Forest_Fires_Data_Transformed)
summary(powerTransform(FF_fit_M_Trans))


#Deleting outliers, performing log transformations and removing influential observations can
be executed as corrective measures.

#Question 5:
#Selecting the best regression model
anova(FF_fit_STFWI_Trans, FF_fit_STM_Trans, FF_fit_FWI_Trans, FF_fit_M_Trans)
AIC(FF_fit_STFWI_Trans, FF_fit_STM_Trans, FF_fit_FWI_Trans, FF_fit_M_Trans)
```

```
summary(FF_fit_STM_Trans)
summary(FF_fit_M_Trans)

#Variable selection
stepAIC(FF_fit_STM_Trans, direction = "backward")
FF_fit_STM_Back = lm(formula = Area ~ X.f + Y.f + Month.f + Wind, data = ForestfireData_Trans)
# The predicted variables have been refined

stepAIC(FF_fit_STM_Trans, direction = "forward")
FF_fit_STM_Forward = lm(formula = Area ~ X.f + Y.f + Month.f + Day.f + Temp + RH + Wind +
Rain, data = ForestfireData_Trans)
# The variables have not changed.

stepAIC(FF_fit_STM_Trans, direction = "both")
```



#Comparing the models, the backward model is better.

#Interpretation of the results
#According to the results of the three kinds of stepwise methods, Forward, Backward and stepwise,
#The RSS of the models , both in the forward and backward direction, were pretty close.
#There was a lower AIC model for the backward method which gave it the advantage of being a better model.
#Therefore the lower AIC value of the backward model makes it the optimal model for this dataset

The Optimal Model would be : Area ~ X.f + Y.f + Month.f + Wind