Group15 - Homework 03

Hansika Karkera, Vrunda Shah

2/28/2020

Problem 01

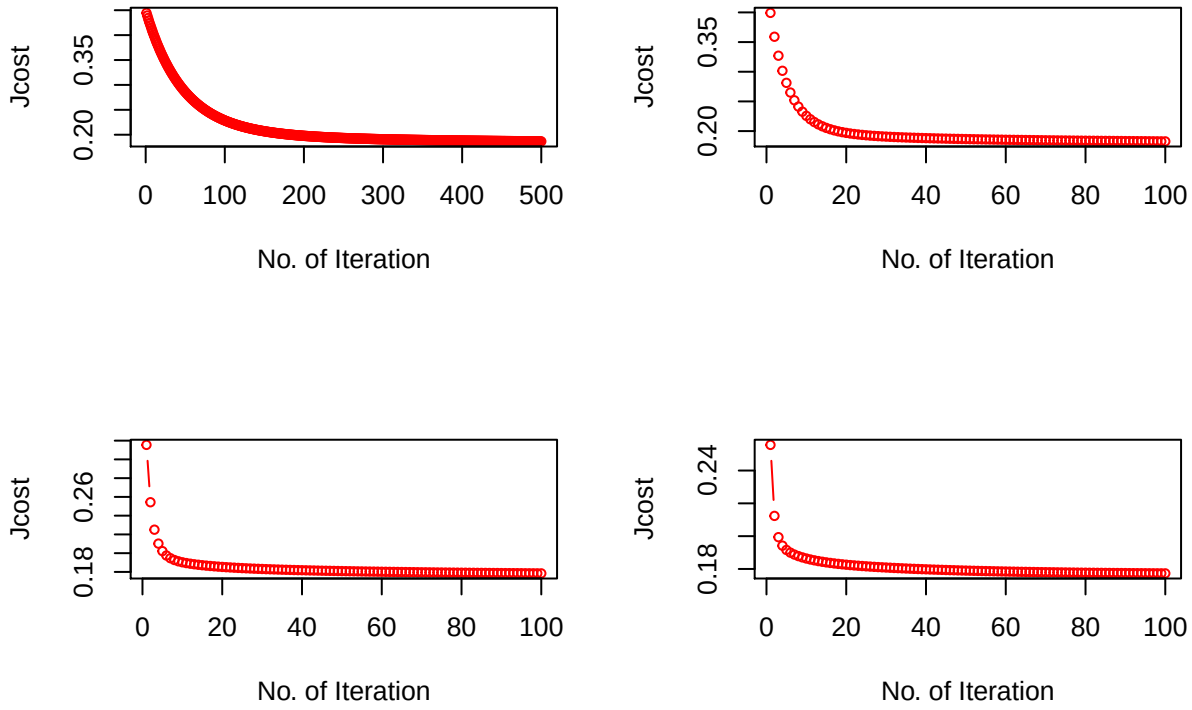
```
library(readxl)
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 3.6.2
```

```
## Loading required package: tidyr
```

```
# Read csv file
concrete_df <- read.csv("concrete.csv")
# Standardize all variables
c_df <- as.data.frame(scale(concrete_df[,]))
# Split data set into training and validation data
split <- initial_split(c_df,prop = 0.60)
train_df <- training(split)
validate_df <- testing(split)
# Ouptput Matix Y
y <- as.matrix(train_df[,9])
m <- nrow(y)
one <- matrix(1,nrow = m, ncol = 1)
x1 <-  as.matrix(train_df[,1:8])
# Input Matrix X
x <- as.matrix(cbind(one,x1))
J <- 0
# Calculate Cost Function
Jcost <- function(X, Y, b){
  m <- length(Y)
  return((t(X%*%t(b)-Y))%*%(X%*%t(b)-Y)/(2*m))
}
gradFunction <- function(x, y, theta) {
  gradient <- (1/m)* (t(x) %*% ((x %*% t(theta)) - y))
  return(t(gradient))
}
# Gradient Descent Function
grad.descent <- function(x, itr,alpha){
  theta <- matrix(c(0,0,0,0,0,0,0,0,0), nrow=1)
  J <- rep(0, itr)
   for (i in 1:itr) {
```

```
   theta <- theta - alpha  * gradFunction(x, y, theta)
   J[i] <- Jcost(x,y,theta)
   }
  theta
  plot(J, xlab= "No. of Iteration", ylab= "Jcost", type= "b", cex=0.7, col= "red")
  return(theta)
}
par(mfrow = c(2,2))
g1 <- grad.descent(x,500,0.01)
g2<- grad.descent(x,100,0.1)
g3<- grad.descent(x,100,0.3)
g4<- grad.descent(x,100,0.5)
```



From the above plots we can obsereve that as alpha increases the speed of convergence also increases. Alpha value of 0.01 converges after 50 iterations while an alpha of 0.1 converges near 10 iterations.

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 3.6.2
```

```
validate_df$x0 <- 1
validate_df <- validate_df[,c(10,c(1:9))]
v_X <- as.matrix(validate_df[c(1:9)])
v_Y <- as.matrix(validate_df$strength)
Y_test <- scale(v_Y[,])
```

```
predict_Y <- v_X %*% t(g3)
residual <- v_Y - predict_Y
Metrics::mae(Y_test,predict_Y)
```

```
## [1] 0.4916008
```

```
Metrics::mape(Y_test,predict_Y)
```

```
## [1] 1.521198
```

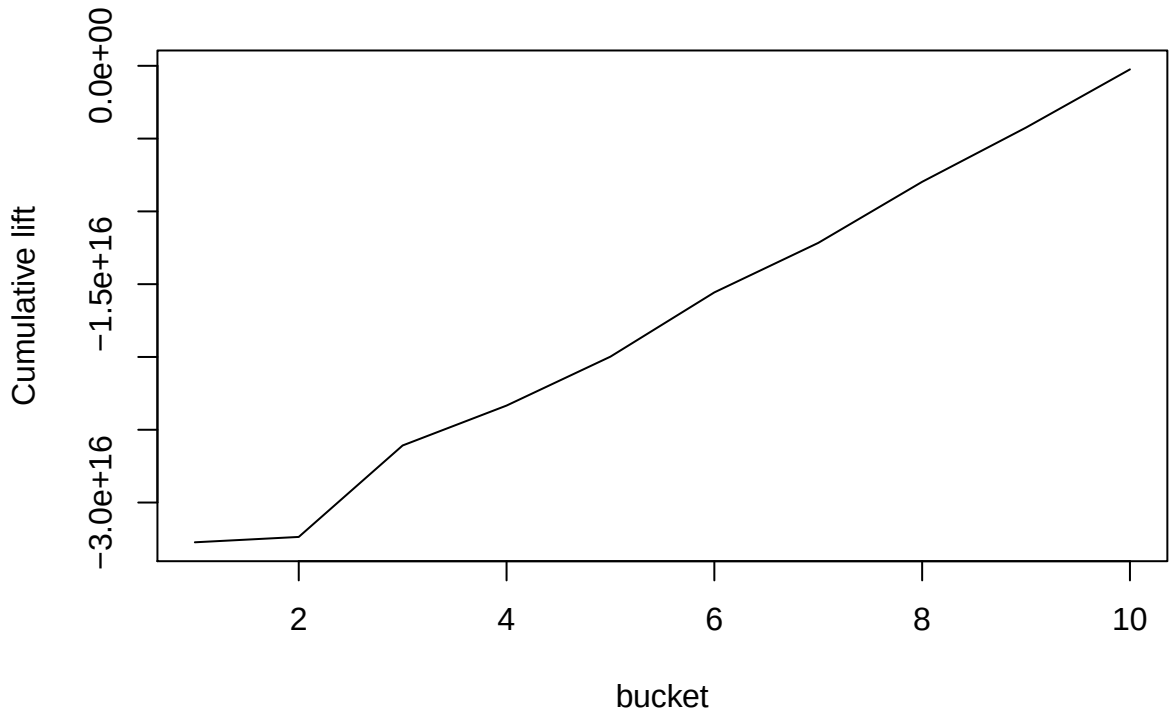
```
rmse(Y_test,predict_Y)
```

```
## [1] 0.6137162
```

```
cor(Y_test,predict_Y)
```

```
##           [,1]
## [1,] 0.7900877
```

```
library(lift)
plotLift(predict_Y,Y_test)
```



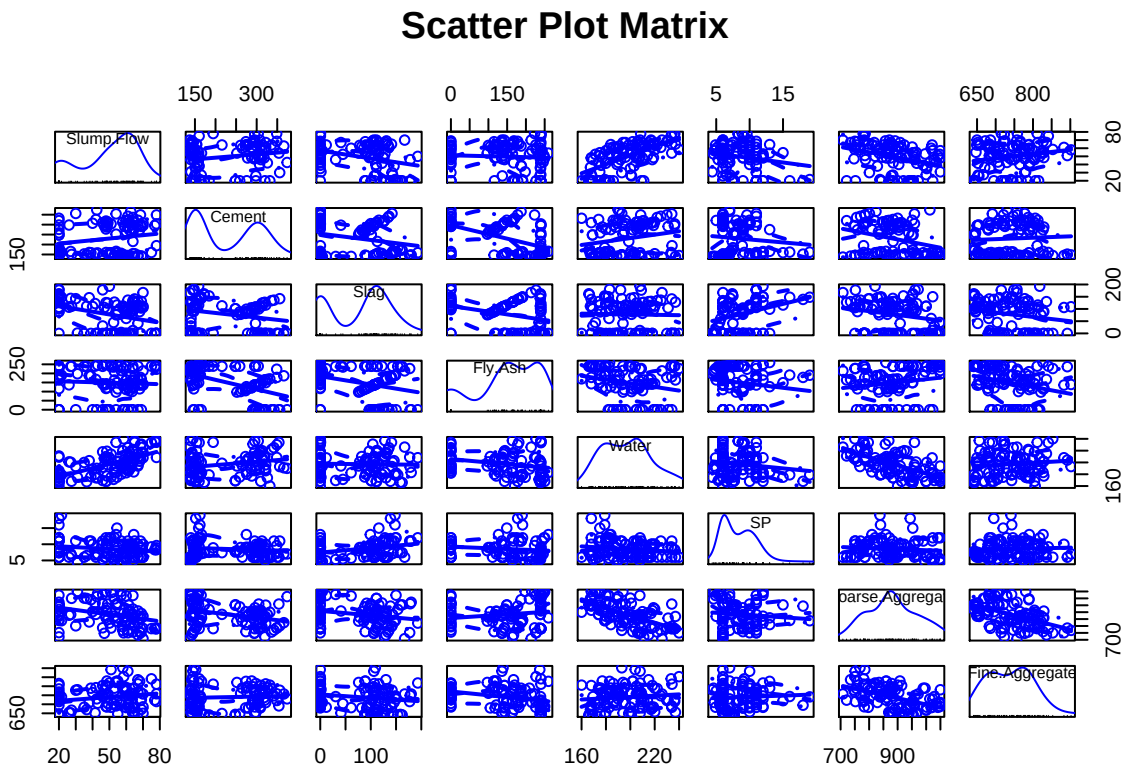
Problem 02

```
library(readxl)
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.2
```

```
## Loading required package: carData
```

```
concreteSlump_df <- read_xlsx("Concrete Slump Test Data.xlsx")
df <- concreteSlump_df[,c(10,2:8)]
#Scatter Plot Matrix for Concrete Slump Test Data
scatterplotMatrix(df,main = "Scatter Plot Matrix")
```




```
# Model 01
fit_df01 <- lm(`Slump Flow` ~ Cement+Slag+`Fly Ash`+Water+SP+`Coarse Aggregate`+`Fine Aggregate`,data =
summary(fit_df01)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate`, data = df)
```

```
##
## 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
## `Fly Ash`             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
## `Coarse Aggregate`    0.07366    0.13510   0.545   0.5869
## `Fine Aggregate`      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
```

From the above summary we can observe that for every unit change in Water there is a 0.73 times increase in Slump Flow. While for unit increase in Slag there is a 0.006 decrease in Slump Flow. 50% of the variance in the dependent variable can be explained by the predictor variables.

```
# Model 02
fit_df2 <- lm(`Slump Flow` ~ Cement+Slag+`Fly Ash`+Water+SP+`Coarse Aggregate`+`Fine Aggregate`+Slag:Wa
summary(fit_df2)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate` + Slag:Water +
##     `Coarse Aggregate`:`Fine Aggregate` + Cement:`Fly Ash` +
##     SP:Slag, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.2227  -5.8980   0.5678   7.6492  22.8381
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -1.172e+03  4.066e+02  -2.881 0.004937 **
## Cement                               1.758e-01  1.135e-01   1.549 0.124777
## Slag                                -3.955e-01  2.379e-01  -1.663 0.099808 .
## `Fly Ash`                            1.316e-01  1.207e-01   1.090 0.278389
## Water                                1.075e+00  3.325e-01   3.233 0.001708 **
## SP                                   2.653e+00  9.645e-01   2.751 0.007175 **
## `Coarse Aggregate`                   7.477e-01  2.708e-01   2.761 0.006964 **
## `Fine Aggregate`                     8.390e-01  3.047e-01   2.754 0.007114 **
## Slag:Water                           3.829e-03  1.099e-03   3.486 0.000757 ***
## `Coarse Aggregate`:`Fine Aggregate` -5.960e-04  3.003e-04  -1.984 0.050233 .
```

```
## Cement:`Fly Ash`                     4.756e-04  2.074e-04   2.293 0.024132 *
## Slag:SP                             -1.321e-02  7.712e-03  -1.714 0.090009 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.74 on 91 degrees of freedom
## Multiple R-squared:  0.6018, Adjusted R-squared:  0.5537
## F-statistic:  12.5 on 11 and 91 DF,  p-value: 4.956e-14
```

From the above summary we can observe that for every unit change in Water there is a 1.075 unit increase in Slump Flow. While for unit increase in Slag there is a 1.45 decrease in Slump Flow. 60% of the variance in the dependent variable can be explained by the predictor variables.Comparing the R-squared value of the first two models we can see that the second model is a better fit.

```
# Model 03
fit_df3 <- lm(`Slump Flow` ~ Cement+Slag+`Fly Ash`+Water+SP+`Coarse Aggregate`+`Fine Aggregate`+Cement:S
`Coarse Aggregate`:`Fine Aggregate`, data = df)
summary(fit_df3)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate` + Cement:Slag +
##     Cement:`Fly Ash` + Cement:Water + Cement:SP + Cement:`Coarse Aggregate` +
##     Cement:`Fine Aggregate` + Slag:`Fly Ash` + Slag:Water + Slag:SP +
##     Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` + `Fly Ash`:Water +
##     `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:SP + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate` + `Coarse Aggregate`:`Fine Aggregate`,
##     data = df)
##
## 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 ***
## `Fly Ash`                          -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
## `Coarse Aggregate`                 -5.943e-01  5.978e-01  -0.994 0.323325
## `Fine Aggregate`                   -9.309e-01  7.902e-01  -1.178 0.242545
## Cement:Slag                        -2.639e-04  5.594e-04  -0.472 0.638511
## Cement:`Fly Ash`                    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:`Coarse Aggregate`           5.554e-04  5.822e-04   0.954 0.343217
## Cement:`Fine Aggregate`             3.448e-04  6.659e-04   0.518 0.606098
## Slag:`Fly Ash`                      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:`Coarse Aggregate`               1.928e-03  5.389e-04   3.577 0.000618 ***
## Slag:`Fine Aggregate`                 1.972e-03  7.217e-04   2.732 0.007860 **
## `Fly Ash`:Water                       5.582e-03  1.770e-03   3.153 0.002331 **
## `Fly Ash`:SP                          4.320e-02  5.692e-02   0.759 0.450241
## `Fly Ash`:`Coarse Aggregate`          1.428e-03  4.753e-04   3.005 0.003624 **
## `Fly Ash`:`Fine Aggregate`            1.433e-03  5.691e-04   2.519 0.013940 *
## Water:SP                              5.024e-02  1.347e-01   0.373 0.710204
## Water:`Coarse Aggregate`              2.135e-03  1.191e-03   1.793 0.077110 .
## Water:`Fine Aggregate`                4.104e-03  1.841e-03   2.229 0.028857 *
## SP:`Coarse Aggregate`                 3.877e-02  5.893e-02   0.658 0.512625
## SP:`Fine Aggregate`                   3.905e-02  6.008e-02   0.650 0.517680
## `Coarse Aggregate`:`Fine Aggregate` -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
```

In the case of the above regression models, we can observe that the predictive power has improved to 0.75 for this interactive model.

Regression Diagnostics

```
# Confidence Interval for Model 1 predictors
confint(fit_df01)
```

```
##                           2.5 %      97.5 %
## (Intercept)        -947.84451365 442.0951684
## Cement               -0.16942710   0.2767133
## Slag                 -0.31614617   0.3047654
## `Fly Ash`            -0.16520290   0.2875048
## Water                 0.03136972   1.4322277
## SP                   -1.01716230   1.6138194
## `Coarse Aggregate`   -0.19454098   0.3418613
## `Fine Aggregate`     -0.18771010   0.3757443
```

The result suggests that you can be 95% confident that the interval [0.03,1.43] contains true change in Slump Flow for a 1% change in Water. Similarly, the true interval for other predictors at a 95% confidence interval is obtained.

```
# Confidence Interval for Model 2 predictors
confint(fit_df2)
```

```
##                                         2.5 %        97.5 %
## (Intercept)                      -1.979212e+03 -3.639414e+02
## Cement                           -4.960092e-02  4.012559e-01
## Slag                             -8.680490e-01  7.698943e-02
## `Fly Ash`                        -1.080998e-01  3.712464e-01
## Water                             4.144060e-01  1.735251e+00
```

```
## SP                                    7.372022e-01  4.568939e+00
## `Coarse Aggregate`                    2.098289e-01  1.285536e+00
## `Fine Aggregate`                      2.337835e-01  1.444156e+00
## Slag:Water                            1.647029e-03  6.011262e-03
## `Coarse Aggregate`:`Fine Aggregate` -1.192581e-03  6.185158e-07
## Cement:`Fly Ash`                      6.366339e-05  8.876293e-04
## Slag:SP                              -2.853341e-02  2.103709e-03
```

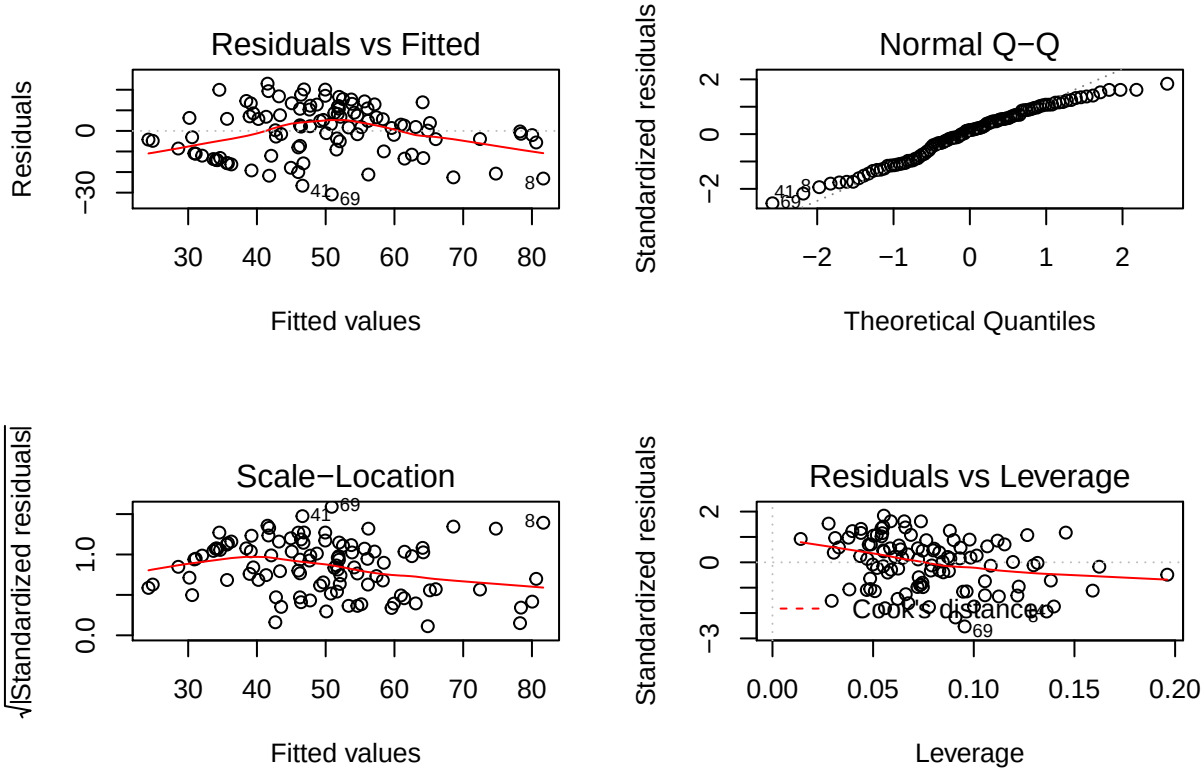
The result suggests that you can be 95% confident that the interval [1,52,1.73] contains true change in Slump Flow for a 1% change in Water. Similarly, the true interval for other predictors at a 95% confidence interval is obtained.

```
# Confidence Interval for Model 3 predictors
confint(fit_df3)
```

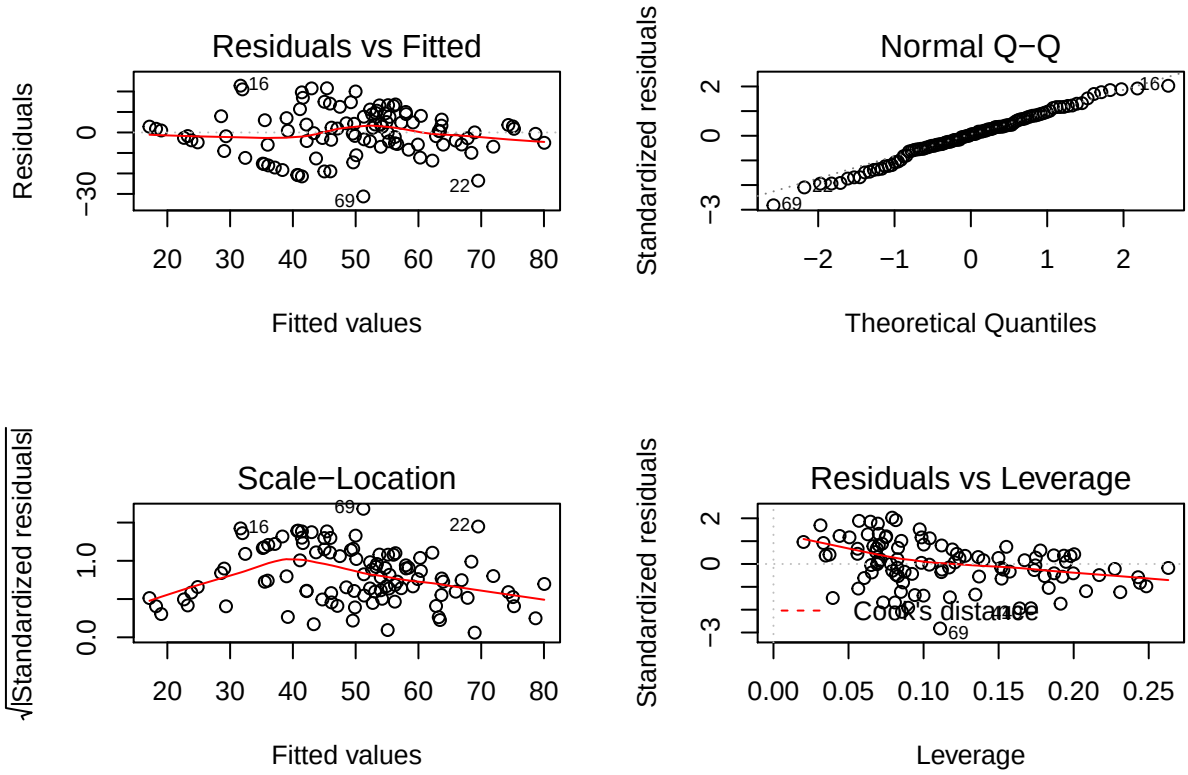
```
##                                          2.5 %        97.5 %
## (Intercept)                         -9.773964e+02  4.110457e+03
## Cement                              -4.401931e+00  1.124973e+00
## Slag                                -8.537963e+00 -2.581070e+00
## `Fly Ash`                           -5.812587e+00 -1.182500e+00
## Water                               -1.170119e+01 -6.295175e-01
## SP                                  -3.857615e+02  2.017023e+02
## `Coarse Aggregate`                  -1.785380e+00  5.967111e-01
## `Fine Aggregate`                    -2.505278e+00  6.435642e-01
## Cement:Slag                         -1.378497e-03  8.507371e-04
## Cement:`Fly Ash`                    -5.247793e-04  1.279678e-03
## Cement:Water                         1.233236e-04  8.821731e-03
## Cement:SP                           -5.275529e-02  1.492668e-01
## Cement:`Coarse Aggregate`           -6.046909e-04  1.715499e-03
## Cement:`Fine Aggregate`             -9.819262e-04  1.671572e-03
## Slag:`Fly Ash`                       8.672375e-06  1.843194e-03
## Slag:Water                           7.393817e-03  1.751875e-02
## Slag:SP                             -1.077855e-01  2.025889e-01
## Slag:`Coarse Aggregate`              8.537064e-04  3.001458e-03
## Slag:`Fine Aggregate`                5.339322e-04  3.410041e-03
## `Fly Ash`:Water                      2.054863e-03  9.108558e-03
## `Fly Ash`:SP                        -7.020604e-02  1.566085e-01
## `Fly Ash`:`Coarse Aggregate`         4.812179e-04  2.375419e-03
## `Fly Ash`:`Fine Aggregate`           2.994021e-04  2.567124e-03
## Water:SP                            -2.181287e-01  3.186088e-01
## Water:`Coarse Aggregate`            -2.380060e-04  4.507667e-03
## Water:`Fine Aggregate`               4.351304e-04  7.771816e-03
## SP:`Coarse Aggregate`               -7.865353e-02  1.562018e-01
## SP:`Fine Aggregate`                 -8.065519e-02  1.587614e-01
## `Coarse Aggregate`:`Fine Aggregate` -9.548951e-04  7.221811e-04
```

The result suggests that you can be 95% confident that the interval [-3.18,-2.31] contains true change in Slump Flow for a 1% change in Water. Similarly, the true interval for other predictors at a 95% confidence interval is obtained.

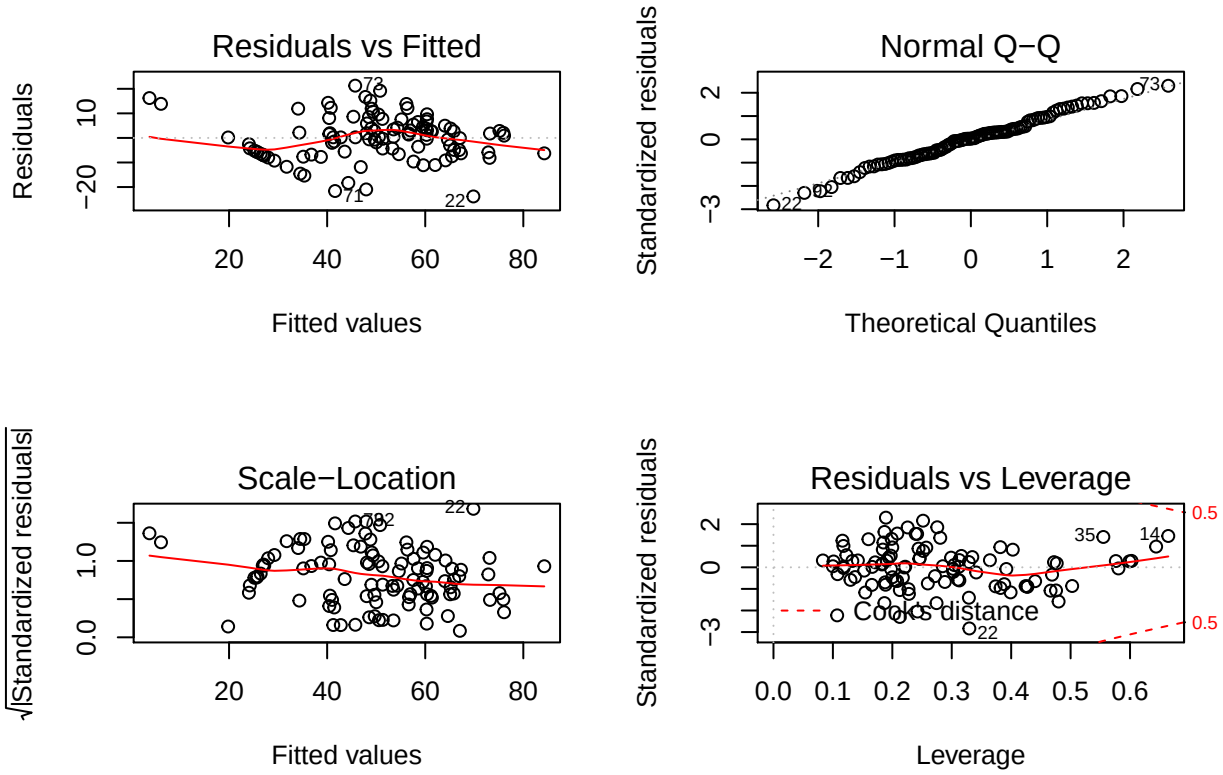
```
# Typical approach Model 1
par(mfrow=c(2,2))
plot(fit_df01)
```




```
# Typical approach Model 2
par(mfrow=c(2,2))
plot(fit_df2)
```



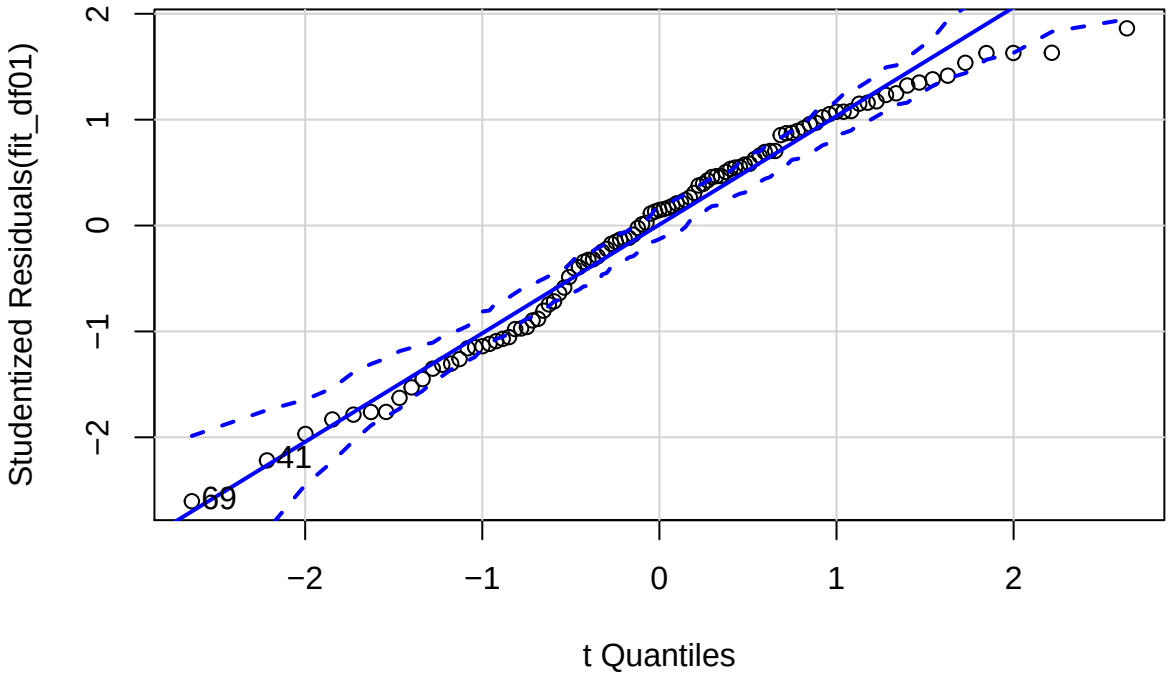

```
# Typical approach
par(mfrow=c(2,2))
plot(fit_df3)
```



###Typical Approach Interpretation :

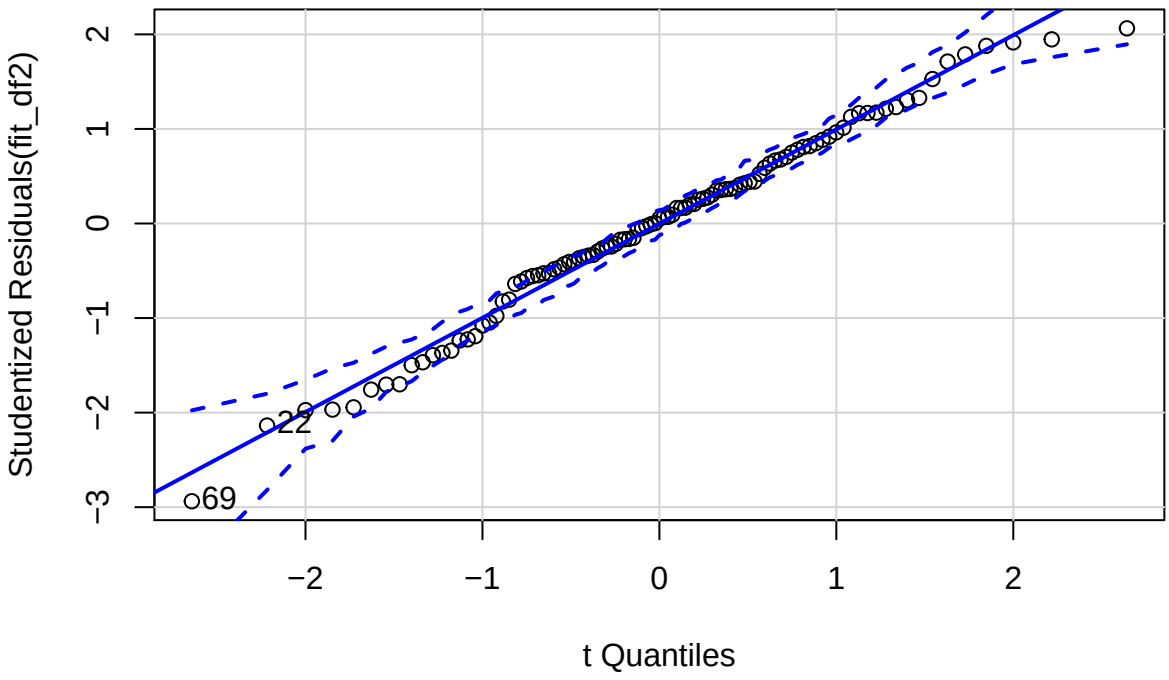
1) Normal Q-Q plot : This plot helps understand if the dependent variable is normally distributed. If the points satisfy the normality condition they must fall on the straight 45-degree line. All the models are generally aligned with the 45-degree line.
2) Residuals vs Fitted : This graph is used to test the linearity assumption, suggesting relation between the dependent and independent variables.
3) Scale-Location : The graph shows a random band around a horizontal line. Hence, we assume that the constant variance assumption is met.
4) Residuals vs Leverage : This graph represents information on individual observations. It helps identify outliers and high leverage values as seen in the plot for model 3.

```
# Enhanced approach Model 1 - qqPlot
par(mfrow=c(1,1))
qqPlot(fit_df01, labels = row.names(df), id.method = "identify",simulate = TRUE)
```



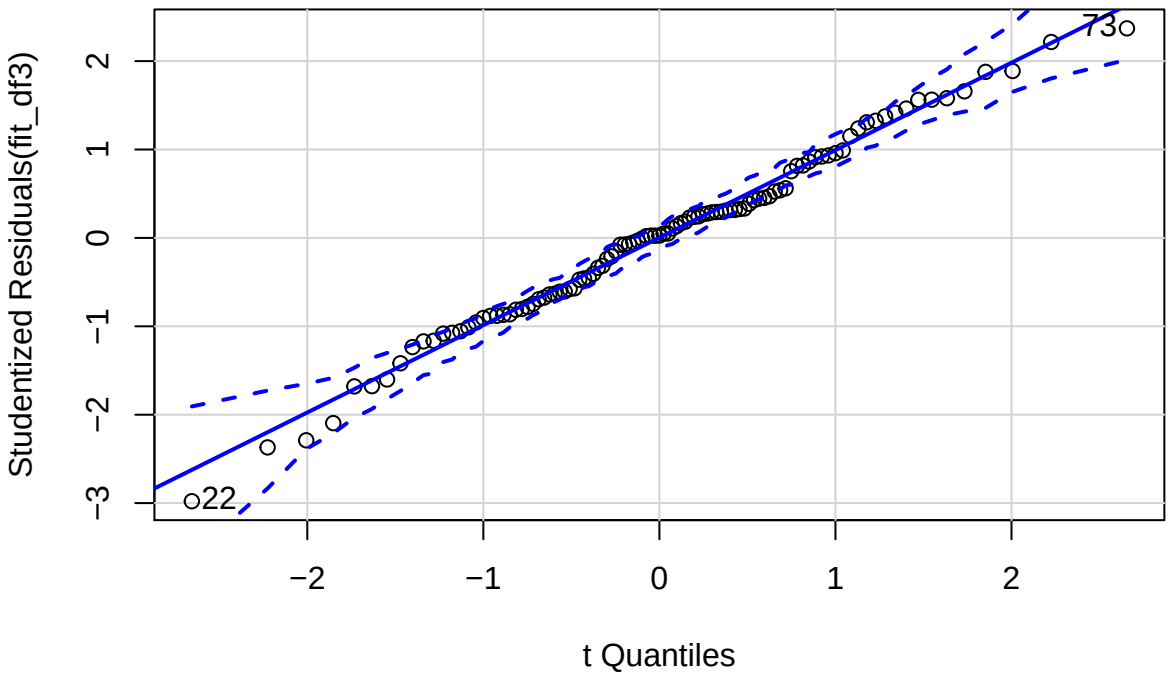

```
## [1] 41 69
```

```
# Enhanced approach Model 2 - qqplot
par(mfrow=c(1,1))
qqPlot(fit_df2, labels = row.names(df), id.method = "identify",simulate = TRUE)
```




```
## [1] 22 69
```

```
# Enhanced approach Model 3 - qqplot
par(mfrow=c(1,1))
qqPlot(fit_df3, labels = row.names(df), id.method = "identify",simulate = TRUE)
```

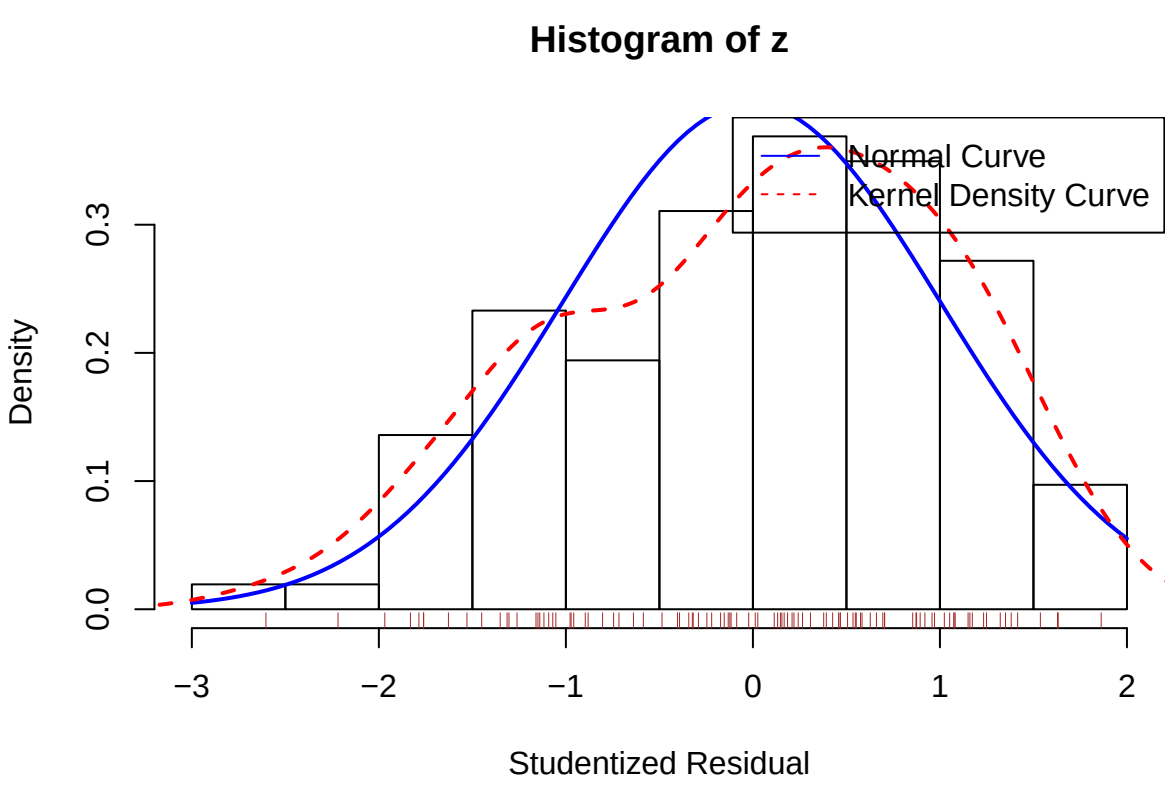



```
## [1] 22 73
```

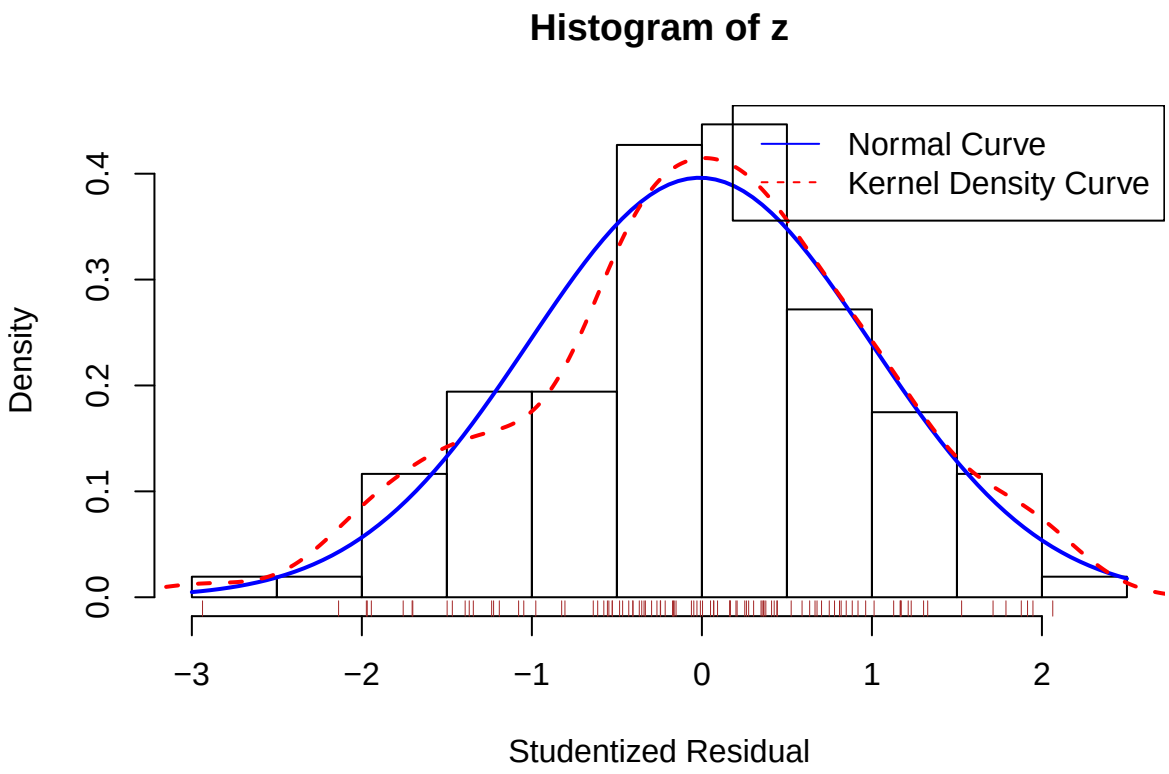
###Interpretation for Enhanced approach - qqPlot

qqPlot() is an accurate method of assesing the normality assumption. All the 3 models fall in line and lie within the confidence interval.

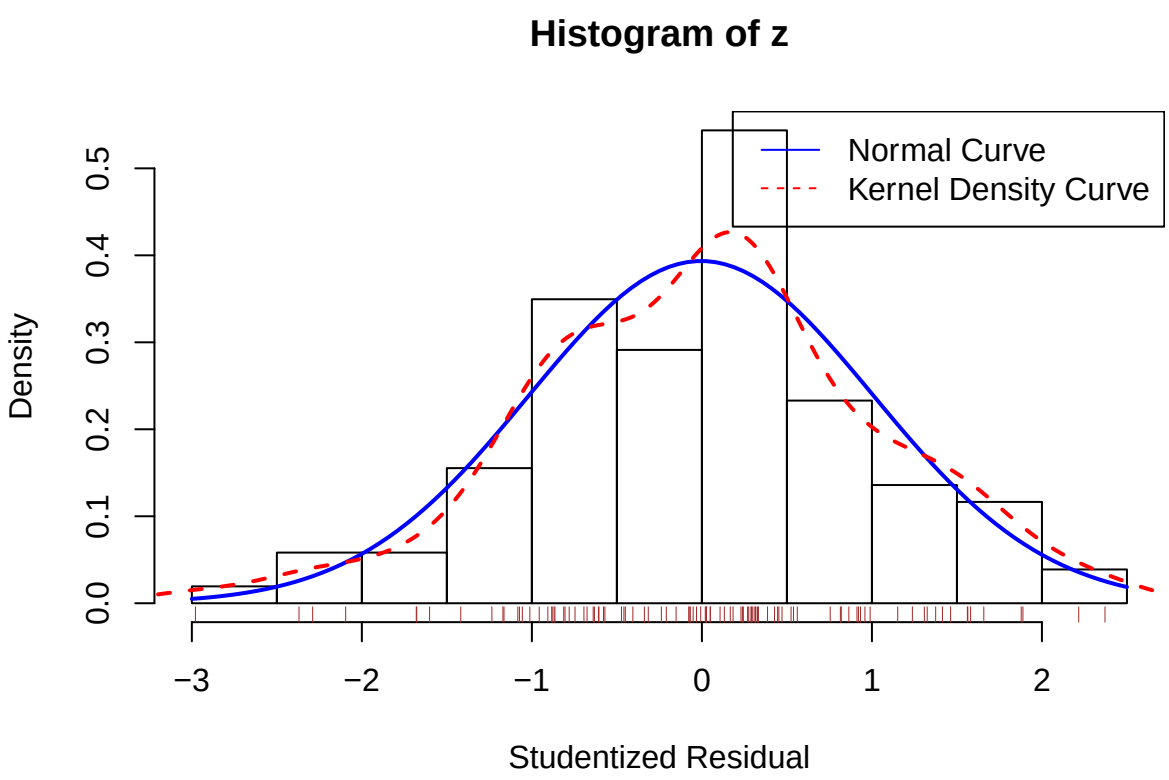
```
residplot <- function(fit, nbreaks=15){
  z <- rstudent(fit)
  hist(z, breaks = nbreaks, freq = FALSE, xlab = "Studentized Residual")
  rug(jitter(z),col="brown")
  curve(dnorm(x,mean=mean(z), sd=sd(z)), add = TRUE, col = "blue",lwd=2)
  lines(density(z)$x,density(z)$y, col="red",lwd=2,lty=2)
  legend("topright",legend = c("Normal Curve","Kernel Density Curve"),lty = 1:2,
         col = c("blue","red"))
}
residplot(fit_df01)
```




```
residplot(fit_df2)
```




```
residplot(fit_df3)
```



###Interpretation for resiplot:

The residplot() function generates a histogram of the studentized residuals along with the normal curve and kernel density curve and rug plot. It is used to visualize the error distribution. All the three models show consistency with the normal distribution.

```
# Enhanced Approach for Model 1 - durbinWatsonTest
durbinWatsonTest(fit_df01)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.01249995      2.009189   0.776
##  Alternative hypothesis: rho != 0
```

```
# Enhanced Approach for Model 2 - durbinWatsonTest
durbinWatsonTest(fit_df2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.03598453        2.0691    0.91
##  Alternative hypothesis: rho != 0
```

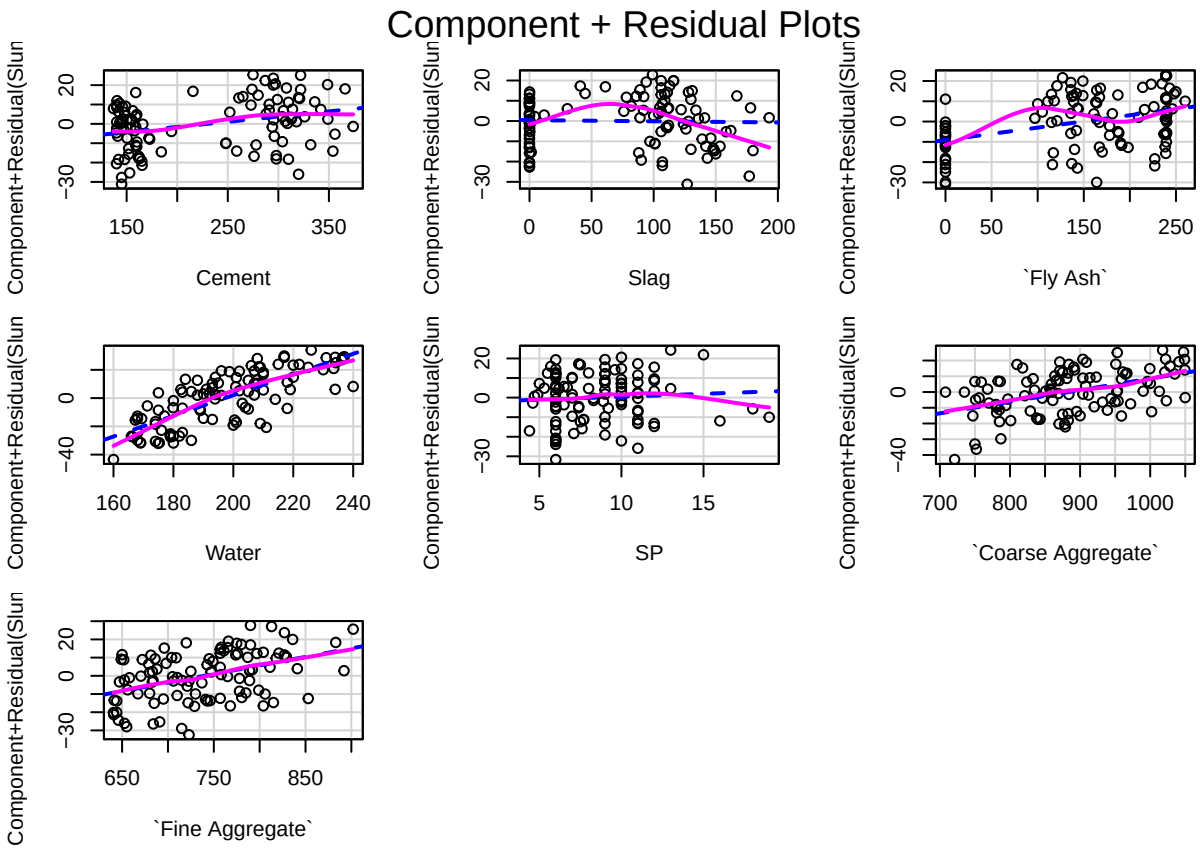
```
# Enhanced Approach for Model 3 - durbinWatsonTest
durbinWatsonTest(fit_df3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.03799301       2.06973   0.598
##  Alternative hypothesis: rho != 0
```

###Interpretation for Durbin Watson Test :

It checks for autocorrelatd errors. p- value > 0.05 suggests lack of autocorrelation and independence of error for all three models.

```
# Enhanced Approach for Model 1 - crPlots
crPlots(fit_df01)
```



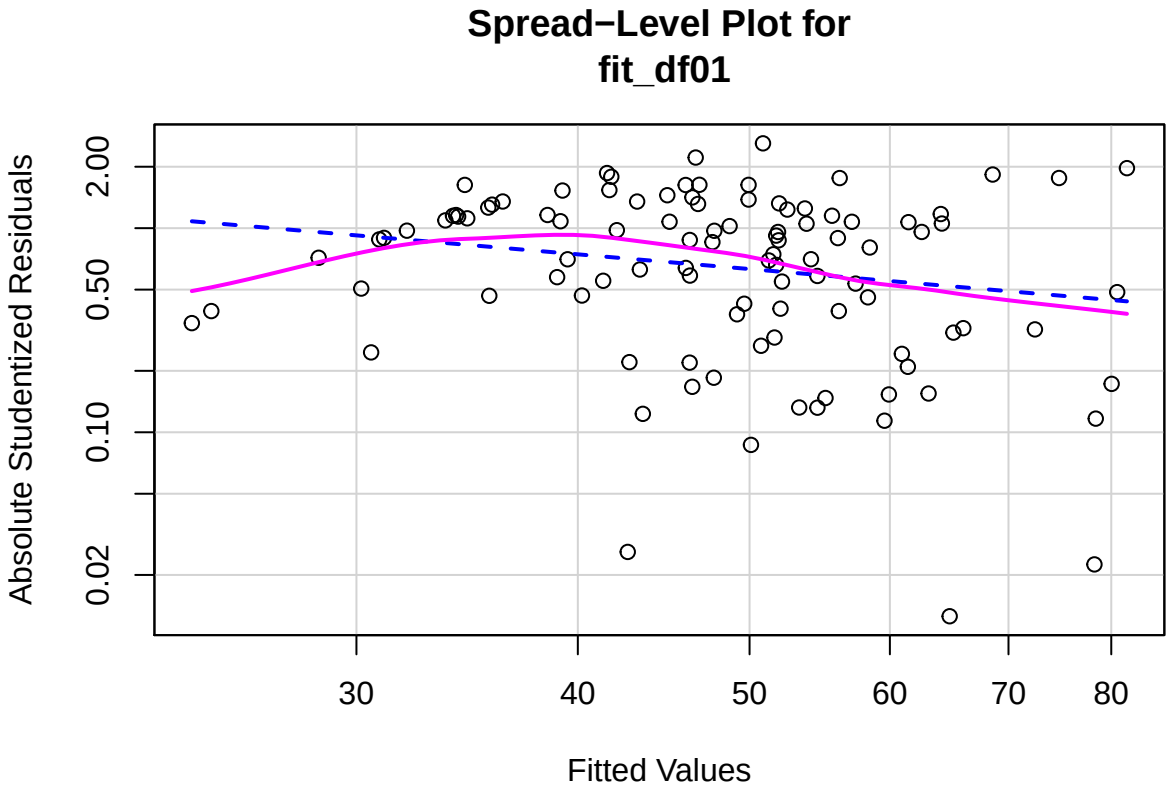
###Interpretation for crPlots() :

crPlots() looks for evidence of non-linearity between dependent variable and independent variable. The above plots confirm the linearity assumption between the variables.

```
# Assessing homoscedasticity
ncvTest(fit_df01)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.2327094, Df = 1, p = 0.62952
```

```
spreadLevelPlot(fit_df01)
```

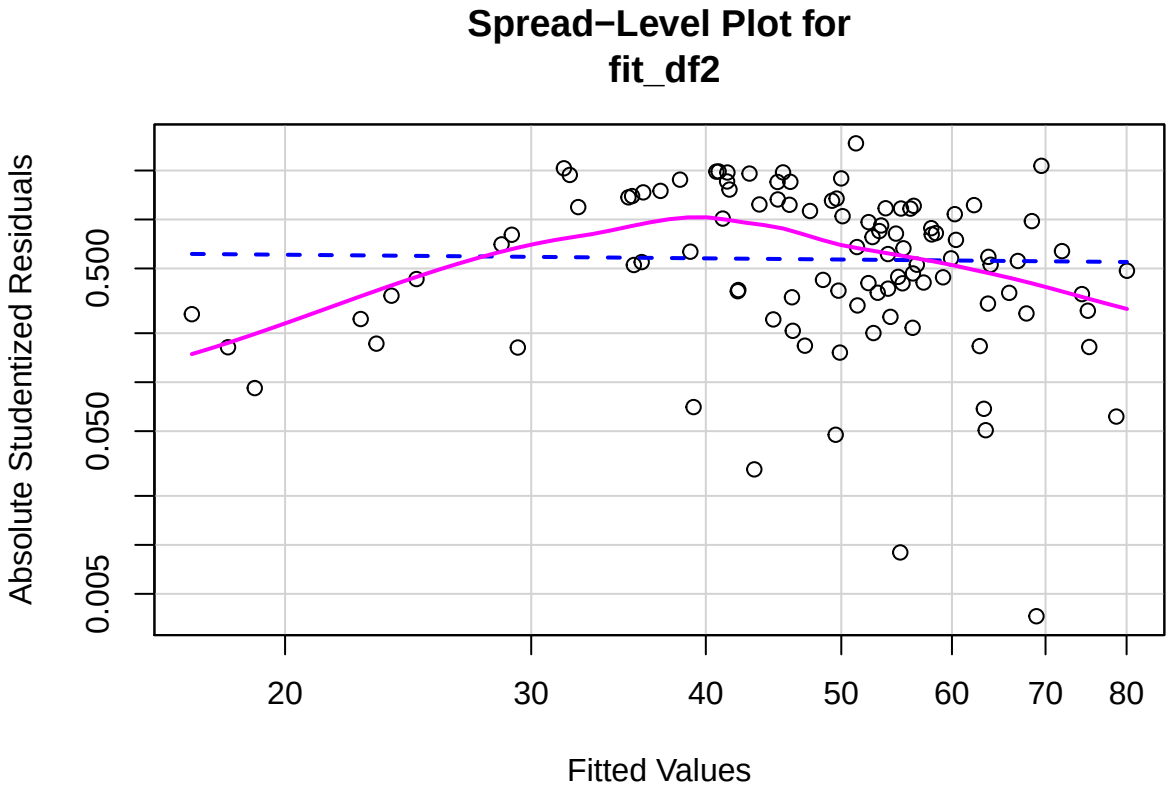



```
##
## Suggested power transformation:  1.743362
```

```
ncvTest(fit_df2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.661547, Df = 1, p = 0.1028
```

```
spreadLevelPlot(fit_df2)
```

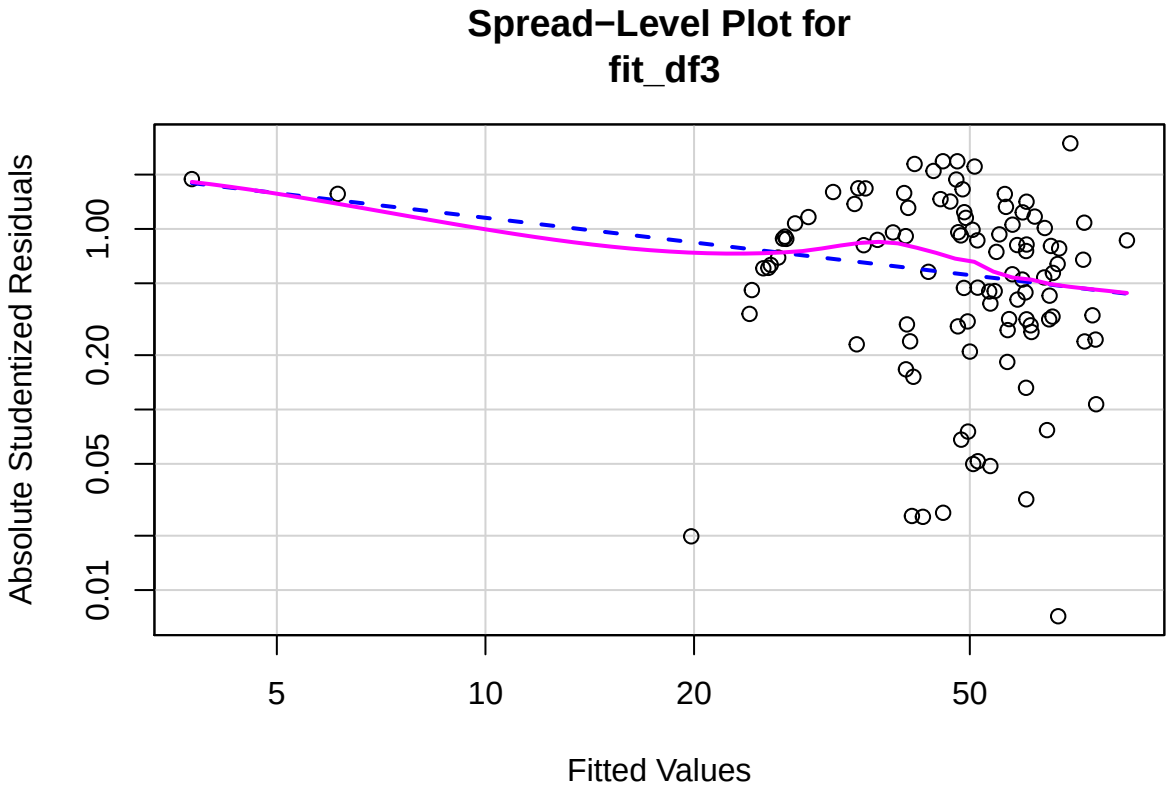



```
##
## Suggested power transformation:  1.07411
```

```
ncvTest(fit_df3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.698219, Df = 1, p = 0.10046
```

```
spreadLevelPlot(fit_df3)
```




```
##
## Suggested power transformation:  1.453976
```

###Assessing Homoscedasticity :

The ncvTest() produces two useful functions for identifying non-constant error variance against the alternative that the error varaince changes with the level of the fitted variables. The p-value is non-significant for all the three models suggesting that the constant variance assumption is met.Also, in the sprealLevelTest the points form a horizontal band around a horizontal line of best fit.

```
# Global validation of linear model assumption
library(gvlma)
gvmodel1 <- gvlma(fit_df01)
summary(gvmodel1)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate`, data = df)
##
## 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
## `Fly Ash`             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
## `Coarse Aggregate`    0.07366    0.13510   0.545   0.5869
## `Fine Aggregate`      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
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit_df01)
##
##                     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.
```

```
gvmodel2 <- gvlma(fit_df2)
summary(gvmodel2)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate` + Slag:Water +
##     `Coarse Aggregate`:`Fine Aggregate` + Cement:`Fly Ash` +
##     SP:Slag, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.2227  -5.8980   0.5678   7.6492  22.8381
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -1.172e+03  4.066e+02  -2.881 0.004937 **
## Cement                             1.758e-01  1.135e-01   1.549 0.124777
## Slag                              -3.955e-01  2.379e-01  -1.663 0.099808 .
## `Fly Ash`                          1.316e-01  1.207e-01   1.090 0.278389
## Water                              1.075e+00  3.325e-01   3.233 0.001708 **
## SP                                 2.653e+00  9.645e-01   2.751 0.007175 **
## `Coarse Aggregate`                 7.477e-01  2.708e-01   2.761 0.006964 **
```

```
## `Fine Aggregate`                      8.390e-01  3.047e-01   2.754 0.007114 **
## Slag:Water                            3.829e-03  1.099e-03   3.486 0.000757 ***
## `Coarse Aggregate`:`Fine Aggregate` -5.960e-04  3.003e-04  -1.984 0.050233 .
## Cement:`Fly Ash`                      4.756e-04  2.074e-04   2.293 0.024132 *
## Slag:SP                              -1.321e-02  7.712e-03  -1.714 0.090009 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.74 on 91 degrees of freedom
## Multiple R-squared:  0.6018, Adjusted R-squared:  0.5537
## F-statistic:  12.5 on 11 and 91 DF,  p-value: 4.956e-14
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit_df2)
##
##                      Value p-value                Decision
## Global Stat        3.33758  0.5030 Assumptions acceptable.
## Skewness           0.87753  0.3489 Assumptions acceptable.
## Kurtosis           0.08371  0.7723 Assumptions acceptable.
## Link Function      2.24984  0.1336 Assumptions acceptable.
## Heteroscedasticity 0.12651  0.7221 Assumptions acceptable.
```

```
gvmodel3 <- gvlma(fit_df3)
summary(gvmodel3)
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate` + Cement:Slag +
##     Cement:`Fly Ash` + Cement:Water + Cement:SP + Cement:`Coarse Aggregate` +
##     Cement:`Fine Aggregate` + Slag:`Fly Ash` + Slag:Water + Slag:SP +
##     Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` + `Fly Ash`:Water +
##     `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:SP + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate` + `Coarse Aggregate`:`Fine Aggregate`,
##     data = df)
##
## 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 ***
## `Fly Ash`                        -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
```

```
## `Coarse Aggregate`                  -5.943e-01  5.978e-01  -0.994 0.323325
## `Fine Aggregate`                    -9.309e-01  7.902e-01  -1.178 0.242545
## Cement:Slag                         -2.639e-04  5.594e-04  -0.472 0.638511
## Cement:`Fly Ash`                     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:`Coarse Aggregate`            5.554e-04  5.822e-04   0.954 0.343217
## Cement:`Fine Aggregate`              3.448e-04  6.659e-04   0.518 0.606098
## Slag:`Fly Ash`                       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:`Coarse Aggregate`              1.928e-03  5.389e-04   3.577 0.000618 ***
## Slag:`Fine Aggregate`                1.972e-03  7.217e-04   2.732 0.007860 **
## `Fly Ash`:Water                      5.582e-03  1.770e-03   3.153 0.002331 **
## `Fly Ash`:SP                         4.320e-02  5.692e-02   0.759 0.450241
## `Fly Ash`:`Coarse Aggregate`         1.428e-03  4.753e-04   3.005 0.003624 **
## `Fly Ash`:`Fine Aggregate`           1.433e-03  5.691e-04   2.519 0.013940 *
## Water:SP                             5.024e-02  1.347e-01   0.373 0.710204
## Water:`Coarse Aggregate`             2.135e-03  1.191e-03   1.793 0.077110 .
## Water:`Fine Aggregate`               4.104e-03  1.841e-03   2.229 0.028857 *
## SP:`Coarse Aggregate`                3.877e-02  5.893e-02   0.658 0.512625
## SP:`Fine Aggregate`                  3.905e-02  6.008e-02   0.650 0.517680
## `Coarse Aggregate`:`Fine Aggregate` -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
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit_df3)
##
##                      Value p-value                Decision
## Global Stat        0.60967  0.9620 Assumptions acceptable.
## Skewness           0.25020  0.6169 Assumptions acceptable.
## Kurtosis           0.14178  0.7065 Assumptions acceptable.
## Link Function      0.18696  0.6655 Assumptions acceptable.
## Heteroscedasticity 0.03073  0.8608 Assumptions acceptable.
```

###Interpretation for global validation of linear model :

The gvlma() performs a global validation of the linear model. On observing the decision column for all the 3 model we can conclude that model 2 and 3 satisfy all the statistical assumptions.

Identify Unusual Observations

```
# Testing for Outliers - Model 1
outlierTest(fit_df01)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 69 -2.603738           0.010717           NA
```

```
# Testing for Outliers - Model 2
outlierTest(fit_df2)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 69 -2.937312          0.0042038      0.43299
```

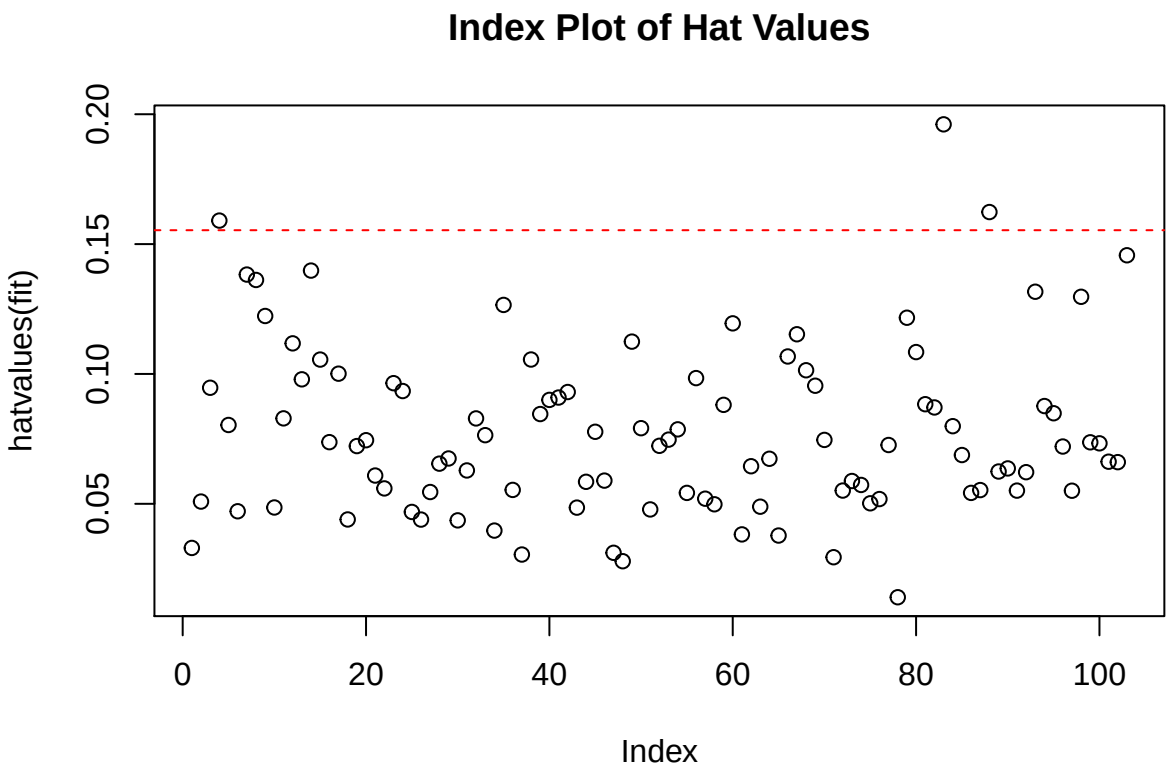
```
# Testing for Outliers - Model 3
outlierTest(fit_df3)
```

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

###Interpretation for Outlier Test :

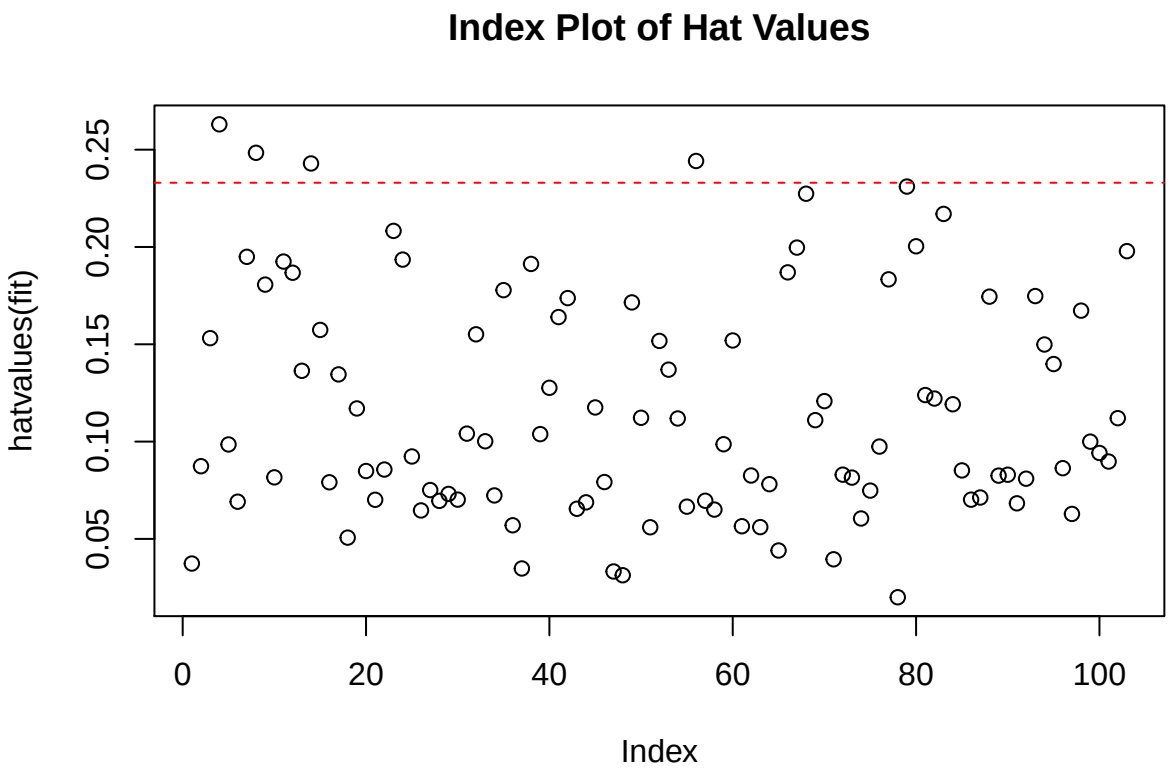
Outliers are the observations that are not predicted well by the model.On performing the outlierTest we observe that there are no significant outliers in the 3 model. However, 69 has the largest studentized residual and can be deleted.

```
# High Leverage Points
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 statistics for model 1
hat.plot(fit_df01)
```



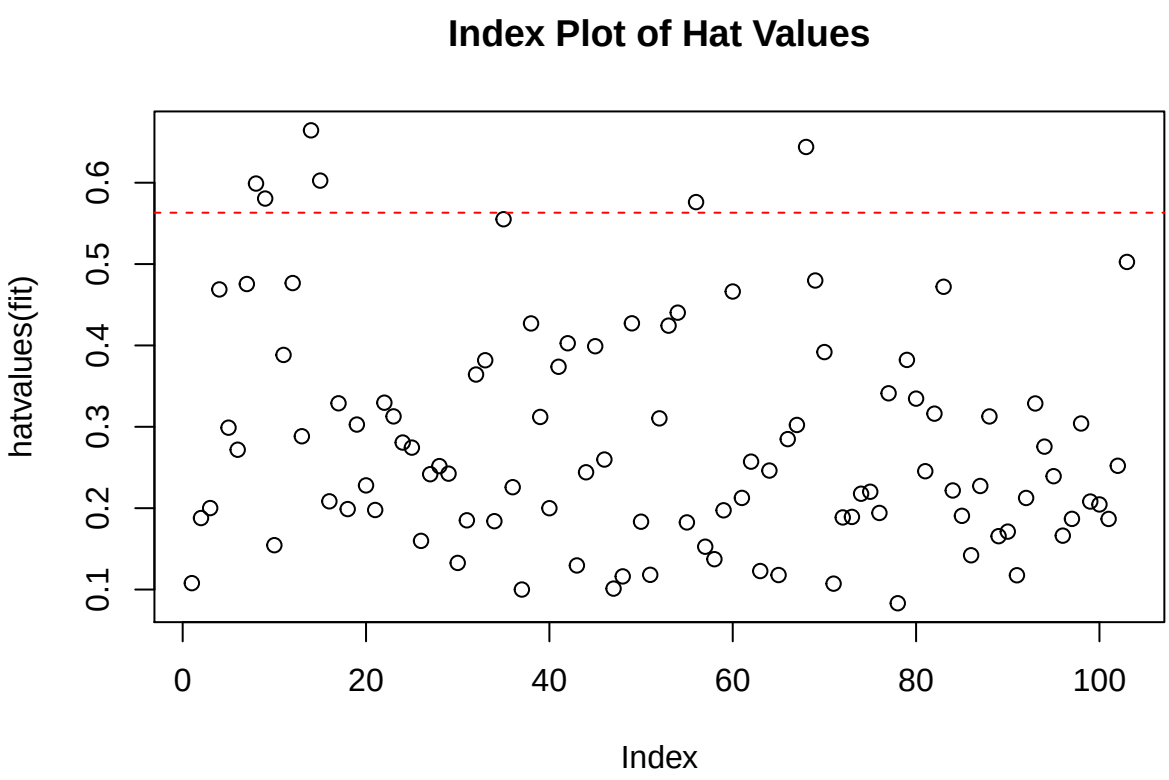

```
## integer(0)
```

```
hat.plot(fit_df2)
```




```
## integer(0)
```

```
hat.plot(fit_df3)
```

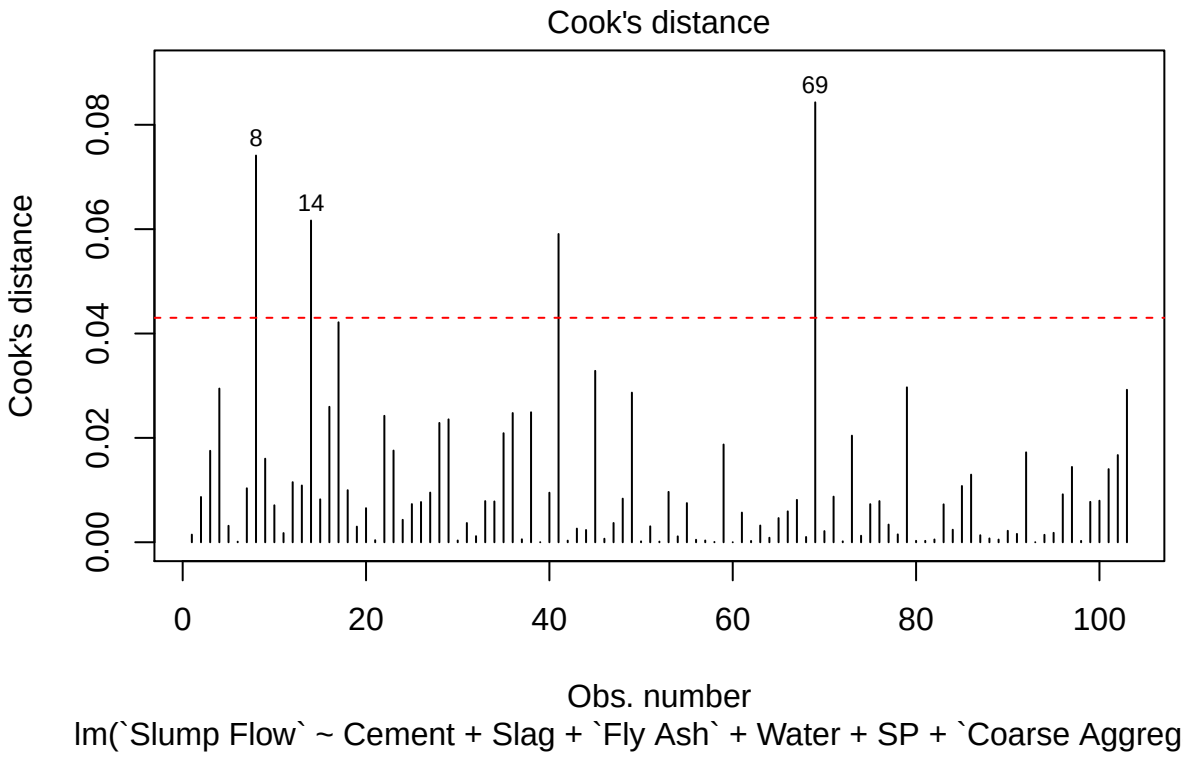



```
## integer(0)
```

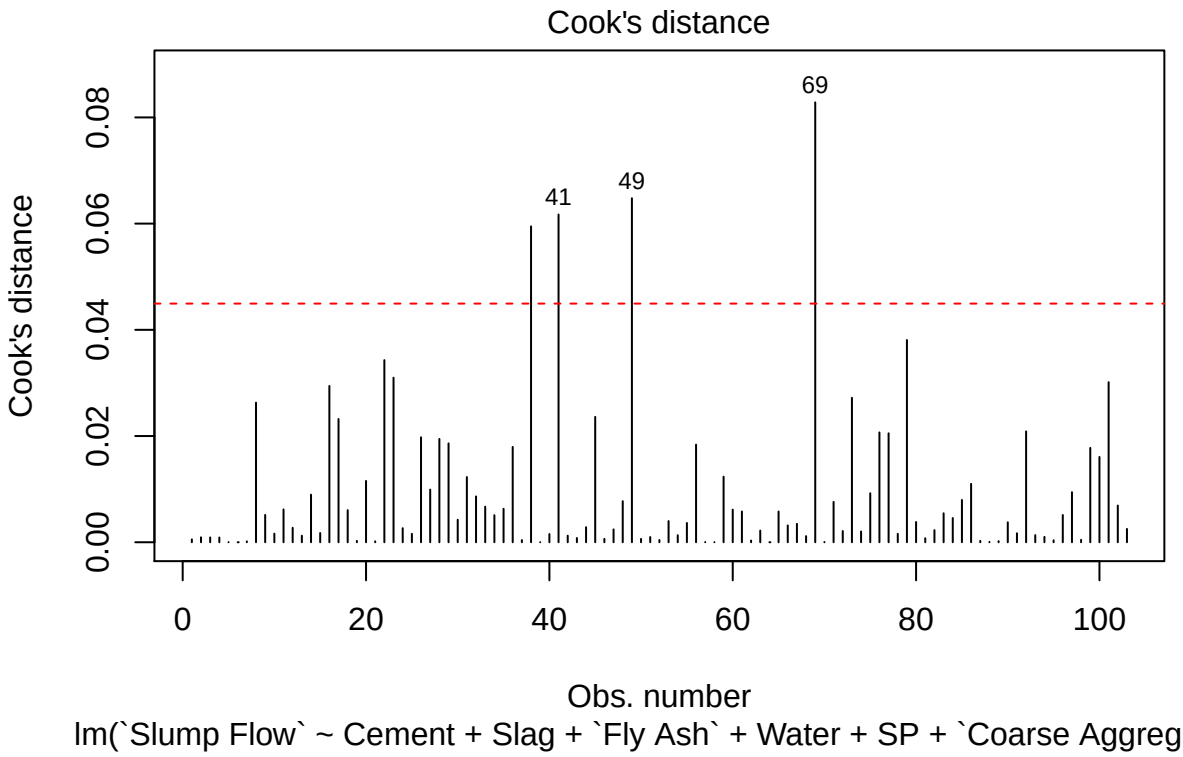
Interpreting hat.plot :

The above plots helps locate if there are any high leverage points. From the above plots we can conclude that there are a few but not very evident.

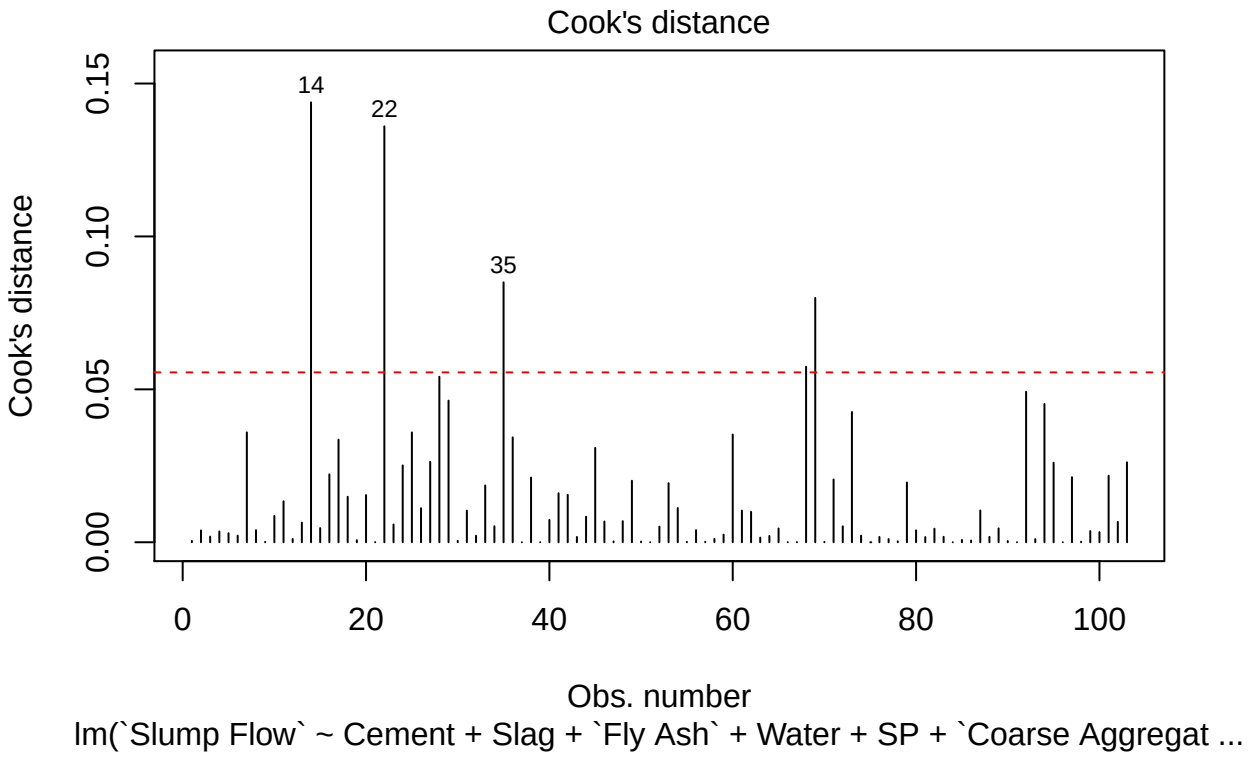
```
# Influential observations
cutoff <- 4/(nrow(df)-length(fit_df01$coefficients)-2)
plot(fit_df01, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```




```r
# Influential observations
cutoff <- 4/(nrow(df)-length(fit_df2$coefficients)-2)
plot(fit_df2, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```



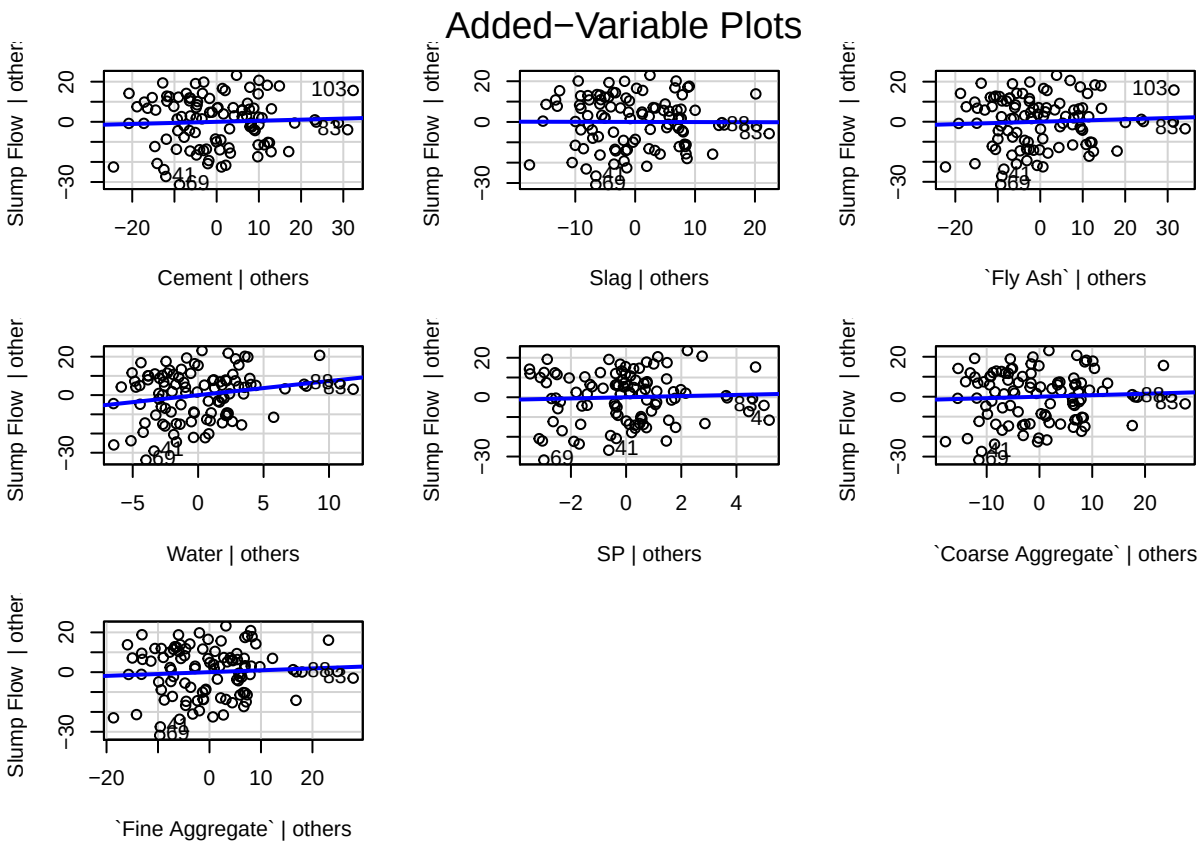

```r
# Influential observations
cutoff <- 4/(nrow(df)-length(fit_df3$coefficients)-2)
plot(fit_df3, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```



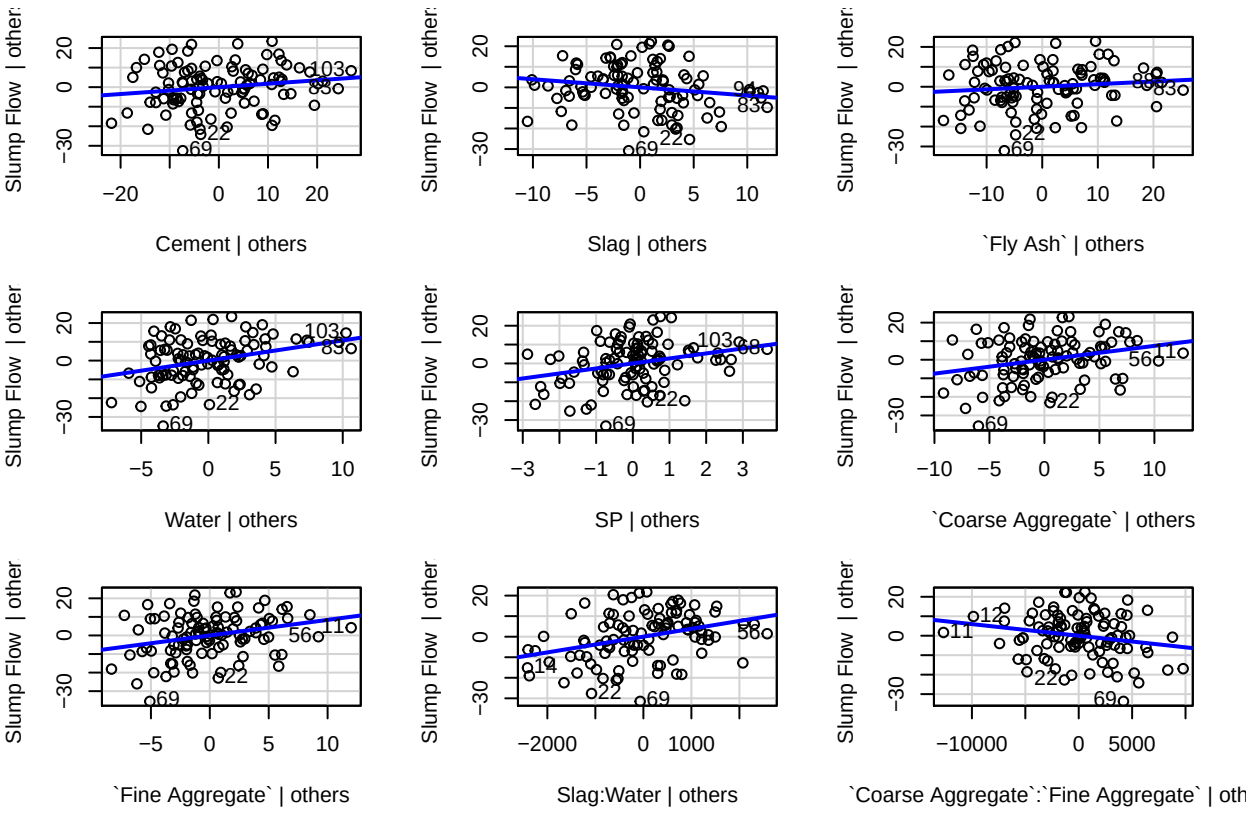
###Interpretation for Influential Observation :

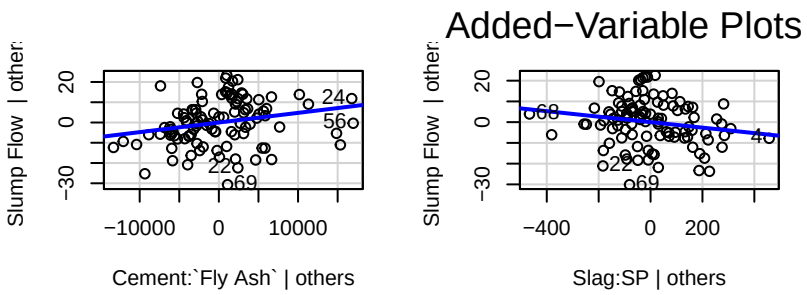
Influential observations are observations that have a disproportionate impact on the model parameters. For the first model 8,14 and 69 are the influential observations. In the second model 41, 49 and 69 are the observational influencers and in the third model 14, 22 and 35.

```
# Added Variable Plot Model 1
avPlots(fit_df01, ask=FALSE)
```

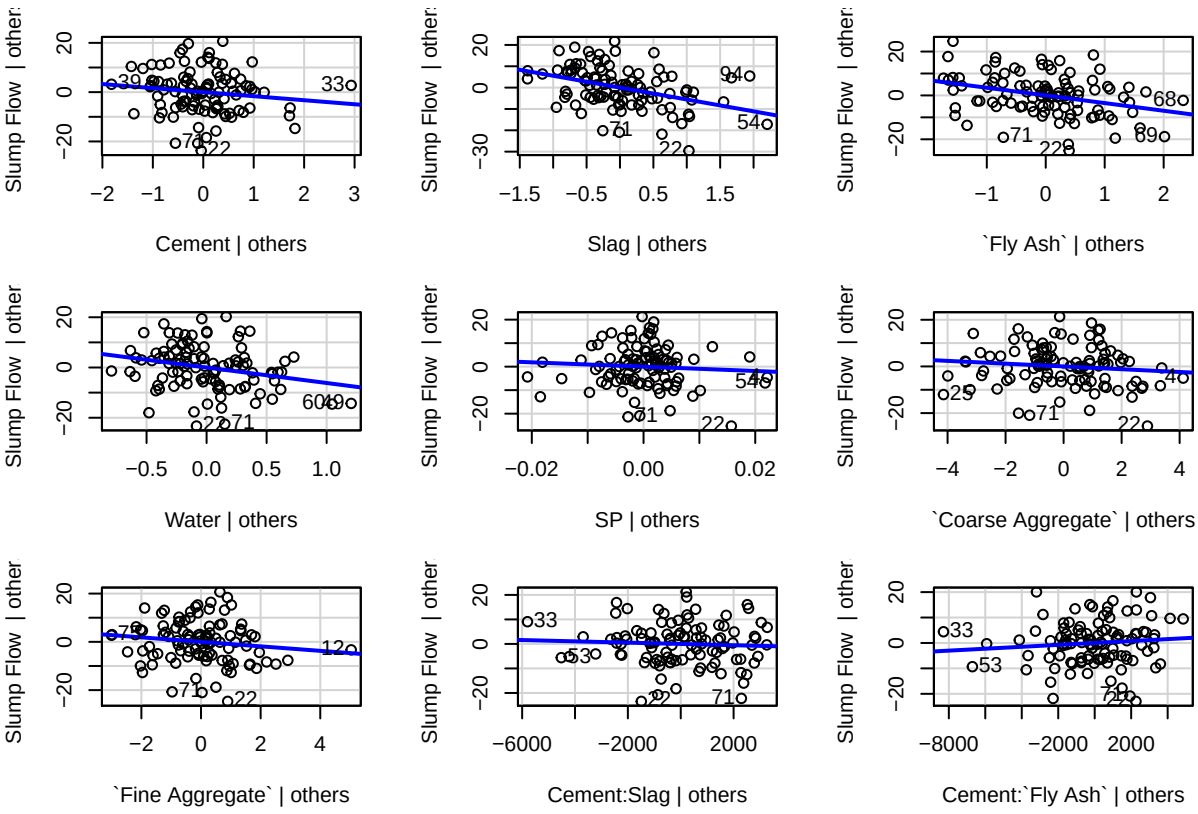


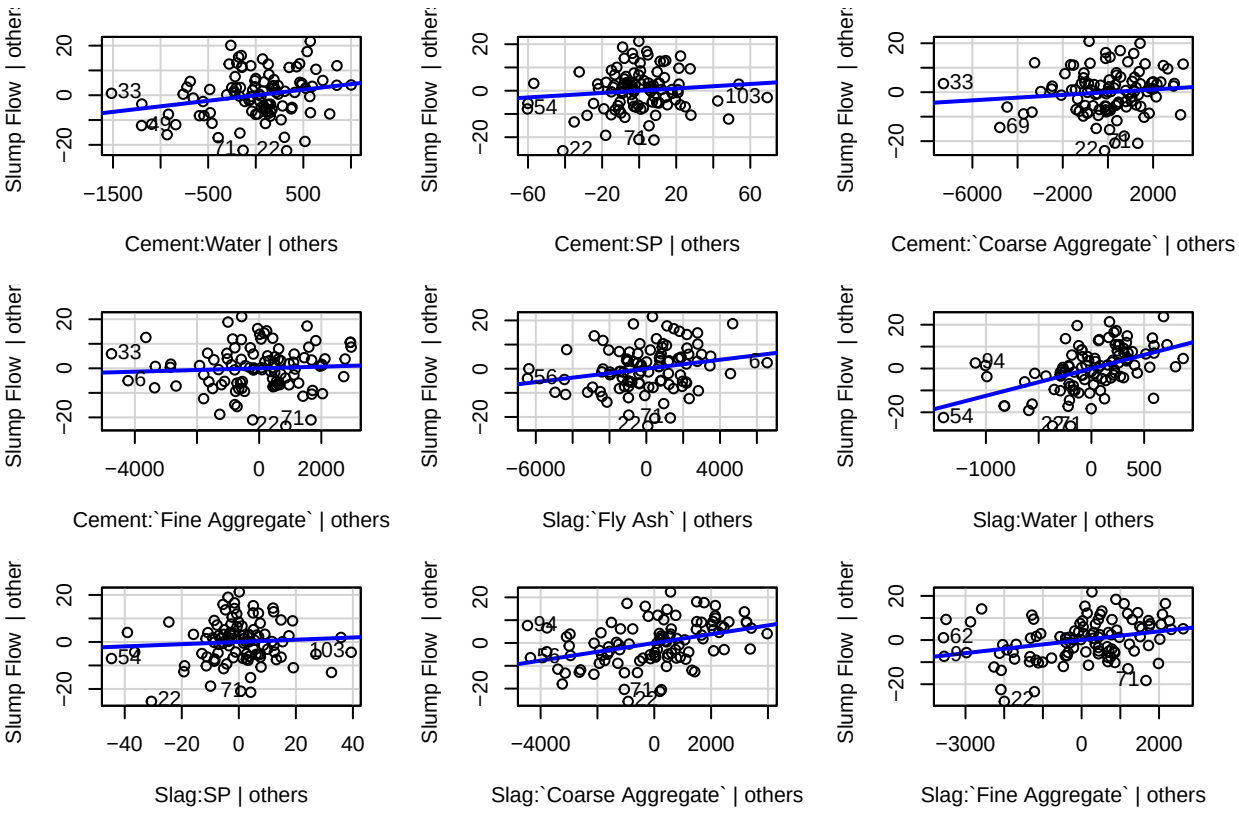

```
# Added Variable Plot Model 2
avPlots(fit_df2, ask=FALSE)
```

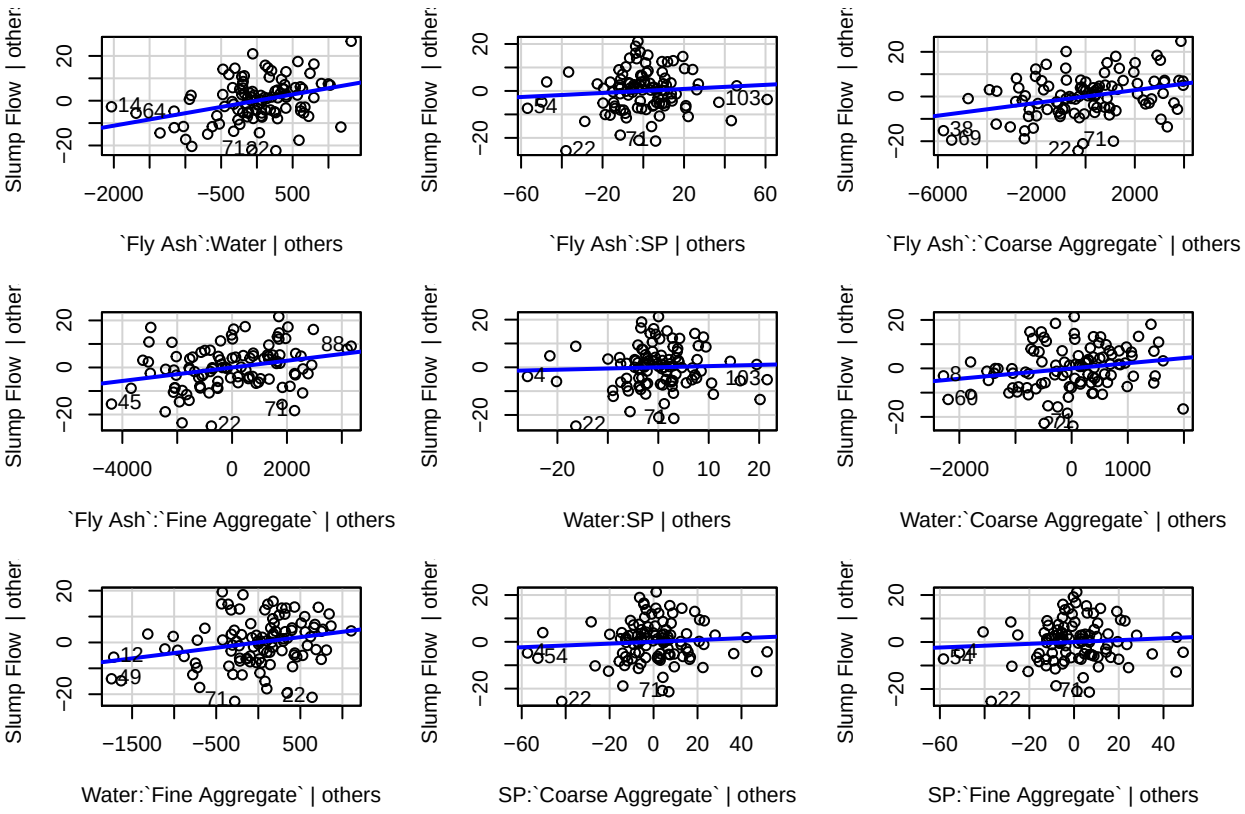

Slump Flow | others
Cement | others
Slag | others
`Fly Ash` | others
Water | others
SP | others
`Coarse Aggregate` | others
`Fine Aggregate` | others
Slag:Water | others
`Coarse Aggregate`:`Fine Aggregate` | oth

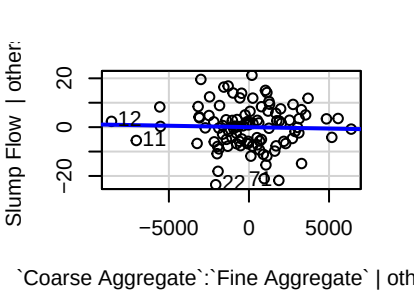



```
# Added Variable Plot Model 3
avPlots(fit_df3, ask=FALSE)
```


Slump Flow | others
Cement | others
Slag | others
`Fly Ash` | others
Water | others
SP | others
`Coarse Aggregate` | others
`Fine Aggregate` | others
Cement:Slag | others
Cement:`Fly Ash` | others


Slump Flow | others
Cement:Water | others
Cement:SP | others
Cement:`Coarse Aggregate` | others
Cement:`Fine Aggregate` | others
Slag:`Fly Ash` | others
Slag:Water | others
Slag:SP | others
Slag:`Coarse Aggregate` | others
Slag:`Fine Aggregate` | others

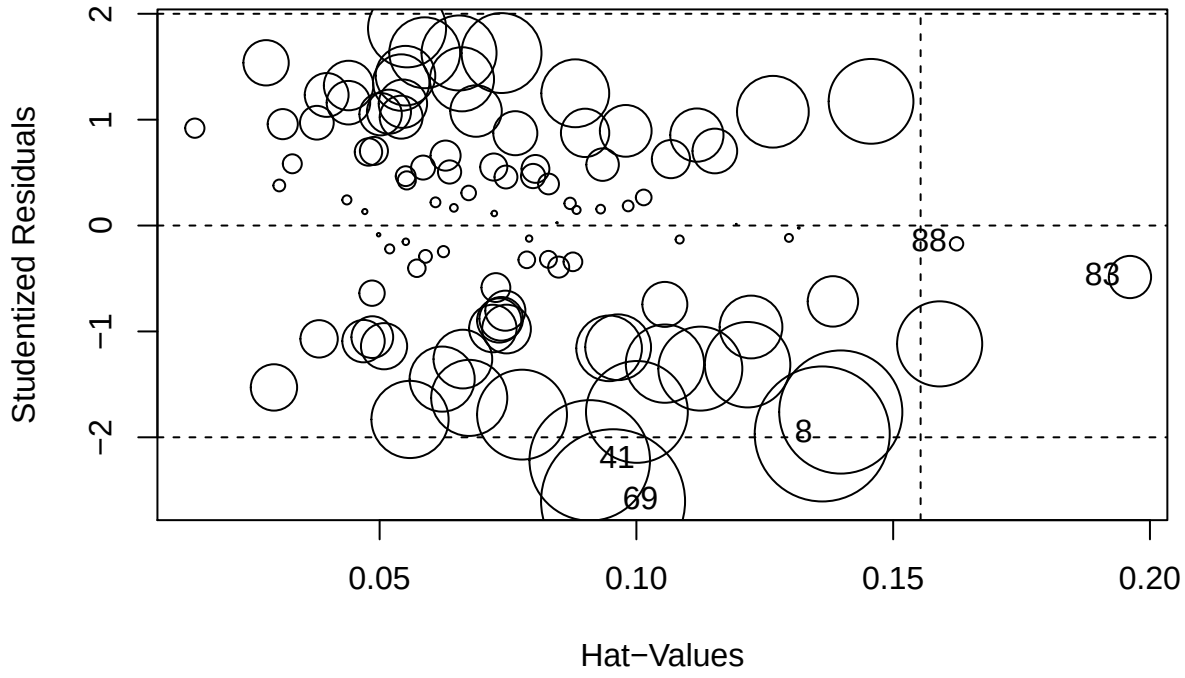

Slump Flow | others
`Fly Ash`:Water | others
`Fly Ash`:SP | others
`Fly Ash`:`Coarse Aggregate` | others
`Fly Ash`:`Fine Aggregate` | others
Water:SP | others
Water:`Coarse Aggregate` | others
Water:`Fine Aggregate` | others
SP:`Coarse Aggregate` | others
SP:`Fine Aggregate` | others



###Interpretation of avPlots :

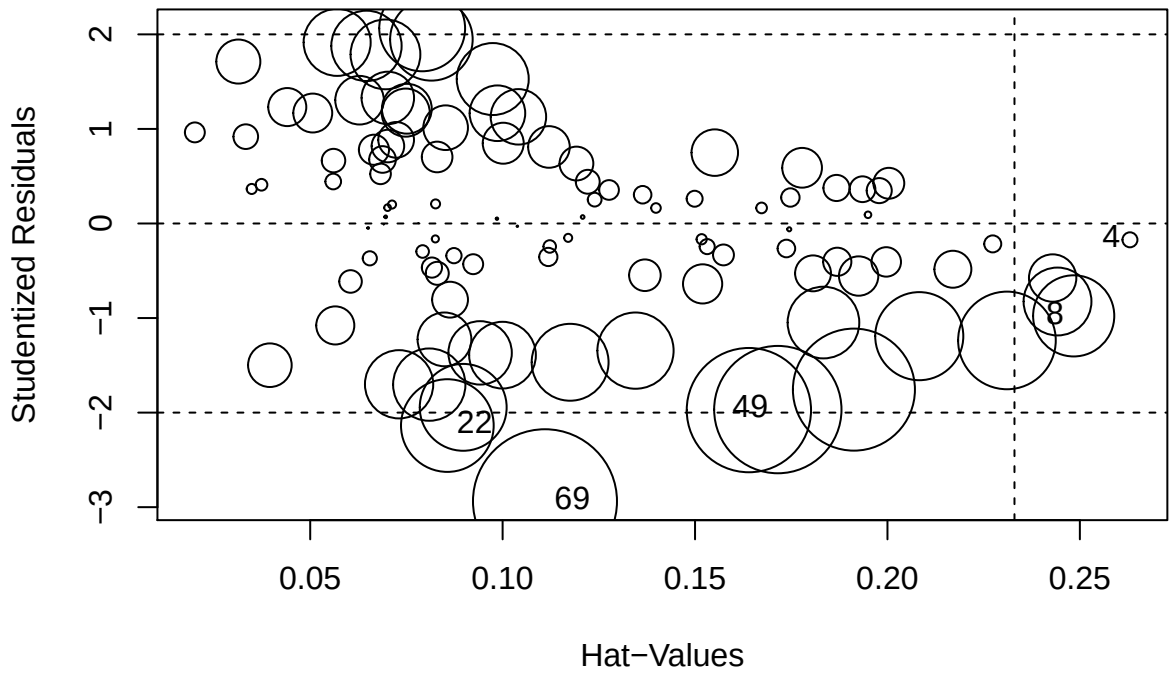
avPlots not only help identify influential observations but also the affect of these observations on the model.

```
# Influence plot for model 1
influencePlot(fit_df01)
```



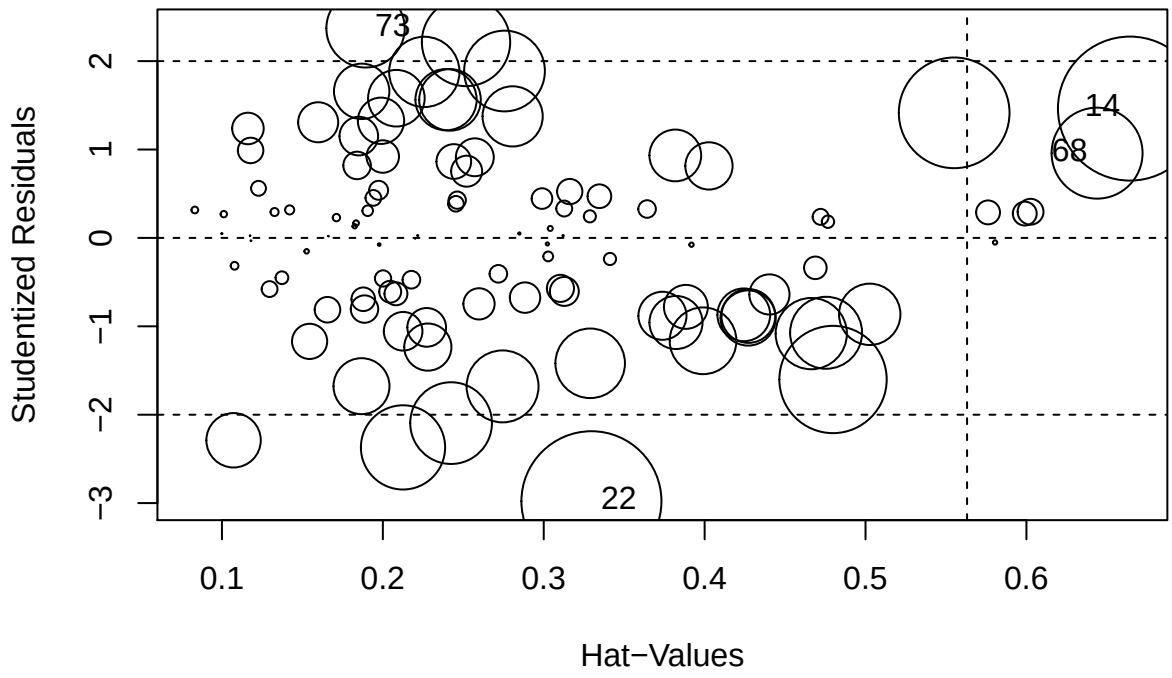

```
##       StudRes        Hat        CookD
## 8  -1.9682860 0.13618195 0.0741036039
## 41 -2.2181634 0.09091319 0.0590686474
## 69 -2.6037375 0.09545758 0.0843019258
## 83 -0.4860173 0.19610512 0.0072612082
## 88 -0.1726772 0.16233785 0.0007297749
```

```
# Influence plot for model 2
influencePlot(fit_df2)
```




```
##       StudRes        Hat        CookD
## 4  -0.1725723 0.26301285 0.0008952273
## 8  -0.9769447 0.24839409 0.0262982815
## 22 -2.1368979 0.08563682 0.0342951684
## 49 -1.9681624 0.17151391 0.0647817009
## 69 -2.9373117 0.11100996 0.0828372974
```

```
# Influence plot for model 3
influencePlot(fit_df3)
```




```
##       StudRes       Hat      CookD
## 14  1.4628989 0.6643472 0.14384526
## 22 -2.9789552 0.3296439 0.13600482
## 68  0.9589935 0.6439176 0.05740963
## 73  2.3706558 0.1892469 0.04257727
```

###Interpreatation Influence Plots :

Model 1 - 1) Outliers : 41 and 69 2) High leverage values : 88 and 83 3) Influentail Obseravtaion : 69

Model 2 - 1) Outliers : 69 2) High leverage values : 8 and 4 3) Influentail Obseravtaion : 69

Model 3 - 1) Outliers : 22 and 73 2) High leverage values : 14 and 68 3) Influentail Obseravtaion : 14

##Corrective Measures

```
#Box cox transformation to Normality
summary(powerTransform(df$`Slump Flow`))
```

```
## bcPower Transformation to Normality
##                 Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## df$`Slump Flow`    1.4678           1       0.9342       2.0015
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                             LRT df       pval
## LR test, lambda = (0) 31.06187  1 2.4993e-08
```

```
##
## Likelihood ratio test that no transformation is needed
##                             LRT df     pval
## LR test, lambda = (1) 3.036391  1 0.081417
```

###Interpretation for power transform :

The results suggest that by replacing Slump Flow with Slump Flowˆ1.4678 normalizes the variable, but there is no strong evidence for the same.

```
# Adding and Deleting Variables
sqrt(vif(fit_df01))>2
```

```
##             Cement               Slag          `Fly Ash`              Water
##               TRUE               TRUE               TRUE               TRUE
##                 SP `Coarse Aggregate`   `Fine Aggregate`
##              FALSE               TRUE               TRUE
```

```
sqrt(vif(fit_df2))>2
```

```
##                              Cement                                Slag
##                                TRUE                                TRUE
##                           `Fly Ash`                               Water
##                                TRUE                                TRUE
##                                  SP                    `Coarse Aggregate`
##                                TRUE                                TRUE
##                     `Fine Aggregate`                          Slag:Water
##                                TRUE                                TRUE
## `Coarse Aggregate`:`Fine Aggregate`                    Cement:`Fly Ash`
##                                TRUE                                TRUE
##                               Slag:SP
##                                TRUE
```

```
sqrt(vif(fit_df3))>2
```

```
##                              Cement                                Slag
##                                TRUE                                TRUE
##                           `Fly Ash`                               Water
##                                TRUE                                TRUE
##                                  SP                    `Coarse Aggregate`
##                                TRUE                                TRUE
##                     `Fine Aggregate`                         Cement:Slag
##                                TRUE                                TRUE
##                    Cement:`Fly Ash`                        Cement:Water
##                                TRUE                                TRUE
##                             Cement:SP           Cement:`Coarse Aggregate`
##                                TRUE                                TRUE
##              Cement:`Fine Aggregate`                       Slag:`Fly Ash`
##                                TRUE                                TRUE
##                           Slag:Water                             Slag:SP
##                                TRUE                                TRUE
##              Slag:`Coarse Aggregate`                Slag:`Fine Aggregate`
```

```
##                                TRUE                                TRUE
##                    `Fly Ash`:Water                        `Fly Ash`:SP
##                                TRUE                                TRUE
##       `Fly Ash`:`Coarse Aggregate`          `Fly Ash`:`Fine Aggregate`
##                                TRUE                                TRUE
##                           Water:SP           Water:`Coarse Aggregate`
##                                TRUE                                TRUE
##             Water:`Fine Aggregate`              SP:`Coarse Aggregate`
##                                TRUE                                TRUE
##                SP:`Fine Aggregate` `Coarse Aggregate`:`Fine Aggregate`
##                                TRUE                                TRUE
```

Interpretation for adding and deleting variables :

If the goal of the model is to make predictions, then multicollinearity isn't a problem . But if interpretations about individual parameter is to be made then deleting one of the variable with multicollinearity helps.

Best Regression Model

```
# Comparing models using anova
anova(fit_df01,fit_df2,fit_df3)
```

```
## Analysis of Variance Table
##
## Model 1: `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate`
## Model 2: `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Slag:Water + `Coarse Aggregate`:`Fine Aggregate` +
##     Cement:`Fly Ash` + SP:Slag
## Model 3: `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Slag + Cement:`Fly Ash` + Cement:Water +
##     Cement:SP + Cement:`Coarse Aggregate` + Cement:`Fine Aggregate` +
##     Slag:`Fly Ash` + Slag:Water + Slag:SP + Slag:`Coarse Aggregate` +
##     Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:SP +
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:SP + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate` + `Coarse Aggregate`:`Fine Aggregate`
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     95 15671.3
## 2     91 12535.5  4    3135.8 7.4274 4.372e-05 ***
## 3     74  7810.5 17    4725.0 2.6334    0.0022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation for best model using anova :

The third model is the best model because it has the least RSS value compared to the other two model.

```
AIC(fit_df01,fit_df2,fit_df3)
```

```
##          df      AIC
## fit_df01  9 827.8614
## fit_df2  13 812.8653
## fit_df3  30 798.1361
```

###Interpretation for the best model using AIC :

Model with the smaller AIC value is preffered. Hence, model 3 if the best fit model.

Fine tune the variable selection

```
library(MASS)
stepAIC(fit_df3,direction = "backward")
```

```
## Start:  AIC=503.83
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Slag + Cement:`Fly Ash` + Cement:Water +
##     Cement:SP + Cement:`Coarse Aggregate` + Cement:`Fine Aggregate` +
##     Slag:`Fly Ash` + Slag:Water + Slag:SP + Slag:`Coarse Aggregate` +
##     Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:SP +
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:SP + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate` + `Coarse Aggregate`:`Fine Aggregate`
##
##                                       Df Sum of Sq     RSS    AIC
## - `Coarse Aggregate`:`Fine Aggregate`  1      8.07  7818.5 501.94
## - Water:SP                             1     14.69  7825.2 502.03
## - Cement:Slag                          1     23.49  7834.0 502.14
## - Cement:`Fine Aggregate`              1     28.31  7838.8 502.21
## - Slag:SP                              1     39.10  7849.6 502.35
## - SP:`Fine Aggregate`                  1     44.60  7855.1 502.42
## - SP:`Coarse Aggregate`                1     45.69  7856.2 502.44
## - `Fly Ash`:SP                         1     60.81  7871.3 502.63
## - Cement:`Fly Ash`                     1     73.34  7883.8 502.80
## - Cement:SP                            1     95.64  7906.1 503.09
## - Cement:`Coarse Aggregate`            1     96.05  7906.5 503.09
## <none>                                              7810.5 503.83
## - Water:`Coarse Aggregate`             1    339.20  8149.7 506.21
## - Slag:`Fly Ash`                       1    427.01  8237.5 507.32
## - Cement:Water                         1    443.15  8253.6 507.52
## - Water:`Fine Aggregate`               1    524.35  8334.8 508.53
## - `Fly Ash`:`Fine Aggregate`           1    669.57  8480.0 510.31
## - Slag:`Fine Aggregate`                1    787.99  8598.5 511.73
## - `Fly Ash`:`Coarse Aggregate`         1    953.06  8763.5 513.69
## - `Fly Ash`:Water                      1   1049.60  8860.1 514.82
## - Slag:`Coarse Aggregate`              1   1350.14  9160.6 518.26
## - Slag:Water                           1   2536.96 10347.4 530.81
##
## Step:  AIC=501.94
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Slag + Cement:`Fly Ash` + Cement:Water +
##     Cement:SP + Cement:`Coarse Aggregate` + Cement:`Fine Aggregate` +
```

```
##     Slag:`Fly Ash` + Slag:Water + Slag:SP + Slag:`Coarse Aggregate` +
##     Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:SP +
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:SP + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - Water:SP                      1     13.40  7831.9 500.12
## - Cement:Slag                   1     22.47  7841.0 500.24
## - Cement:`Fine Aggregate`       1     27.54  7846.1 500.30
## - Slag:SP                       1     36.31  7854.9 500.42
## - SP:`Fine Aggregate`           1     41.98  7860.5 500.49
## - SP:`Coarse Aggregate`         1     43.82  7862.4 500.52
## - `Fly Ash`:SP                  1     56.72  7875.3 500.69
## - Cement:`Fly Ash`              1     65.34  7883.9 500.80
## - Cement:SP                     1     91.88  7910.4 501.14
## - Cement:`Coarse Aggregate`     1     93.59  7912.1 501.17
## <none>                                     7818.5 501.94
## - Water:`Coarse Aggregate`      1    369.93  8188.5 504.70
## - Cement:Water                  1    439.03  8257.6 505.57
## - Slag:`Fly Ash`                1    445.11  8263.7 505.64
## - `Fly Ash`:`Fine Aggregate`    1    664.53  8483.1 508.34
## - Water:`Fine Aggregate`        1    669.44  8488.0 508.40
## - `Fly Ash`:`Coarse Aggregate`  1    945.10  8763.6 511.69
## - `Fly Ash`:Water               1   1086.69  8905.2 513.35
## - Slag:`Fine Aggregate`         1   1174.22  8992.8 514.35
## - Slag:`Coarse Aggregate`       1   1592.54  9411.1 519.04
## - Slag:Water                    1   2531.03 10349.6 528.83
##
## Step:  AIC=500.12
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Slag + Cement:`Fly Ash` + Cement:Water +
##     Cement:SP + Cement:`Coarse Aggregate` + Cement:`Fine Aggregate` +
##     Slag:`Fly Ash` + Slag:Water + Slag:SP + Slag:`Coarse Aggregate` +
##     Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:SP +
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:`Coarse Aggregate` + Water:`Fine Aggregate` + SP:`Coarse Aggregate` +
##     SP:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - Cement:Slag                   1     21.12  7853.1 498.39
## - Cement:`Fine Aggregate`       1     32.00  7863.9 498.54
## - Cement:`Fly Ash`              1     69.82  7901.8 499.03
## - Slag:SP                       1    105.10  7937.0 499.49
## - Cement:`Coarse Aggregate`     1    107.59  7939.5 499.52
## - SP:`Fine Aggregate`           1    127.46  7959.4 499.78
## <none>                                     7831.9 500.12
## - SP:`Coarse Aggregate`         1    227.11  8059.1 501.06
## - `Fly Ash`:SP                  1    232.92  8064.9 501.14
## - Water:`Coarse Aggregate`      1    366.79  8198.7 502.83
## - Cement:SP                     1    388.20  8220.1 503.10
## - Cement:Water                  1    432.99  8264.9 503.66
## - Slag:`Fly Ash`                1    433.65  8265.6 503.67
## - `Fly Ash`:`Fine Aggregate`    1    651.48  8483.4 506.35
```

```
## - Water:`Fine Aggregate`        1    664.43  8496.4 506.50
## - `Fly Ash`:`Coarse Aggregate`  1    937.46  8769.4 509.76
## - `Fly Ash`:Water               1   1074.34  8906.3 511.36
## - Slag:`Fine Aggregate`         1   1179.56  9011.5 512.57
## - Slag:`Coarse Aggregate`       1   1583.07  9415.0 517.08
## - Slag:Water                    1   2584.58 10416.5 527.49
##
## Step:  AIC=498.39
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:`Fly Ash` + Cement:Water + Cement:SP +
##     Cement:`Coarse Aggregate` + Cement:`Fine Aggregate` + Slag:`Fly Ash` +
##     Slag:Water + Slag:SP + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate`
##
##                                 Df Sum of Sq     RSS    AIC
## - Cement:`Fine Aggregate`        1    118.40  7971.5 497.94
## - Slag:SP                        1    123.02  7976.1 498.00
## - SP:`Fine Aggregate`            1    142.38  7995.5 498.25
## <none>                                        7853.1 498.39
## - Cement:`Fly Ash`               1    170.21  8023.3 498.60
## - `Fly Ash`:SP                   1    221.25  8074.3 499.26
## - Cement:`Coarse Aggregate`      1    225.76  8078.8 499.31
## - SP:`Coarse Aggregate`          1    229.06  8082.1 499.36
## - Water:`Coarse Aggregate`       1    347.22  8200.3 500.85
## - Cement:SP                      1    381.93  8235.0 501.29
## - Slag:`Fly Ash`                 1    535.69  8388.8 503.19
## - Cement:Water                   1    583.78  8436.9 503.78
## - `Fly Ash`:`Fine Aggregate`     1    647.01  8500.1 504.55
## - Water:`Fine Aggregate`         1    649.08  8502.1 504.57
## - `Fly Ash`:`Coarse Aggregate`   1    932.93  8786.0 507.96
## - `Fly Ash`:Water                1   1053.82  8906.9 509.36
## - Slag:`Fine Aggregate`          1   1501.00  9354.1 514.41
## - Slag:`Coarse Aggregate`        1   1742.63  9595.7 517.04
## - Slag:Water                     1   2694.68 10547.8 526.78
##
## Step:  AIC=497.94
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:`Fly Ash` + Cement:Water + Cement:SP +
##     Cement:`Coarse Aggregate` + Slag:`Fly Ash` + Slag:Water +
##     Slag:SP + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate`
##
##                                 Df Sum of Sq     RSS    AIC
## - Cement:`Fly Ash`               1     70.33  8041.8 496.84
## - Cement:`Coarse Aggregate`      1    107.36  8078.8 497.31
## - Slag:SP                        1    125.21  8096.7 497.54
## - SP:`Fine Aggregate`            1    129.12  8100.6 497.59
## <none>                                        7971.5 497.94
## - SP:`Coarse Aggregate`          1    206.92  8178.4 498.58
## - `Fly Ash`:SP                   1    221.11  8192.6 498.75
```

```
## - Water:`Coarse Aggregate`      1    294.27  8265.7 499.67
## - Cement:SP                     1    353.73  8325.2 500.41
## - Cement:Water                  1    472.06  8443.5 501.86
## - Water:`Fine Aggregate`        1    559.38  8530.9 502.92
## - `Fly Ash`:`Fine Aggregate`    1    559.89  8531.4 502.93
## - `Fly Ash`:`Coarse Aggregate`  1    814.57  8786.0 505.96
## - Slag:`Fly Ash`                1    899.57  8871.0 506.95
## - `Fly Ash`:Water               1    942.80  8914.3 507.45
## - Slag:`Fine Aggregate`         1   1583.02  9554.5 514.59
## - Slag:`Coarse Aggregate`       1   2383.18 10354.7 522.88
## - Slag:Water                    1   2816.46 10787.9 527.10
##
## Step:  AIC=496.84
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Cement:`Coarse Aggregate` +
##     Slag:`Fly Ash` + Slag:Water + Slag:SP + Slag:`Coarse Aggregate` +
##     Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:SP +
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:`Coarse Aggregate` + Water:`Fine Aggregate` + SP:`Coarse Aggregate` +
##     SP:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - Cement:`Coarse Aggregate`     1      69.2  8111.0 495.72
## - SP:`Fine Aggregate`           1     101.3  8143.1 496.13
## - Slag:SP                       1     102.0  8143.8 496.14
## <none>                                       8041.8 496.84
## - `Fly Ash`:SP                  1     166.6  8208.4 496.95
## - SP:`Coarse Aggregate`         1     177.7  8219.5 497.09
## - Cement:SP                     1     295.3  8337.1 498.56
## - Water:`Coarse Aggregate`      1     337.9  8379.7 499.08
## - Cement:Water                  1     431.3  8473.1 500.22
## - `Fly Ash`:`Fine Aggregate`    1     503.4  8545.2 501.09
## - Water:`Fine Aggregate`        1     531.3  8573.1 501.43
## - `Fly Ash`:`Coarse Aggregate`  1     755.4  8797.2 504.09
## - Slag:`Fly Ash`                1     879.7  8921.5 505.53
## - `Fly Ash`:Water               1     887.7  8929.5 505.63
## - Slag:`Fine Aggregate`         1    2413.7 10455.5 521.88
## - Slag:Water                    1    3023.5 11065.3 527.71
## - Slag:`Coarse Aggregate`       1    3311.9 11353.7 530.36
##
## Step:  AIC=495.72
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Slag:`Fly Ash` +
##     Slag:Water + Slag:SP + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - Slag:SP                       1      80.6  8191.6 494.74
## - SP:`Fine Aggregate`           1      85.4  8196.3 494.80
## - SP:`Coarse Aggregate`         1     153.1  8264.1 495.65
## <none>                                       8111.0 495.72
## - `Fly Ash`:SP                  1     159.9  8270.9 495.73
```

```
## - Cement:SP                     1     270.1  8381.0 497.10
## - Water:`Coarse Aggregate`      1     339.5  8450.4 497.95
## - Cement:Water                  1     384.9  8495.8 498.50
## - `Fly Ash`:`Fine Aggregate`    1     500.0  8611.0 499.88
## - Water:`Fine Aggregate`        1     579.3  8690.3 500.83
## - `Fly Ash`:`Coarse Aggregate`  1     730.6  8841.6 502.61
## - `Fly Ash`:Water               1     841.4  8952.4 503.89
## - Slag:`Fly Ash`                1    1165.7  9276.7 507.55
## - Slag:`Fine Aggregate`         1    2815.3 10926.2 524.41
## - Slag:Water                    1    2955.3 11066.3 525.72
## - Slag:`Coarse Aggregate`       1    3245.5 11356.5 528.39
##
## Step:  AIC=494.74
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Slag:`Fly Ash` +
##     Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate` + SP:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - SP:`Fine Aggregate`           1      12.6  8204.2 492.90
## - SP:`Coarse Aggregate`         1      76.3  8267.8 493.70
## - `Fly Ash`:SP                  1      85.7  8277.3 493.81
## <none>                                       8191.6 494.74
## - Cement:SP                     1     267.8  8459.4 496.06
## - Cement:Water                  1     340.1  8531.7 496.93
## - Water:`Coarse Aggregate`      1     368.3  8559.9 497.27
## - Water:`Fine Aggregate`        1     528.8  8720.4 499.19
## - `Fly Ash`:`Fine Aggregate`    1     595.7  8787.3 499.97
## - `Fly Ash`:`Coarse Aggregate`  1     812.1  9003.7 502.48
## - `Fly Ash`:Water               1     856.6  9048.1 502.99
## - Slag:`Fly Ash`                1    1150.2  9341.8 506.27
## - Slag:`Fine Aggregate`         1    2800.8 10992.4 523.03
## - Slag:Water                    1    3076.1 11267.7 525.58
## - Slag:`Coarse Aggregate`       1    3233.8 11425.4 527.01
##
## Step:  AIC=492.9
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Slag:`Fly Ash` +
##     Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate` +
##     SP:`Coarse Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - SP:`Coarse Aggregate`         1      63.8  8268.0 491.70
## - `Fly Ash`:SP                  1      80.7  8284.9 491.91
## <none>                                       8204.2 492.90
## - Cement:SP                     1     269.6  8473.8 494.23
## - Cement:Water                  1     348.2  8552.5 495.18
## - Water:`Coarse Aggregate`      1     400.3  8604.5 495.81
## - Water:`Fine Aggregate`        1     539.7  8743.9 497.46
## - `Fly Ash`:`Fine Aggregate`    1     583.2  8787.4 497.97
```

```
## - `Fly Ash`:`Coarse Aggregate`  1     799.9  9004.1 500.48
## - `Fly Ash`:Water               1     846.9  9051.2 501.02
## - Slag:`Fly Ash`                1    1138.6  9342.8 504.29
## - Slag:`Fine Aggregate`         1    2839.1 11043.4 521.51
## - Slag:Water                    1    3122.3 11326.5 524.12
## - Slag:`Coarse Aggregate`       1    3332.7 11536.9 526.01
##
## Step:  AIC=491.7
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Slag:`Fly Ash` +
##     Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:SP + `Fly Ash`:`Coarse Aggregate` +
##     `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - `Fly Ash`:SP                  1     110.0  8378.0 491.06
## <none>                                       8268.0 491.70
## - Cement:SP                     1     242.6  8510.6 492.68
## - Water:`Coarse Aggregate`      1     358.4  8626.5 494.07
## - Cement:Water                  1     395.6  8663.7 494.51
## - `Fly Ash`:`Fine Aggregate`    1     556.0  8824.1 496.40
## - Water:`Fine Aggregate`        1     686.0  8954.1 497.91
## - `Fly Ash`:`Coarse Aggregate`  1     748.7  9016.7 498.63
## - `Fly Ash`:Water               1     912.2  9180.2 500.48
## - Slag:`Fly Ash`                1    1096.1  9364.2 502.52
## - Slag:`Fine Aggregate`         1    2940.8 11208.8 521.04
## - Slag:Water                    1    3469.9 11738.0 525.79
## - Slag:`Coarse Aggregate`       1    3973.1 12241.1 530.12
##
## Step:  AIC=491.06
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Cement:SP + Slag:`Fly Ash` +
##     Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:`Coarse Aggregate` + Water:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## - Cement:SP                     1     136.5  8514.5 490.72
## <none>                                       8378.0 491.06
## - Water:`Coarse Aggregate`      1     402.7  8780.8 493.90
## - Cement:Water                  1     434.4  8812.5 494.27
## - `Fly Ash`:`Fine Aggregate`    1     572.9  8951.0 495.87
## - Water:`Fine Aggregate`        1     718.2  9096.2 497.53
## - `Fly Ash`:`Coarse Aggregate`  1     825.1  9203.1 498.73
## - `Fly Ash`:Water               1     928.9  9306.9 499.89
## - Slag:`Fly Ash`                1    1435.8  9813.8 505.35
## - Slag:`Fine Aggregate`         1    2897.8 11275.8 519.66
## - Slag:Water                    1    3572.7 11950.7 525.64
## - Slag:`Coarse Aggregate`       1    3892.7 12270.7 528.36
##
## Step:  AIC=490.72
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate` + Cement:Water + Slag:`Fly Ash` + Slag:Water +
##     Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` + `Fly Ash`:Water +
```

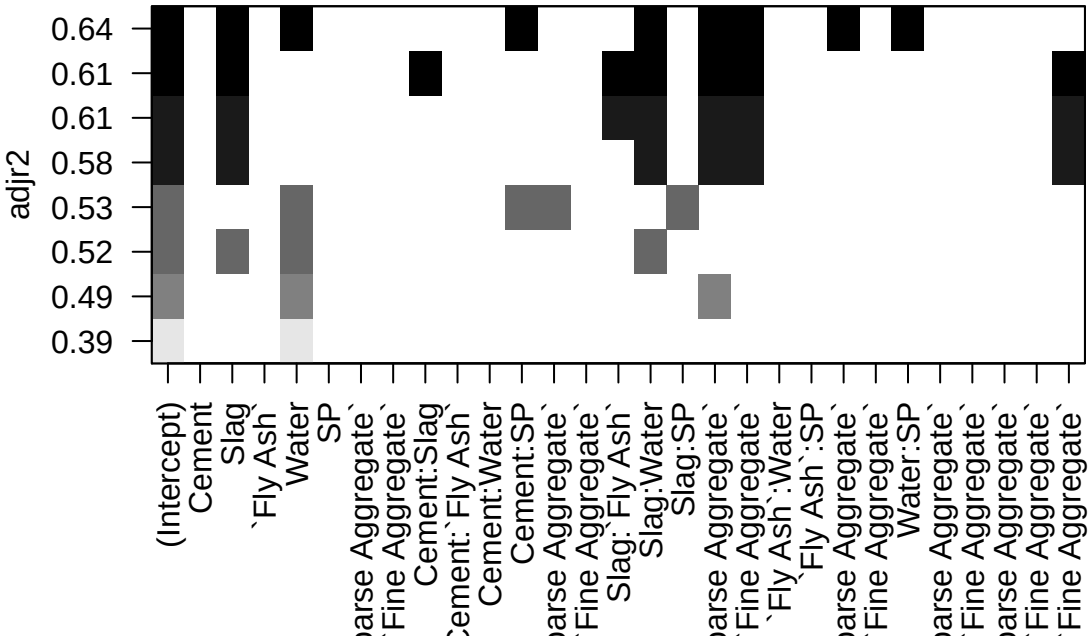
```
##     `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:`Coarse Aggregate` + Water:`Fine Aggregate`
##
##                                Df Sum of Sq     RSS    AIC
## <none>                                      8514.5 490.72
## - Cement:Water                  1     415.5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

##
## Call:
## lm(formula = `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water +
##     SP + `Coarse Aggregate` + `Fine Aggregate` + Cement:Water +
##     Slag:`Fly Ash` + Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` +
##     `Fly Ash`:Water + `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` +
##     Water:`Coarse Aggregate` + Water:`Fine Aggregate`, data = df)
##
## Coefficients:
##                   (Intercept)                        Cement
##                    872.162339                     -0.188274
##                          Slag                     `Fly Ash`
##                     -6.278219                     -2.399647
##                         Water                            SP
##                     -5.574350                      1.867179
##           `Coarse Aggregate`              `Fine Aggregate`
##                     -0.337620                     -0.768420
##                 Cement:Water                Slag:`Fly Ash`
##                      0.002577                      0.001353
##                   Slag:Water       Slag:`Coarse Aggregate`
##                      0.012597                      0.002325
##        Slag:`Fine Aggregate`                `Fly Ash`:Water
##                      0.002588                      0.004615
## `Fly Ash`:`Coarse Aggregate`    `Fly Ash`:`Fine Aggregate`
##                      0.001030                      0.001121
##     Water:`Coarse Aggregate`         Water:`Fine Aggregate`
##                      0.002183                      0.004231
```

```r
# All subset regression
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.6.2
```

```r
leaps <- regsubsets(`Slump Flow` ~ Cement+Slag+`Fly Ash`+Water+SP+`Coarse Aggregate`+`Fine Aggregate`+C
`Coarse Aggregate`:`Fine Aggregate`,data = df)
plot(leaps, scale = "adjr2")
```



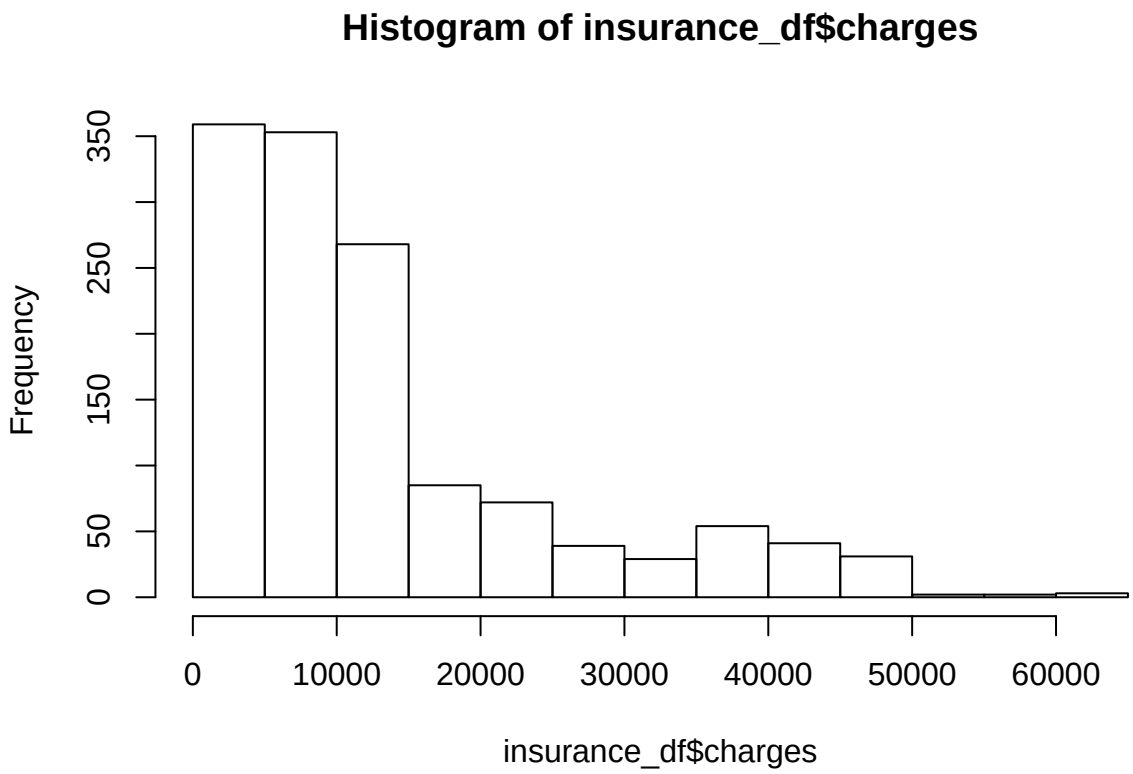
We can conclude from the above results that the below model is the most suitable linear model. `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` + `Fine Aggregate` + Cement:Water + Slag:`Fly Ash` + Slag:Water + Slag:`Coarse Aggregate` + Slag:`Fine Aggregate` + `Fly Ash`:Water + `Fly Ash`:`Coarse Aggregate` + `Fly Ash`:`Fine Aggregate` + Water:`Coarse Aggregate` + Water:`Fine Aggregate`

Problem 03

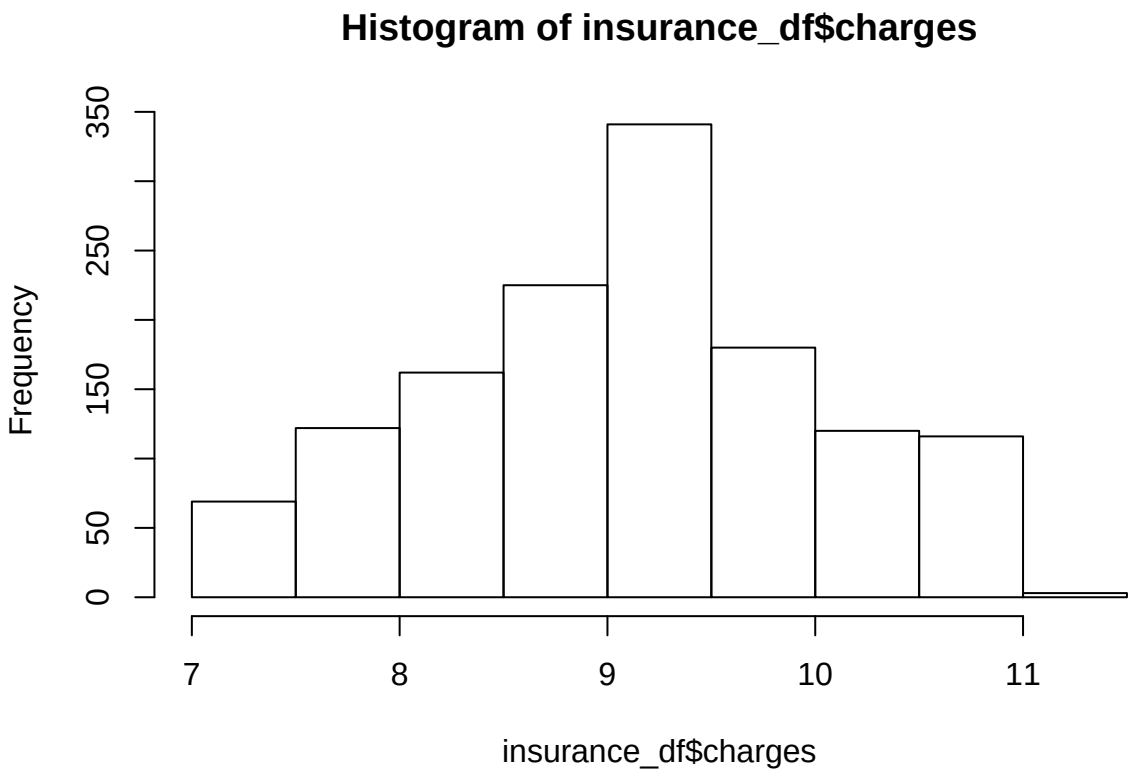
```
insurance_df <- read.csv("insurance.csv", stringsAsFactors = TRUE)
summary(insurance_df$charges)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4740    9382   13270   16640   63770
```

```
# Check for Normality
hist(insurance_df$charges)
```




```
# Log Transformation
insurance_df$charges <- log(insurance_df$charges)
hist(insurance_df$charges)
```




```
summary(insurance_df$charges)
```

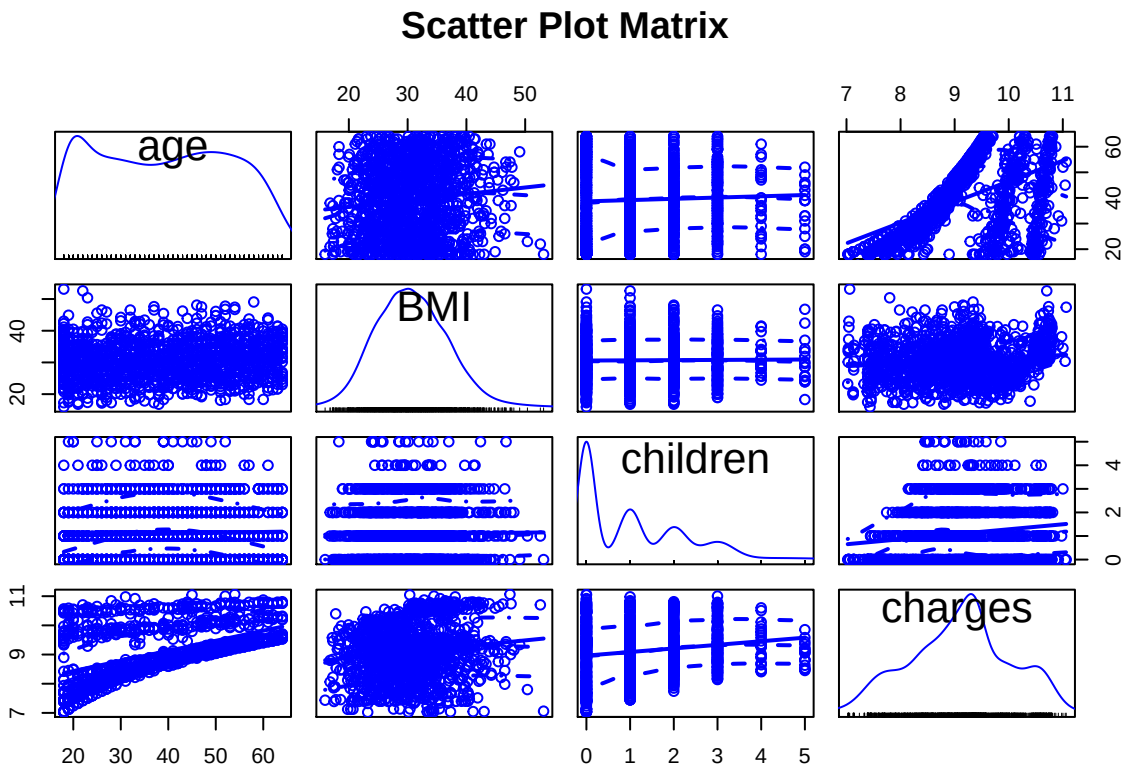
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.023   8.464   9.147   9.099   9.720  11.063
```

a) From the summary of our dependent variable "charges", it can be seen that it is not in normal form . We used log function to normalize the variable and we can see from the histogram that now our dependent variable is almost normal.

```
input_df <- insurance_df[ ,c(1,3,4,7)]
cor(input_df)
```

```
##                age       BMI  children   charges
## age      1.0000000 0.1092719 0.0424690 0.5278340
## BMI      0.1092719 1.0000000 0.0127589 0.1326694
## children 0.0424690 0.0127589 1.0000000 0.1613363
## charges  0.5278340 0.1326694 0.1613363 1.0000000
```

```
scatterplotMatrix(input_df, spread = FALSE, lty.smooth = 2, main = "Scatter Plot Matrix" )
```



b) The above scatter plot is based on four variables selected i.e age , BMI, children and charges. The principal diagonal contains density and rug plots for each variables . It is also seen that BMI is normally distributed while all others show bimodal behaviour.

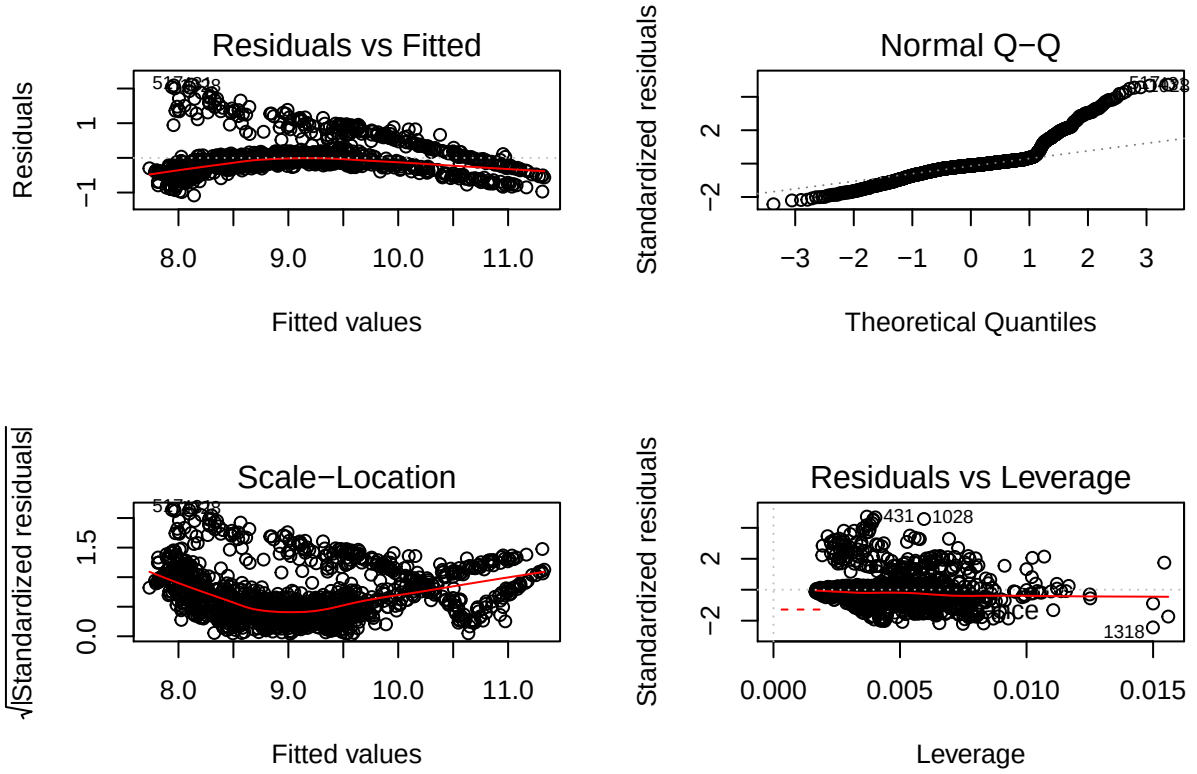
```
# Building regression model
fit <- lm(charges ~ age+BMI+sex+children+smoker, data=insurance_df)
summary(fit)
```

```
##
## Call:
## lm(formula = charges ~ age + BMI + sex + children + smoker, data = insurance_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08241 -0.20315 -0.05185  0.07057  2.11173
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.0121103  0.0701685  99.932  < 2e-16 ***
## age          0.0347158  0.0008781  39.536  < 2e-16 ***
## BMI          0.0109087  0.0020225   5.394 8.16e-08 ***
## sexmale     -0.0750088  0.0245899  -3.050  0.00233 **
## children     0.1017275  0.0101688  10.004  < 2e-16 ***
## smokeryes    1.5502366  0.0304293  50.946  < 2e-16 ***
## ---
```

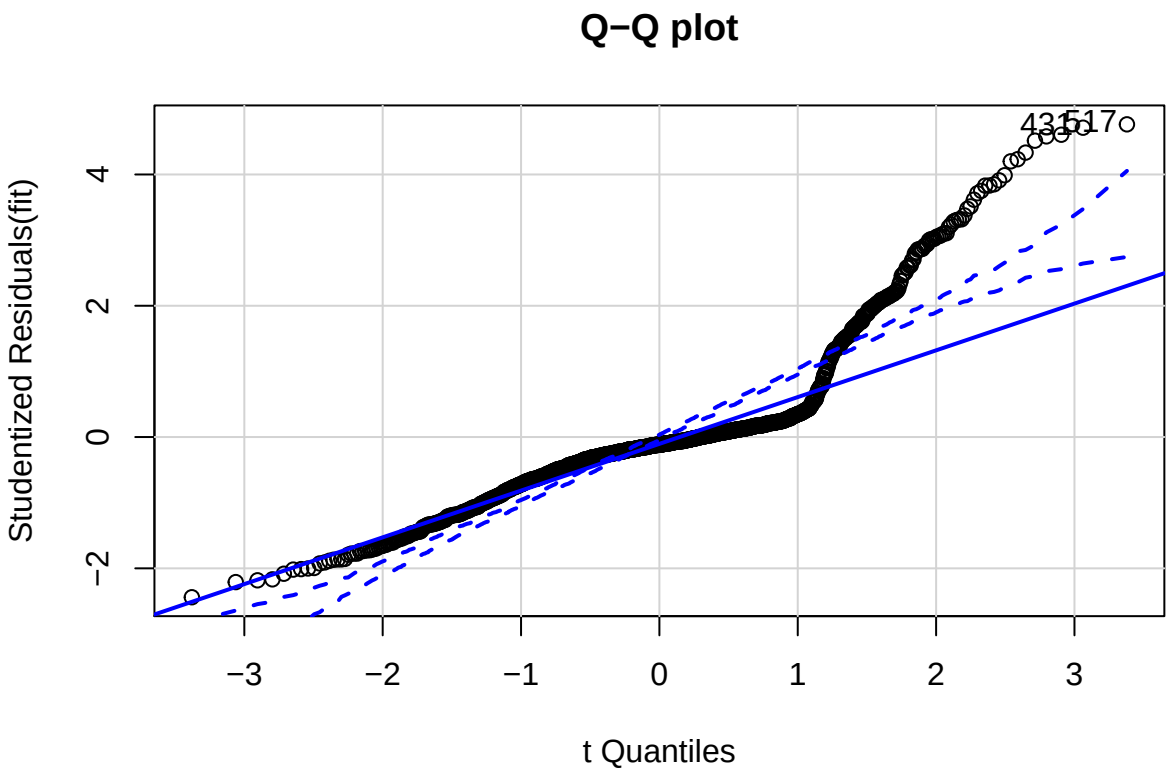
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4477 on 1332 degrees of freedom
## Multiple R-squared:  0.7638, Adjusted R-squared:  0.7629
## F-statistic: 861.5 on 5 and 1332 DF,  p-value: < 2.2e-16
```

c) The regression model shows the effect one variable has over another variable while keeping control over other variables. For instance, regression coefficient for BMI is 0.010 which means an increase of 1% BMI increases the dependent variable by 0.010%.

```r
# Typical approach
par(mfrow=c(2,2))
plot(fit)
```




```r
# Enhanced approach
# Normality error
library(stats)
par(mfrow=c(1,1))
qqPlot(fit, id.method="identify" , labels = row.names(data_df) , simulate = TRUE, main = "Q-Q plot")
```

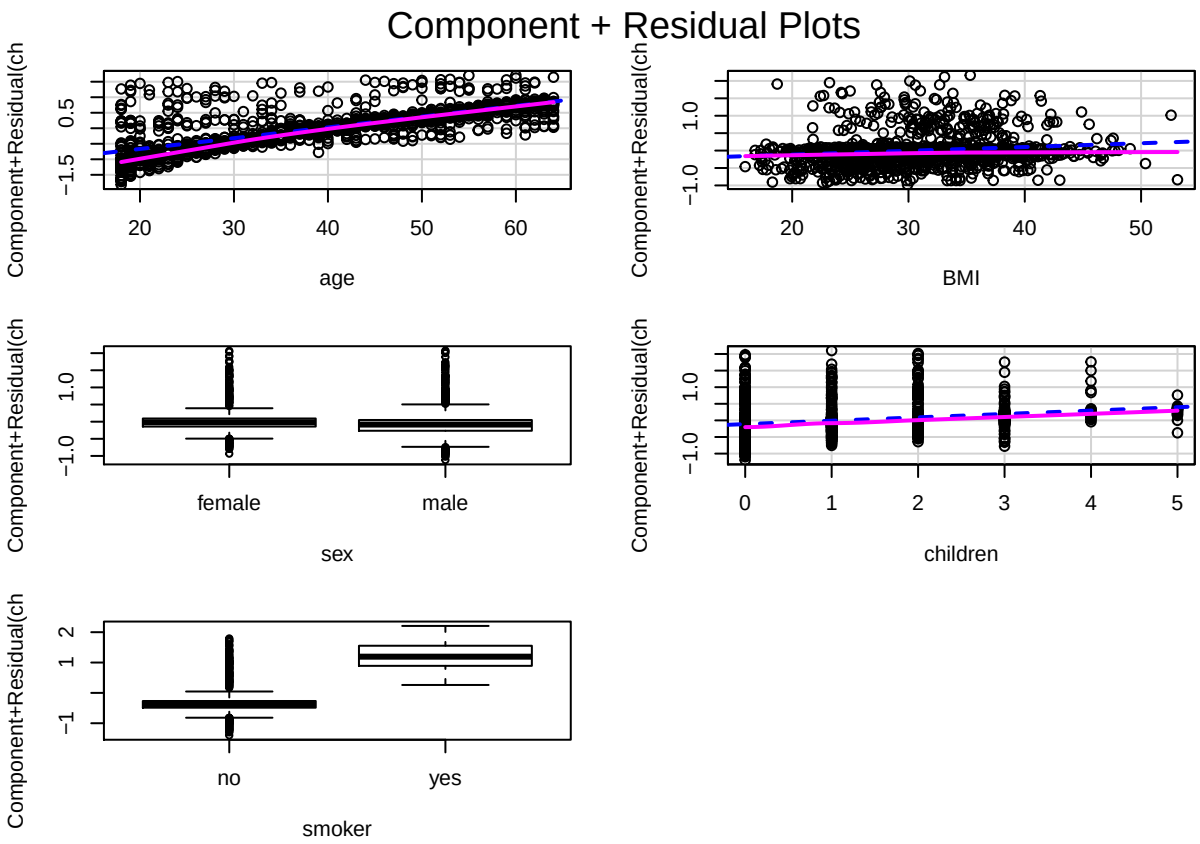



```
## [1] 431 517
```

```
# Independence of errors
durbinWatsonTest(fit)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02701482      2.051941   0.326
##  Alternative hypothesis: rho != 0
```

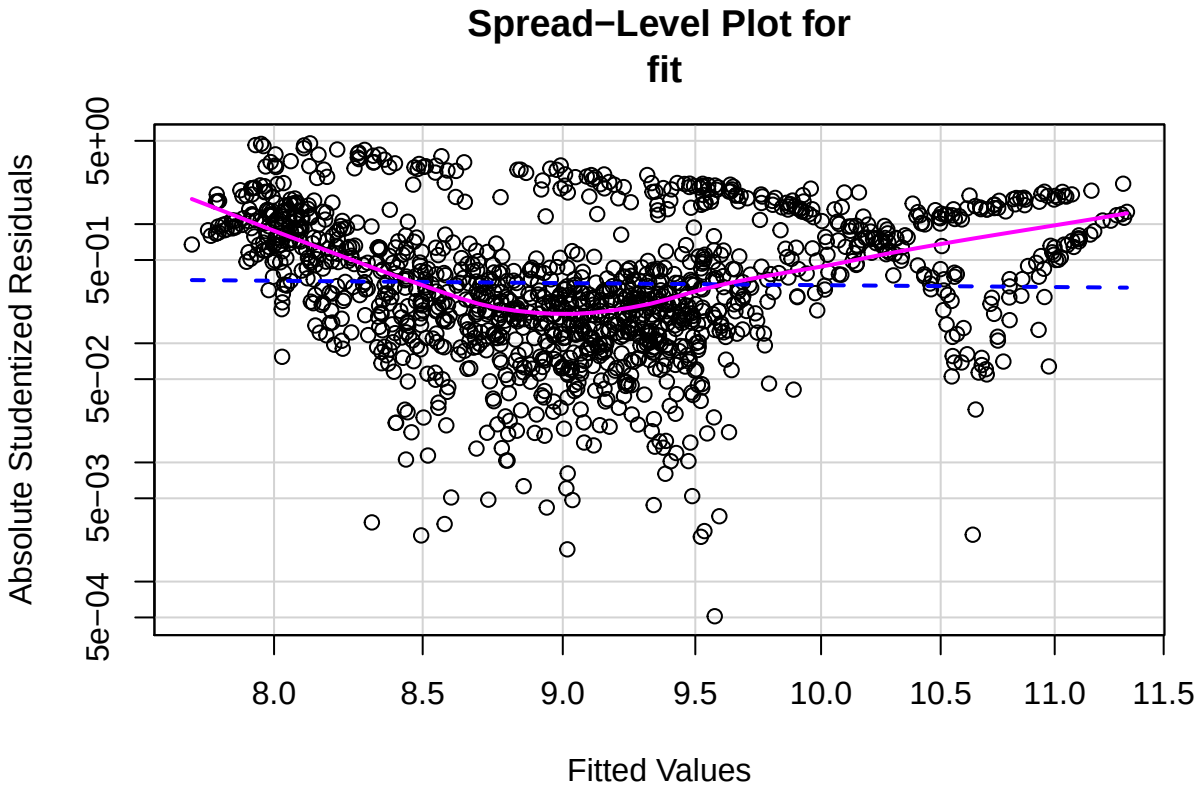
```
# Linearlity
crPlots(fit)
```




```
# Homoscedasticity
ncvTest(fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 65.33192, Df = 1, p = 6.3288e-16
```

```
spreadLevelPlot(fit)
```




```
##
## Suggested power transformation:  1.382722
```

```
# Problem 3(e)
obs <- insurance_df[insurance_df$BMI >= 30.0,]
nor <- insurance_df[insurance_df$BMI <= 30.0,]
fit1 <- lm(charges ~ age+ I(age^2)+BMI+sex+children+smoker, data=obs)
summary(fit1)
```

```
##
## Call:
## lm(formula = charges ~ age + I(age^2) + BMI + sex + children +
##     smoker, data = obs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82841 -0.19343 -0.05953  0.05978  2.20102
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.116e+00  1.988e-01  35.792  < 2e-16 ***
## age          5.305e-02  7.911e-03   6.706 4.12e-11 ***
## I(age^2)    -2.362e-04  9.776e-05  -2.416  0.01594 *
## BMI         -1.292e-03  4.051e-03  -0.319  0.74987
## sexmale     -9.535e-02  3.322e-02  -2.870  0.00423 **
```

```
## children     8.787e-02  1.442e-02   6.094 1.81e-09 ***
## smokeryes    1.854e+00  4.112e-02  45.084  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4379 on 700 degrees of freedom
## Multiple R-squared:  0.8052, Adjusted R-squared:  0.8036
## F-statistic: 482.4 on 6 and 700 DF,  p-value: < 2.2e-16
```

```
fit2 <- lm(charges ~ age+I(age^2)+sex+children+smoker,data = nor)
summary(fit2)
```

```
##
## Call:
## lm(formula = charges ~ age + I(age^2) + sex + children + smoker,
##     data = nor)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78199 -0.18384 -0.06455  0.03505  1.98200
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.9785631  0.1487025  46.930  < 2e-16 ***
## age          0.0542872  0.0083961   6.466 2.03e-10 ***
## I(age^2)    -0.0002319  0.0001061  -2.186   0.0292 *
## sexmale     -0.0758605  0.0330646  -2.294   0.0221 *
## children     0.0989099  0.0143139   6.910 1.19e-11 ***
## smokeryes    1.2122789  0.0407928  29.718  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4143 on 627 degrees of freedom
## Multiple R-squared:  0.7478, Adjusted R-squared:  0.7458
## F-statistic: 371.8 on 5 and 627 DF,  p-value: < 2.2e-16
```

e) Here we have divided data into two data frames one having BMI greater than 30 and other having BMI less than 30. We made separate regression model for each data frame and we can see that coefficient of each variables have changed. For the regression model having BMI greater than 30 the value of coefficient is very small.

Problem 04

```
library(readxl)
library(car)
forestFire_df <- read_xlsx("Forest Fires Data.xlsx")
forestFire_df$Area <- log1p(forestFire_df$Area)
ff_df <- forestFire_df[,c(13,1:12)]

fit_ff1 <- lm(Area ~ X+Y+FFMC+DMC+DC+ISI, data=ff_df)
summary(fit_ff1)
```

```
##
## Call:
## lm(formula = Area ~ X + Y + FFMC + DMC + DC + ISI, data = ff_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4889 -1.0771 -0.6240  0.9032  5.7551
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3468427  1.1815182  -0.294    0.769
## X            0.0387271  0.0316779   1.223    0.222
## Y            0.0103899  0.0600377   0.173    0.863
## FFMC         0.0129021  0.0137590   0.938    0.349
## DMC          0.0008730  0.0013691   0.638    0.524
## DC           0.0002385  0.0003446   0.692    0.489
## ISI         -0.0182500  0.0160833  -1.135    0.257
##
## Residual standard error: 1.398 on 510 degrees of freedom
## Multiple R-squared:  0.01282,    Adjusted R-squared:  0.001206
## F-statistic: 1.104 on 6 and 510 DF,  p-value: 0.3588
```

```
fit_ff2 <- lm(Area ~ X+Y+Temp+RH+Wind+Rain, data=ff_df)
summary(fit_ff1)
```

```
##
## Call:
## lm(formula = Area ~ X + Y + FFMC + DMC + DC + ISI, data = ff_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4889 -1.0771 -0.6240  0.9032  5.7551
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3468427  1.1815182  -0.294    0.769
## X            0.0387271  0.0316779   1.223    0.222
## Y            0.0103899  0.0600377   0.173    0.863
## FFMC         0.0129021  0.0137590   0.938    0.349
## DMC          0.0008730  0.0013691   0.638    0.524
## DC           0.0002385  0.0003446   0.692    0.489
## ISI         -0.0182500  0.0160833  -1.135    0.257
##
## Residual standard error: 1.398 on 510 degrees of freedom
## Multiple R-squared:  0.01282,    Adjusted R-squared:  0.001206
## F-statistic: 1.104 on 6 and 510 DF,  p-value: 0.3588
```

```
fit_ff3<- lm(Area ~ FFMC+DMC+DC+ISI, data=ff_df)
summary(fit_ff3)
```

```
##
## Call:
## lm(formula = Area ~ FFMC + DMC + DC + ISI, data = ff_df)
```

```
##
## 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
```

```
fit_ff4<- lm(Area ~ Temp+RH+Wind+Rain, data=ff_df)
summary(fit_ff4)
```

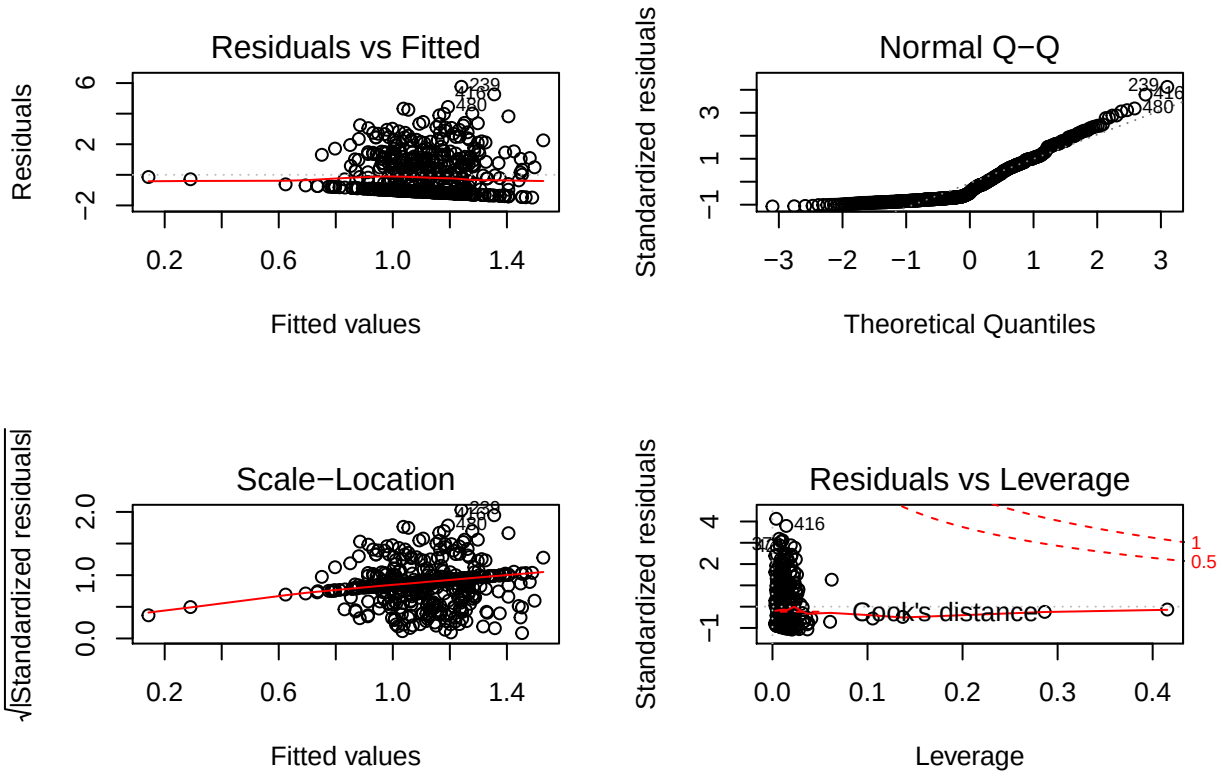
```
##
## Call:
## lm(formula = Area ~ Temp + RH + Wind + Rain, data = ff_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3993 -1.0978 -0.7081  0.9121  5.7593
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.742041   0.443148   1.674   0.0946 .
## 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
```

```
fit_ff5<- lm(Area ~ FFMC+DMC+DC+ISI+Temp+RH+Wind+Rain, data=ff_df)
summary(fit_ff5)
```

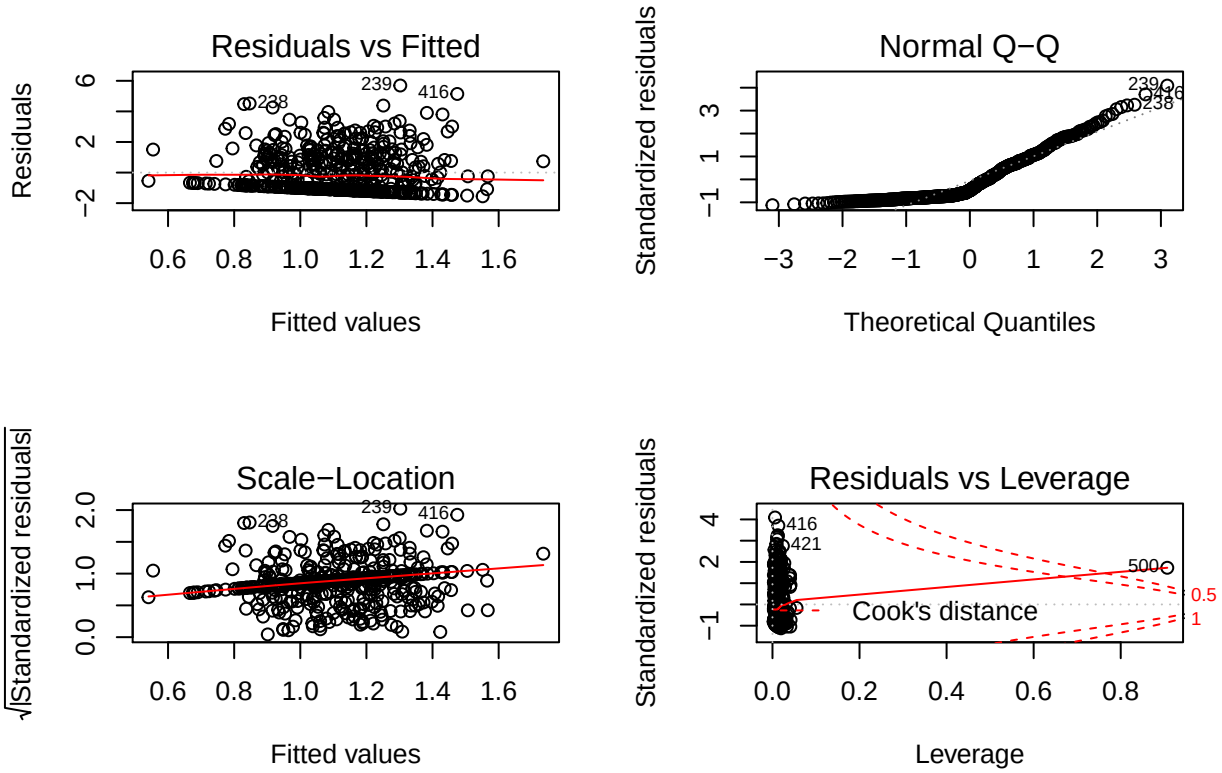
```
##
## Call:
## lm(formula = Area ~ FFMC + DMC + DC + ISI + Temp + RH + Wind +
##     Rain, data = ff_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5203 -1.1129 -0.6158  0.8787  5.7121
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2224140  1.3604350   0.163    0.870
## FFMC         0.0077082  0.0144884   0.532    0.595
## DMC          0.0011915  0.0014642   0.814    0.416
## DC           0.0002737  0.0003570   0.767    0.444
## ISI         -0.0239494  0.0169248  -1.415    0.158
## Temp         0.0024618  0.0172593   0.143    0.887
## RH          -0.0051729  0.0051889  -0.997    0.319
## Wind         0.0757669  0.0366155   2.069    0.039 *
## Rain         0.0965122  0.2121461   0.455    0.649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.395 on 508 degrees of freedom
## Multiple R-squared:  0.01988,    Adjusted R-squared:  0.004446
## F-statistic: 1.288 on 8 and 508 DF,  p-value: 0.2472
```

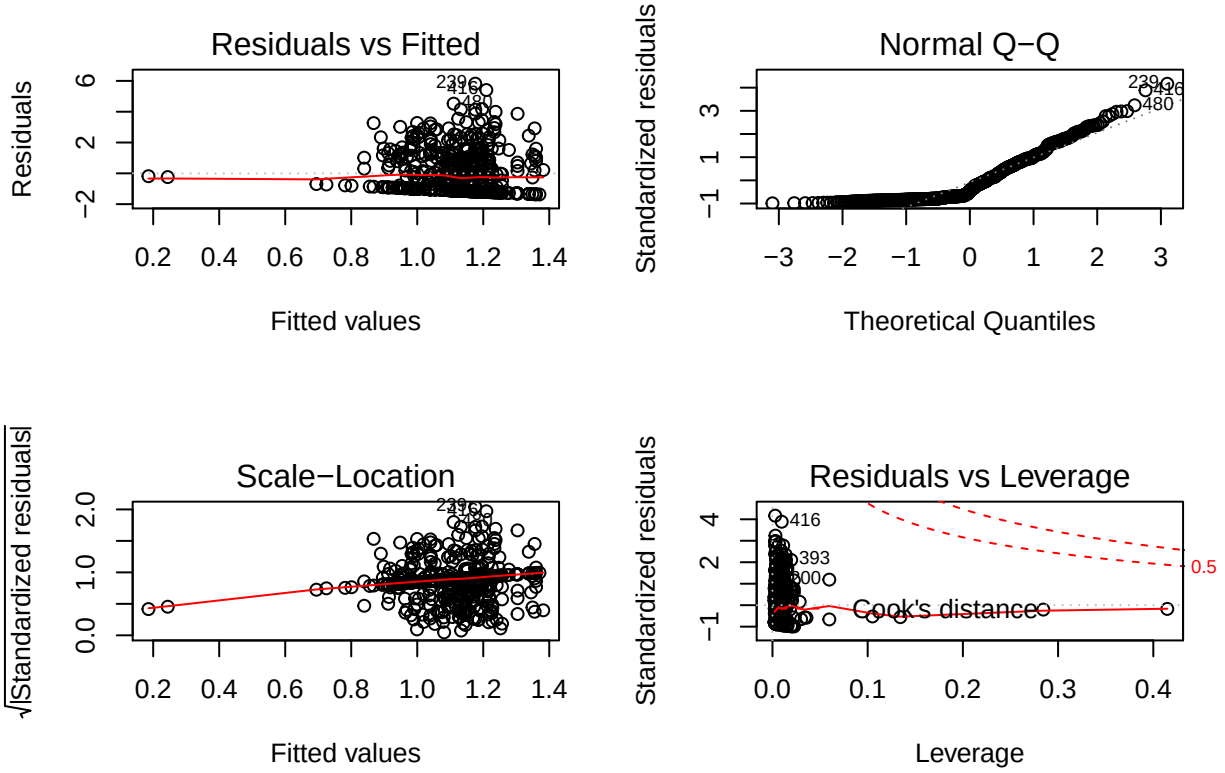
```r
# Regression Diagnostics
# Typical Approach - Model 1
par(mfrow=c(2,2))
plot(fit_ff1)
```



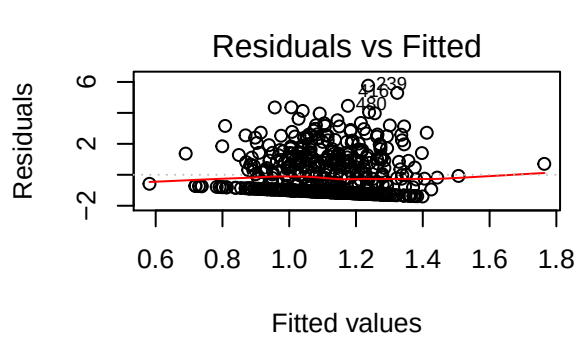
```
# Typical Approach - Model 2
par(mfrow=c(2,2))
plot(fit_ff2)
```

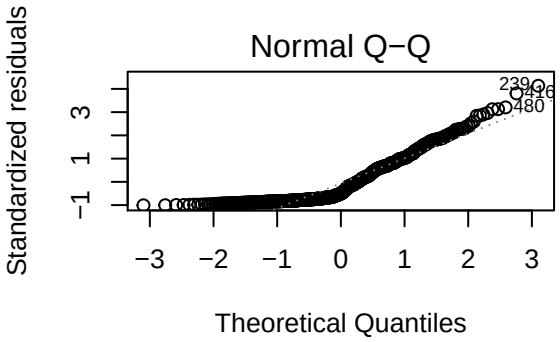


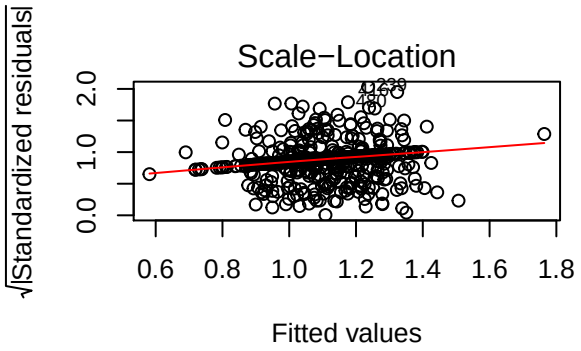

```
# Typical Approach - Model 3
par(mfrow=c(2,2))
plot(fit_ff3)
```

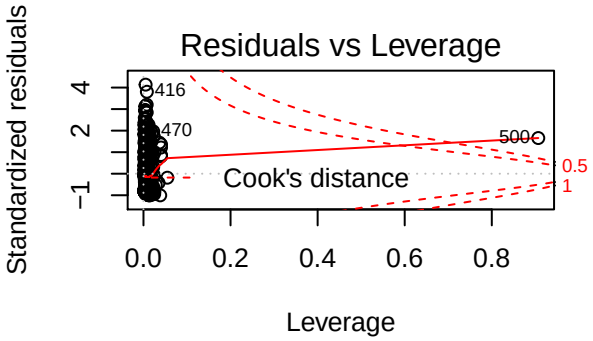



```
# Typical Approach - Model 3
par(mfrow=c(2,2))
plot(fit_ff4)
```

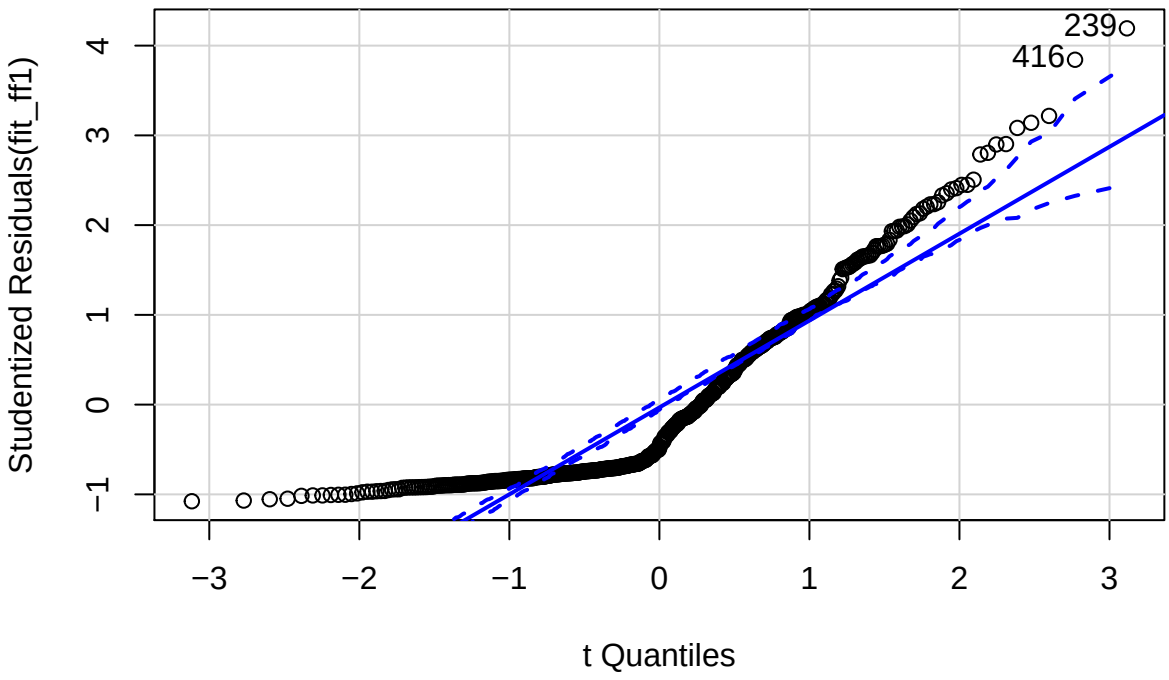






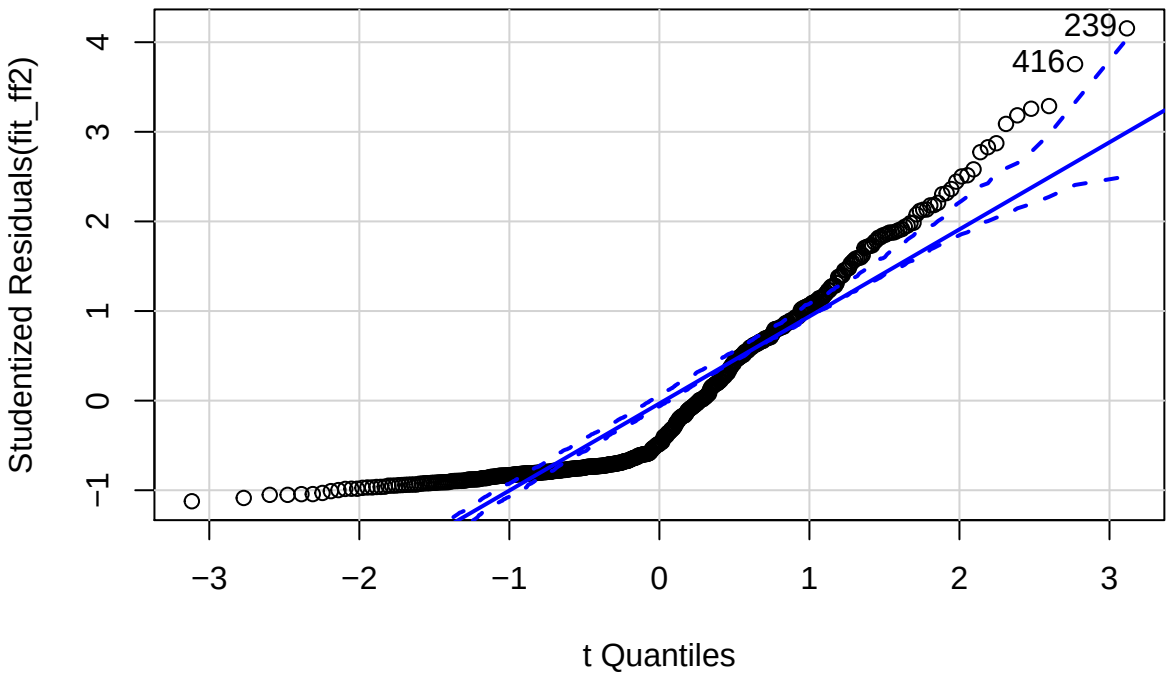



```
# Enhanced Approach
# qqPlot - Model 1
par(mfrow=c(1,1))
qqPlot(fit_ff1,labels = row.names(df), id.method = "identify",simulate = TRUE)
```



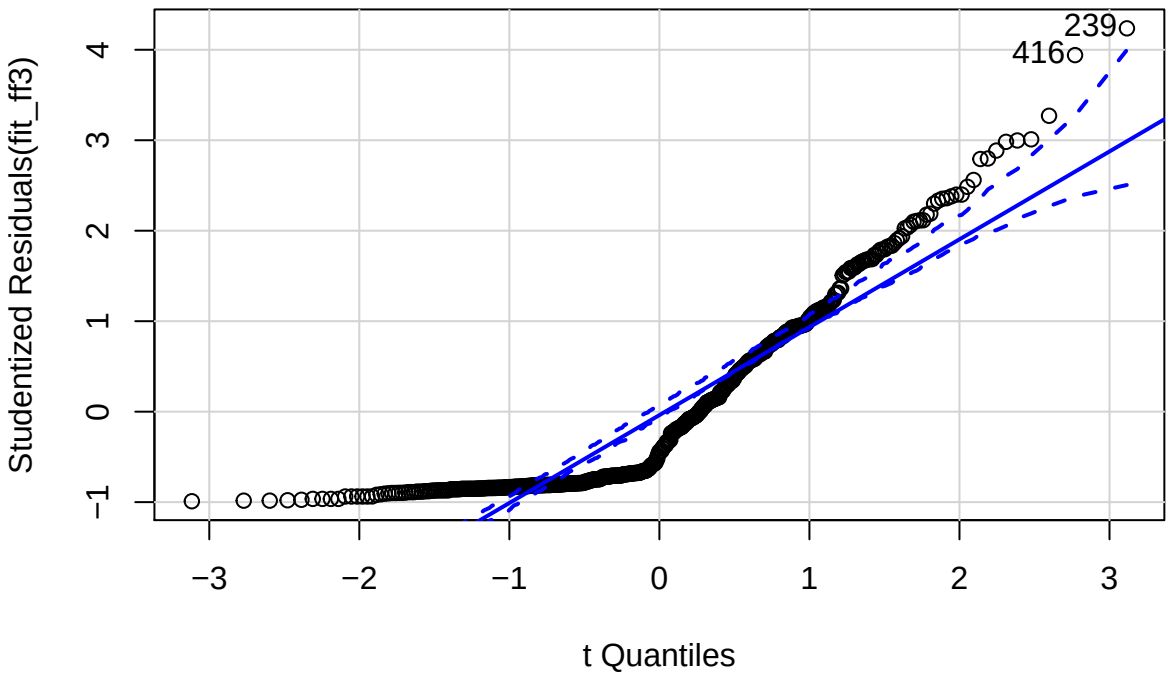

```
## [1] 239 416
```

```
# qqPlot - Model 2
par(mfrow=c(1,1))
qqPlot(fit_ff2,labels = row.names(df), id.method = "identify",simulate = TRUE)
```



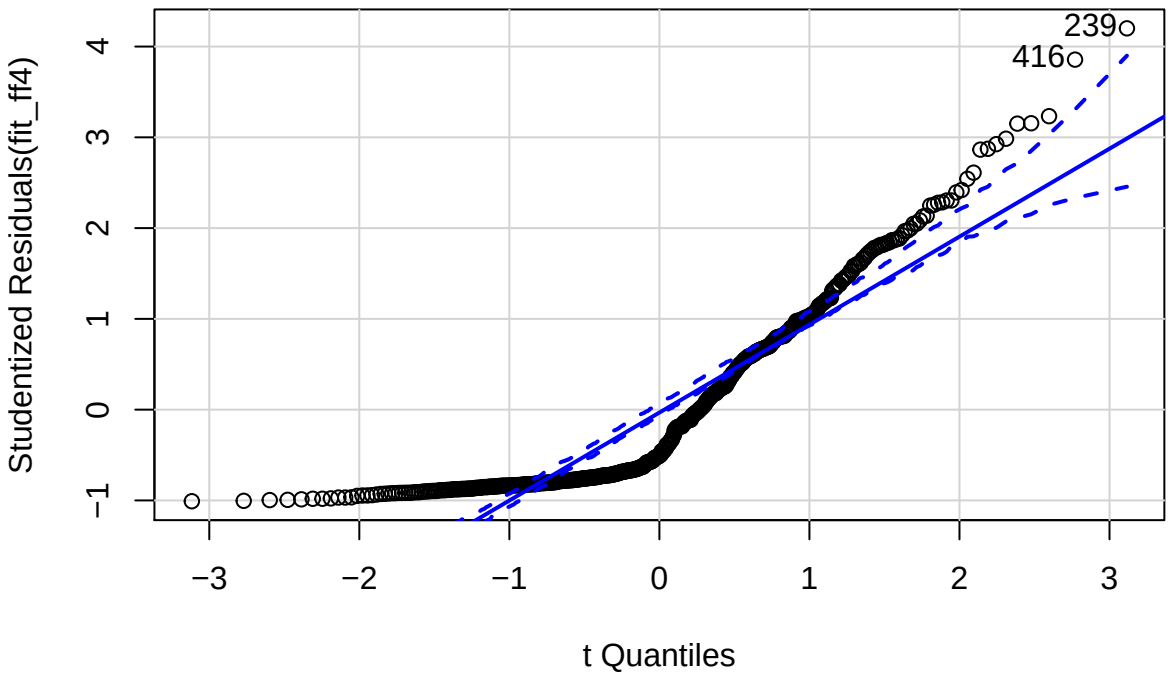

```
## [1] 239 416
```

```
# qqPlot - Model 3
qqPlot(fit_ff3,labels = row.names(df),id.method = "identify",simulate = TRUE)
```




```
## [1] 239 416
```

```
# qqPlot - Model 4
qqPlot(fit_ff4,labels = row.names(df),id.method = "identify",simulate = TRUE)
```




```
## [1] 239 416
```

```
# Confidence Interval
confint(fit_ff1)
```

```
##                     2.5 %       97.5 %
## (Intercept) -2.6680845685 1.9743992250
## X           -0.0235081634 0.1009624067
## Y           -0.1075617543 0.1283415037
## FFMC        -0.0141292332 0.0399334549
## DMC         -0.0018166633 0.0035627564
## DC          -0.0004383686 0.0009154575
## ISI         -0.0498475955 0.0133476893
```

```
confint(fit_ff2)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.429970701 1.494290339
## X           -0.026403687 0.098032922
## Y           -0.103435897 0.130012344
## Temp        -0.012446746 0.038464793
## RH          -0.012118320 0.005617314
## Wind        -0.007118641 0.132208031
## Rain        -0.349814888 0.483823202
```

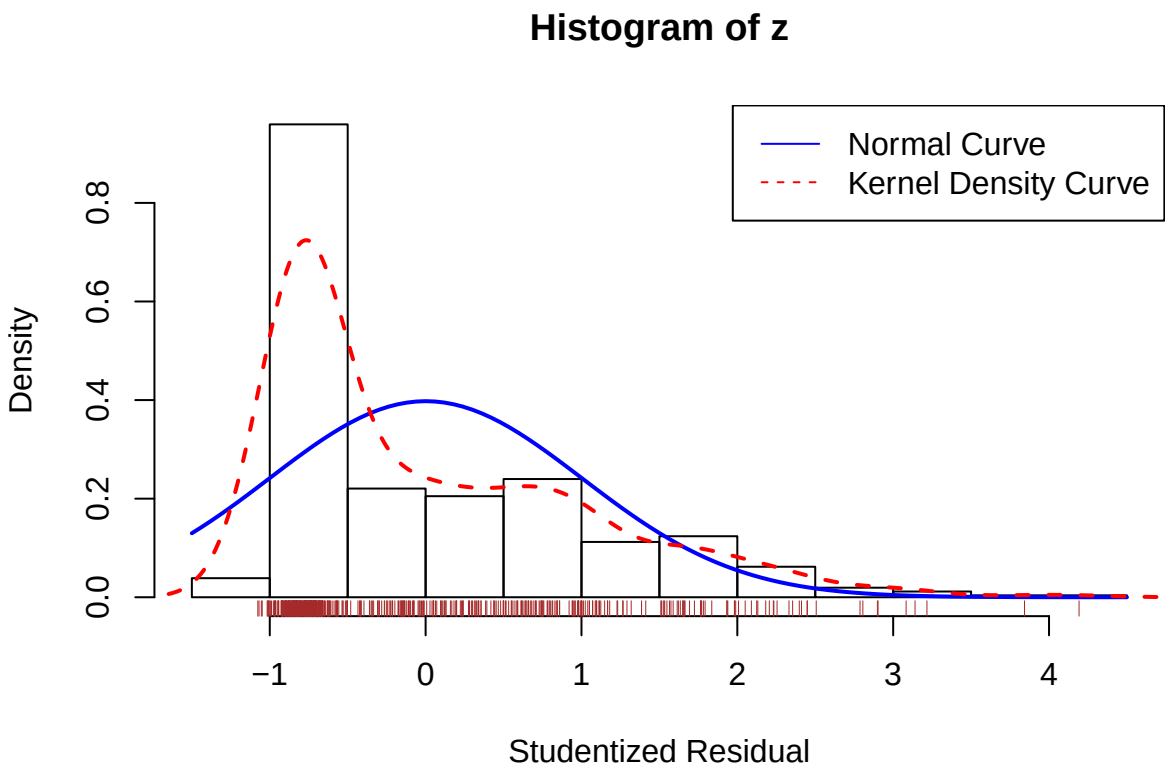
```
confint(fit_ff3)
```

```
##                     2.5 %       97.5 %
## (Intercept) -2.3472119097 2.1768681640
## FFMC        -0.0143665978 0.0396903696
## DMC         -0.0017458795 0.0035927209
## DC          -0.0004775953 0.0008632444
## ISI         -0.0492807632 0.0139052270
```

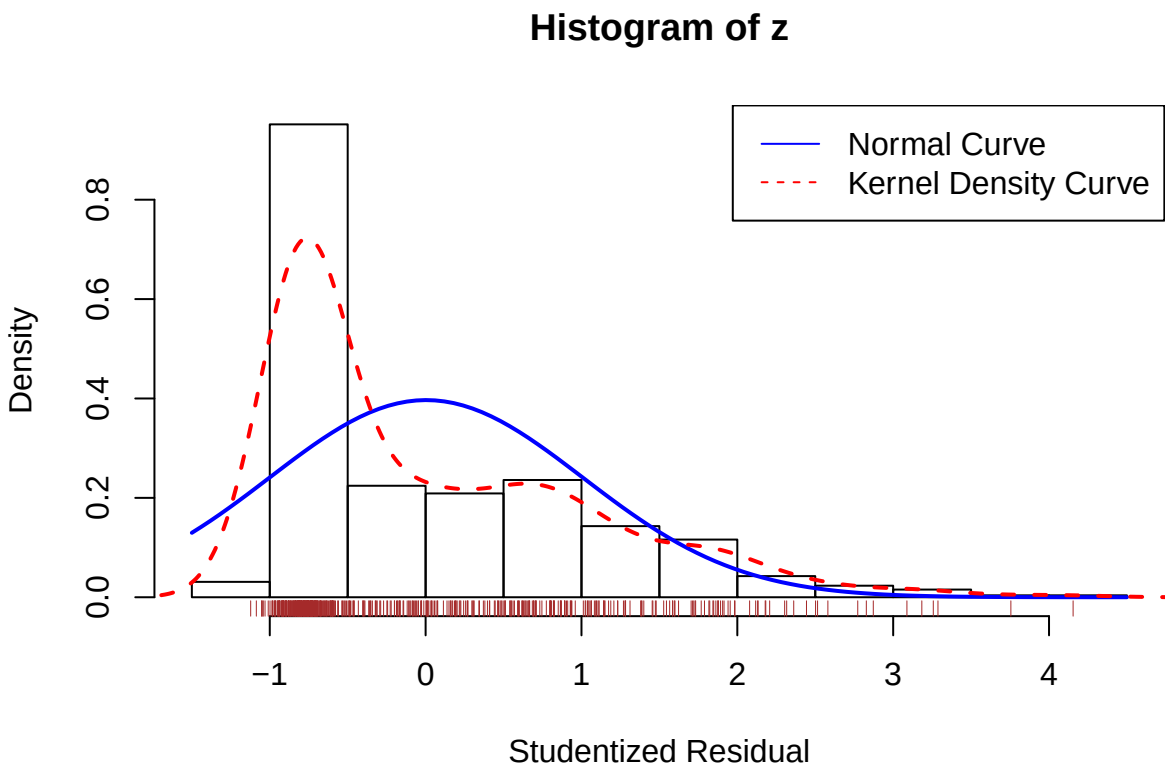
```
confint(fit_ff4)
```

```
##                    2.5 %      97.5 %
## (Intercept) -0.128571013 1.612653538
## Temp        -0.012691787 0.038224577
## RH          -0.011686420 0.006017963
## Wind        -0.007034024 0.132240956
## Rain        -0.331038568 0.501372815
```

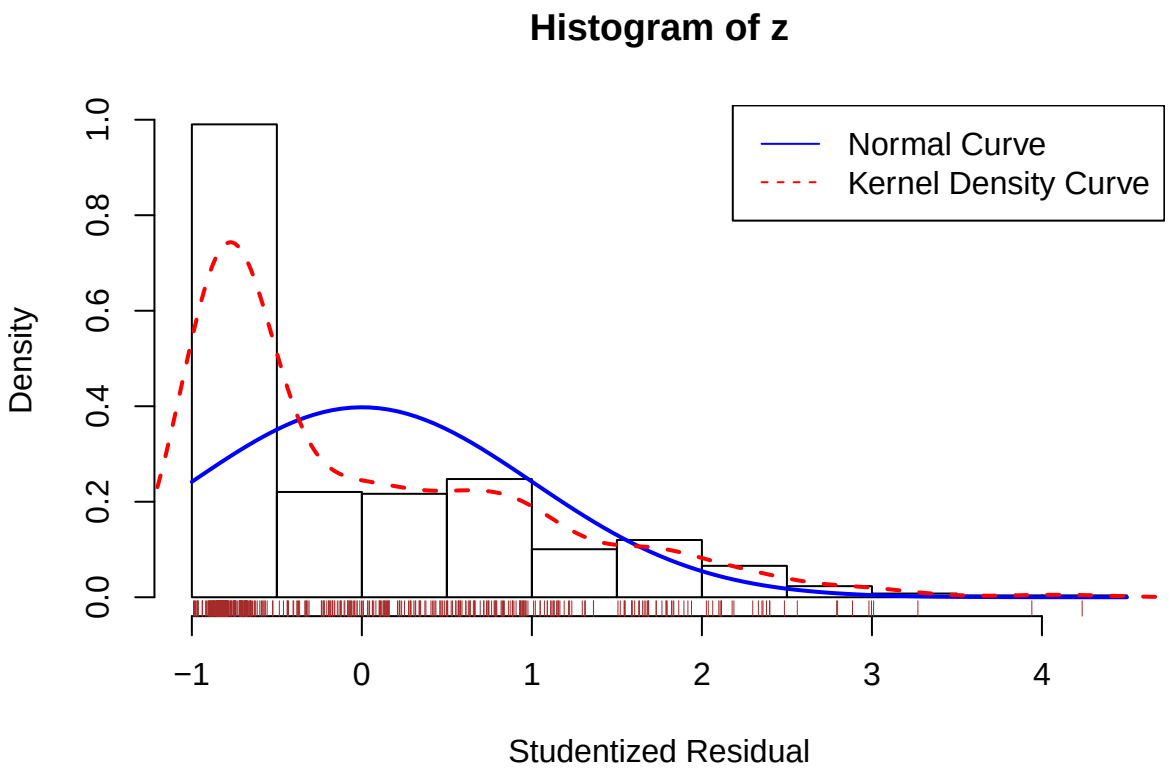
```
residplot <- function(fit, nbreaks=15){
  z <- rstudent(fit)
  hist(z, breaks = nbreaks, freq = FALSE, xlab = "Studentized Residual")
  rug(jitter(z),col="brown")
  curve(dnorm(x,mean=mean(z), sd=sd(z)), add = TRUE, col = "blue",lwd=2)
  lines(density(z)$x,density(z)$y, col="red",lwd=2,lty=2)
  legend("topright",legend = c("Normal Curve","Kernel Density Curve"),lty = 1:2,
         col = c("blue","red"))
}
residplot(fit_ff1)
```



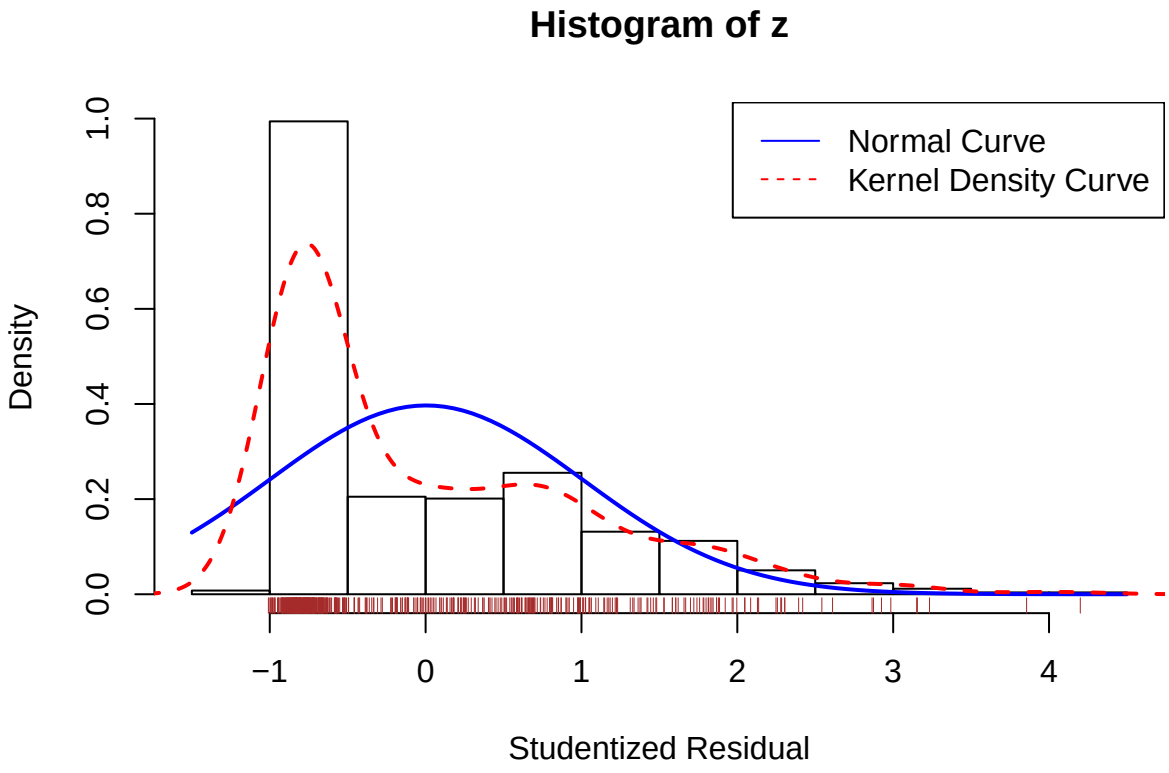

```
residplot(fit_ff2)
```




```
residplot(fit_ff3)
```




```
residplot(fit_ff4)
```




```
durbinWatsonTest(fit_ff1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5322293     0.9335475       0
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(fit_ff2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5285384     0.9401882       0
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(fit_ff3)
```

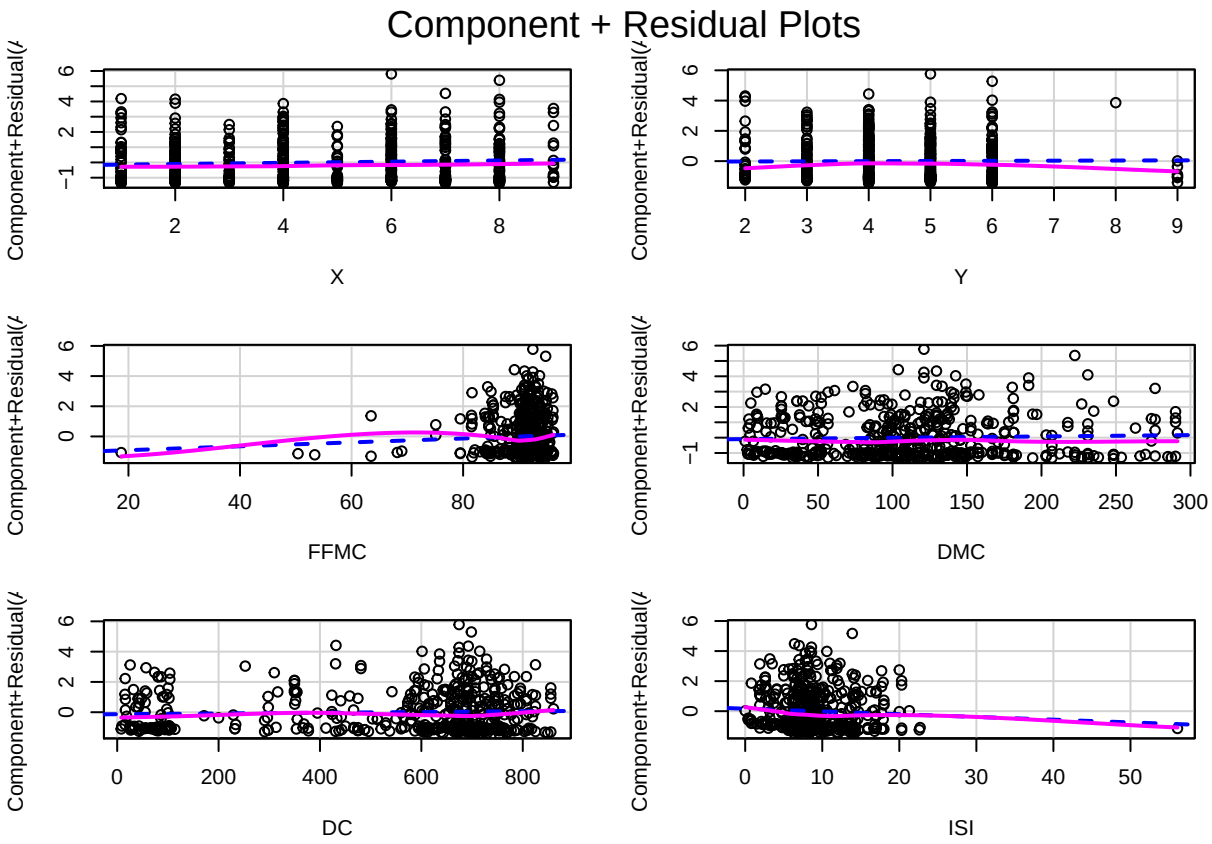
```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5397034       0.91882       0
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(fit_ff4)
```

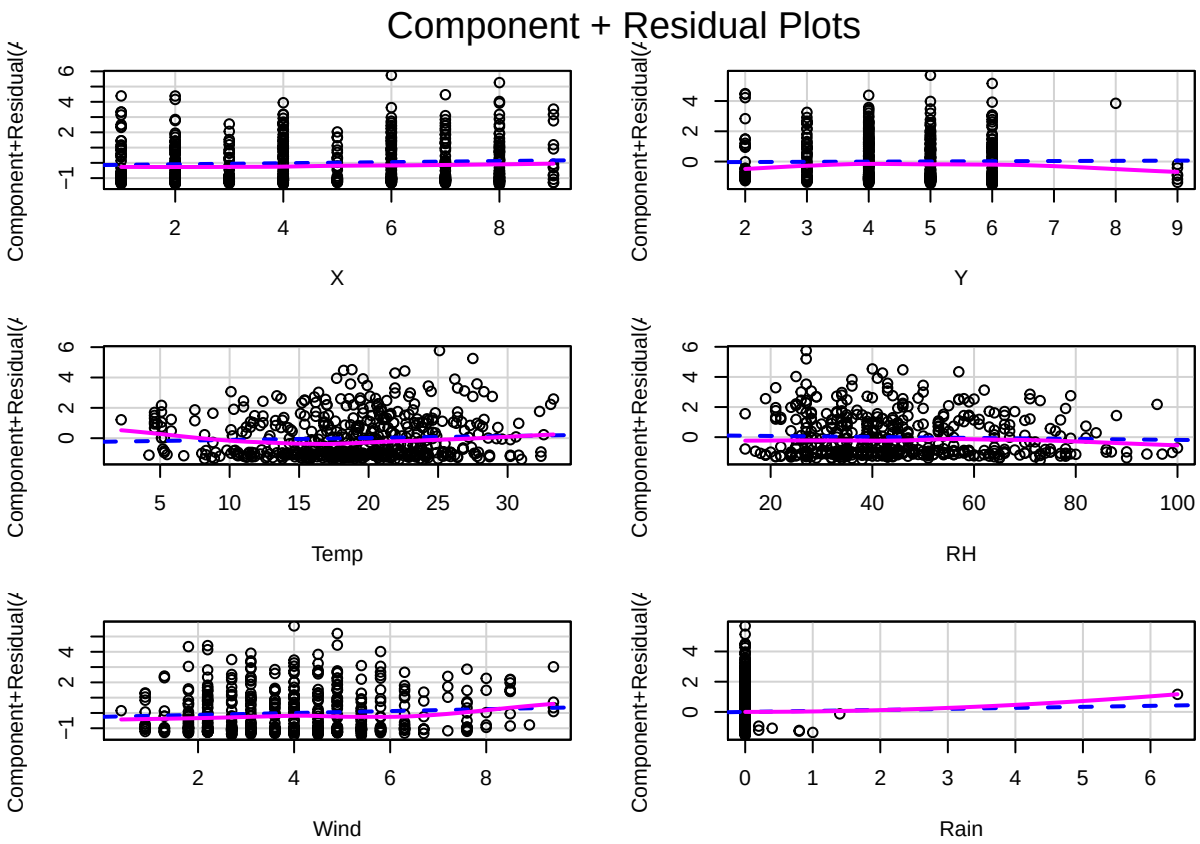
```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5365163     0.9245255       0
##  Alternative hypothesis: rho != 0
```

Interpretation for Durbin Watson Test : It checks for autocorrelatd errors. p- value > 0.05 suggests lack of autocorrelation and independence of error which is not the case for all four models.

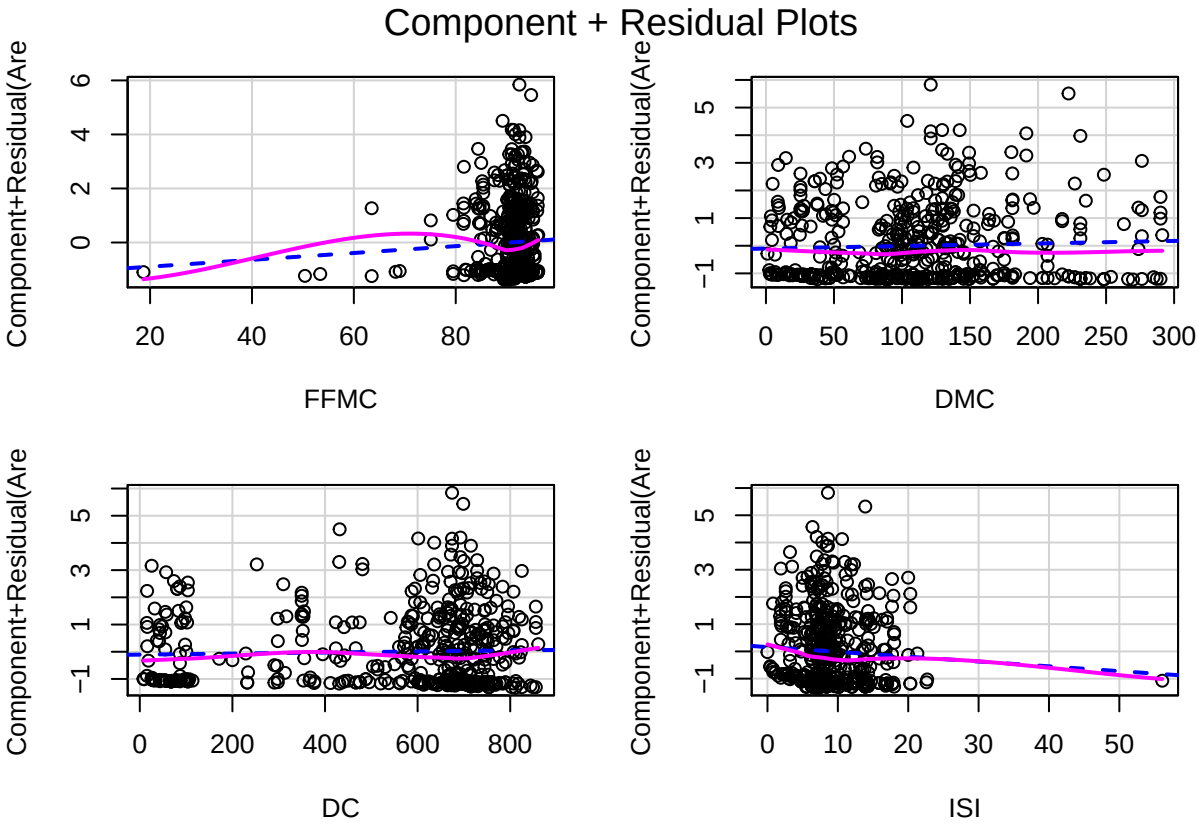
```
# Enhanced Approach for Model 1 - crPlots
crPlots(fit_ff1)
```



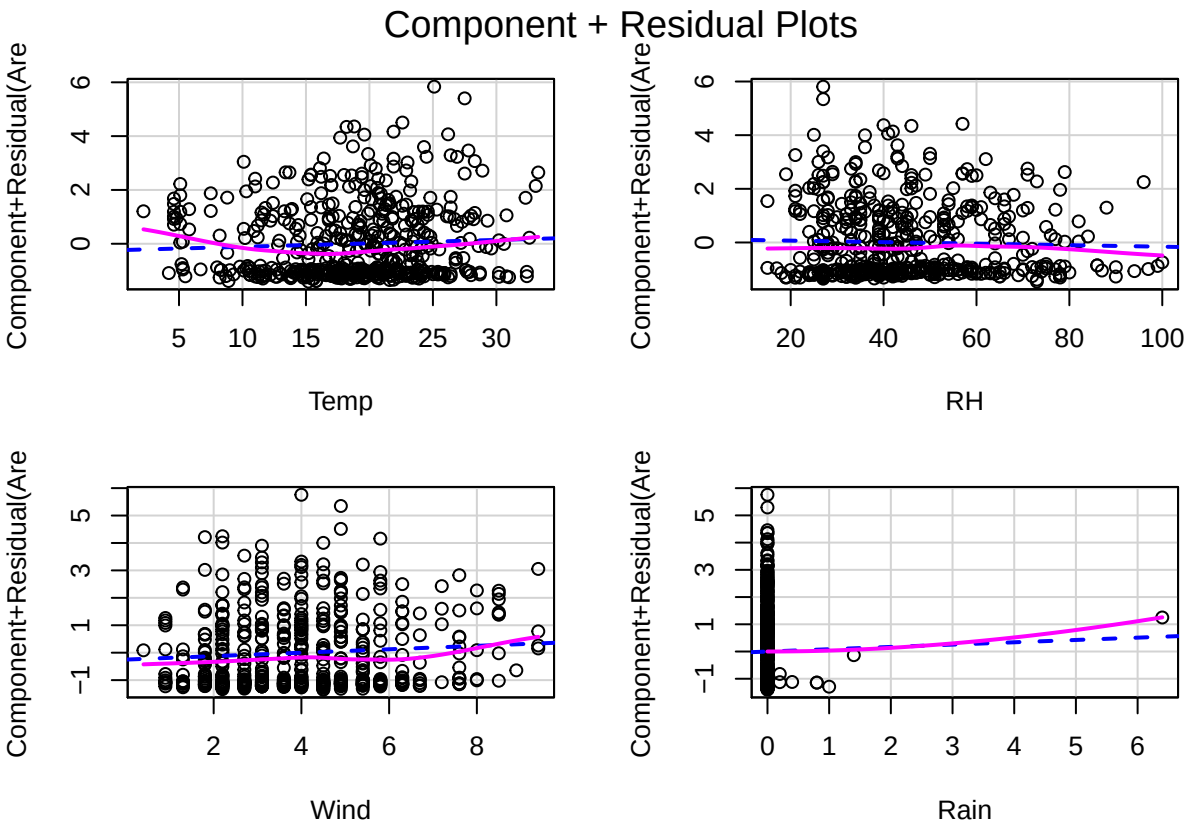

```
# Enhanced Approach for Model 2 - crPlots
crPlots(fit_ff2)
```




```
# Enhanced Approach for Model 3 - crPlots
crPlots(fit_ff3)
```




```
# Enhanced Approach for Model 4 - crPlots
crPlots(fit_ff4)
```

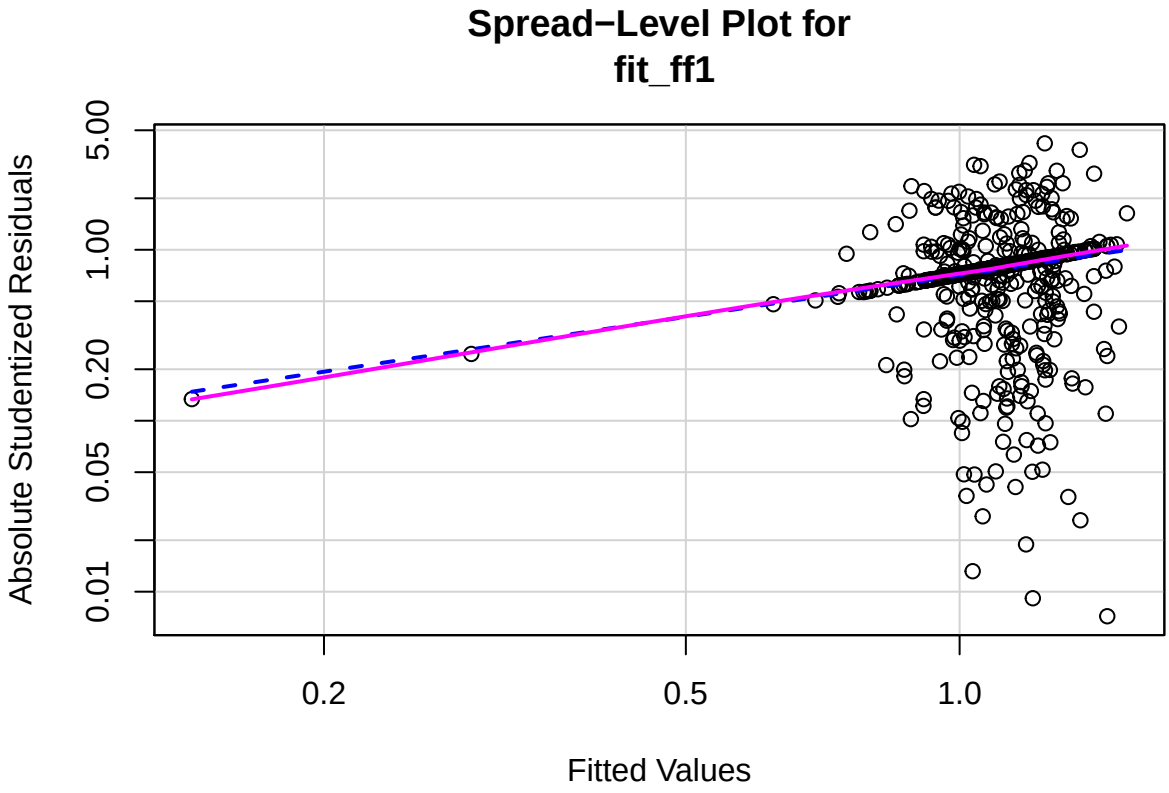


Interpretation for crPlots() : crPlots() looks for evidence of non-linearity between dependent variable and independent variable. The above plots confirm the linearity assumption between the variables.

```
ncvTest(fit_ff1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 8.953326, Df = 1, p = 0.0027696
```

```
spreadLevelPlot(fit_ff1)
```

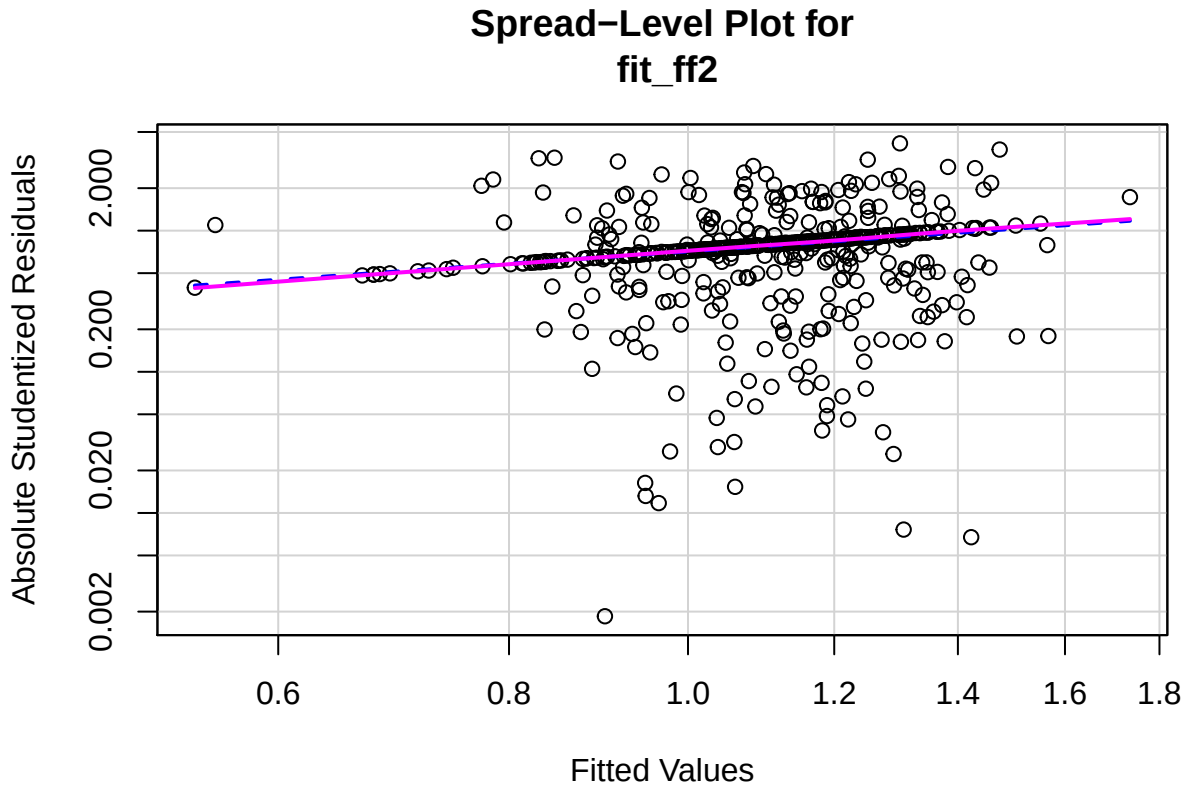



```
##
## Suggested power transformation:  0.1934578
```

```
ncvTest(fit_ff2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 7.2886, Df = 1, p = 0.0069394
```

```
spreadLevelPlot(fit_ff2)
```

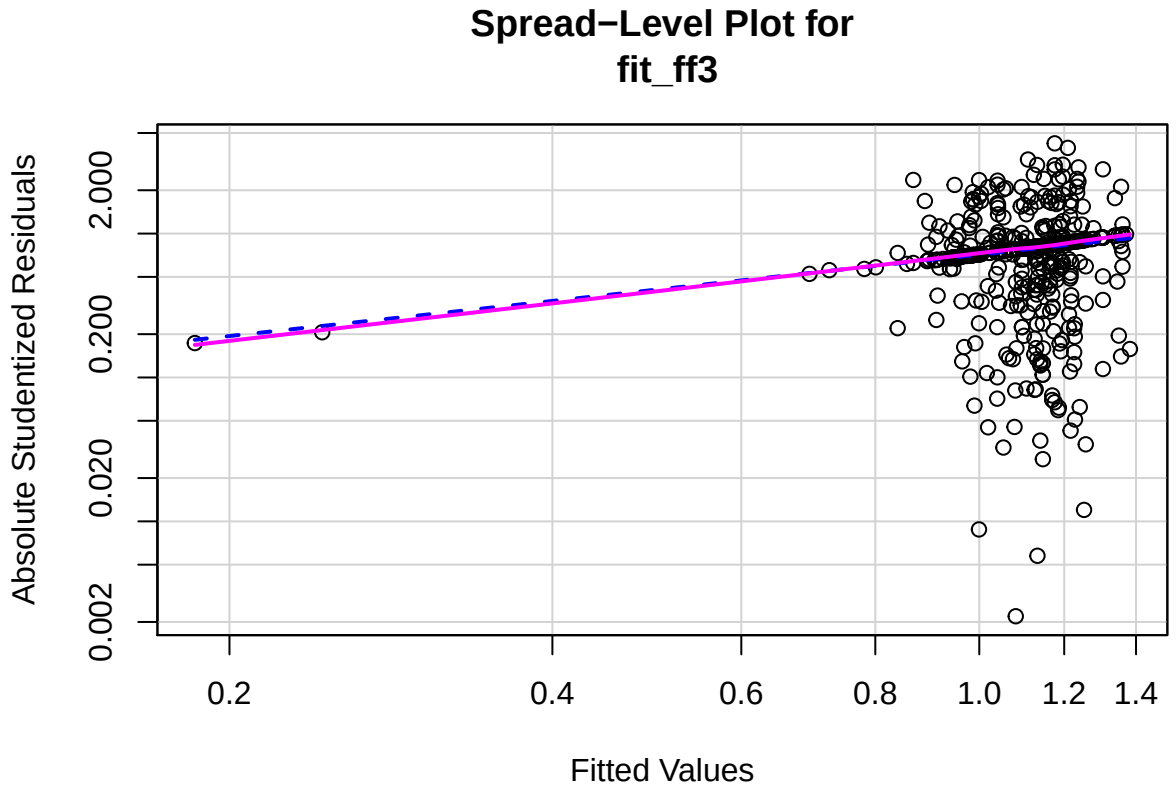



```
##
## Suggested power transformation:  0.0934361
```

```
ncvTest(fit_ff3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 5.812039, Df = 1, p = 0.015917
```

```
spreadLevelPlot(fit_ff3)
```

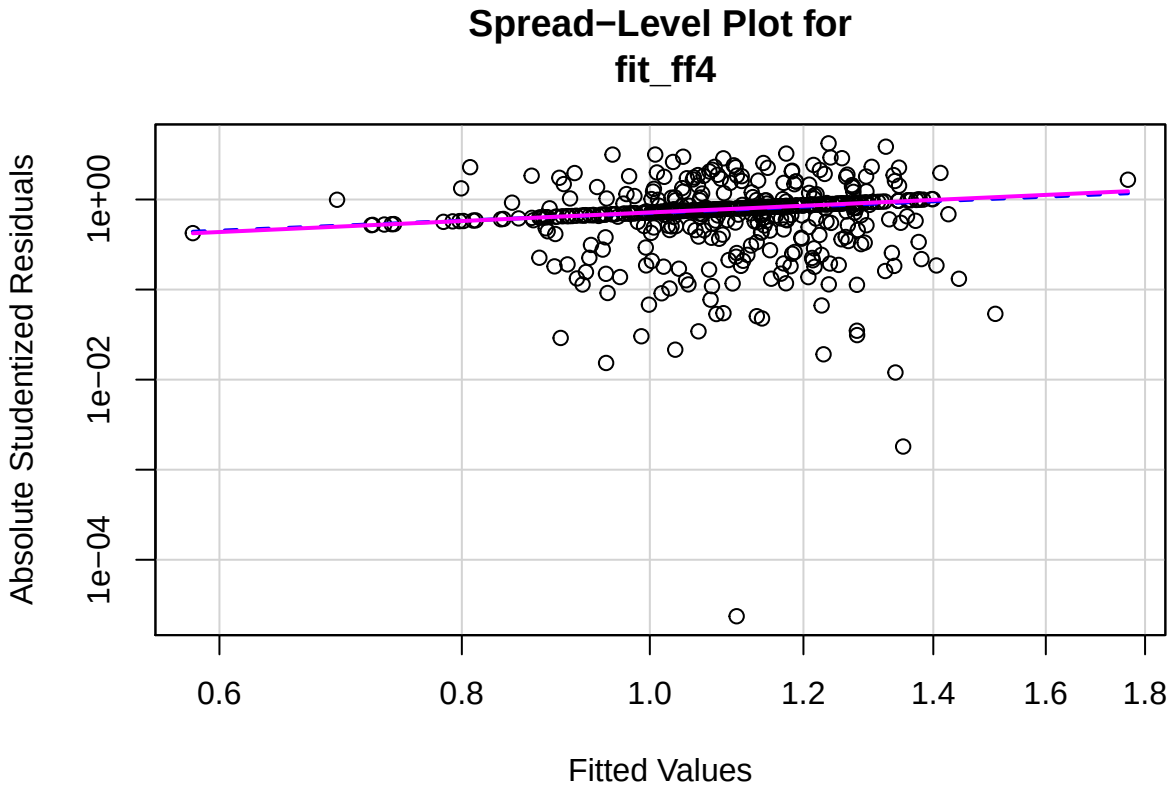



```
##
## Suggested power transformation:  0.1904659
```

```
ncvTest(fit_ff4)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 5.59294, Df = 1, p = 0.018033
```

```
spreadLevelPlot(fit_ff4)
```




```
##
## Suggested power transformation:  0.113547
```

Assessing Homoscedasticity : The ncvTest() produces two useful functions for identifying non-constant error variance against the alternative that the error varaince changes with the level of the fitted variables. The p-value is significantly low for all the 4 models suggesting that the constant variance assumption is not met.

```
# Testing for Outliers - Model 1
outlierTest(fit_ff1)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 239 4.192472         3.2542e-05     0.016824
```

```
# Testing for Outliers - Model 2
outlierTest(fit_ff2)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 239  4.15451         3.8233e-05     0.019766
```

```
# Testing for Outliers - Model 3
outlierTest(fit_ff3)
```

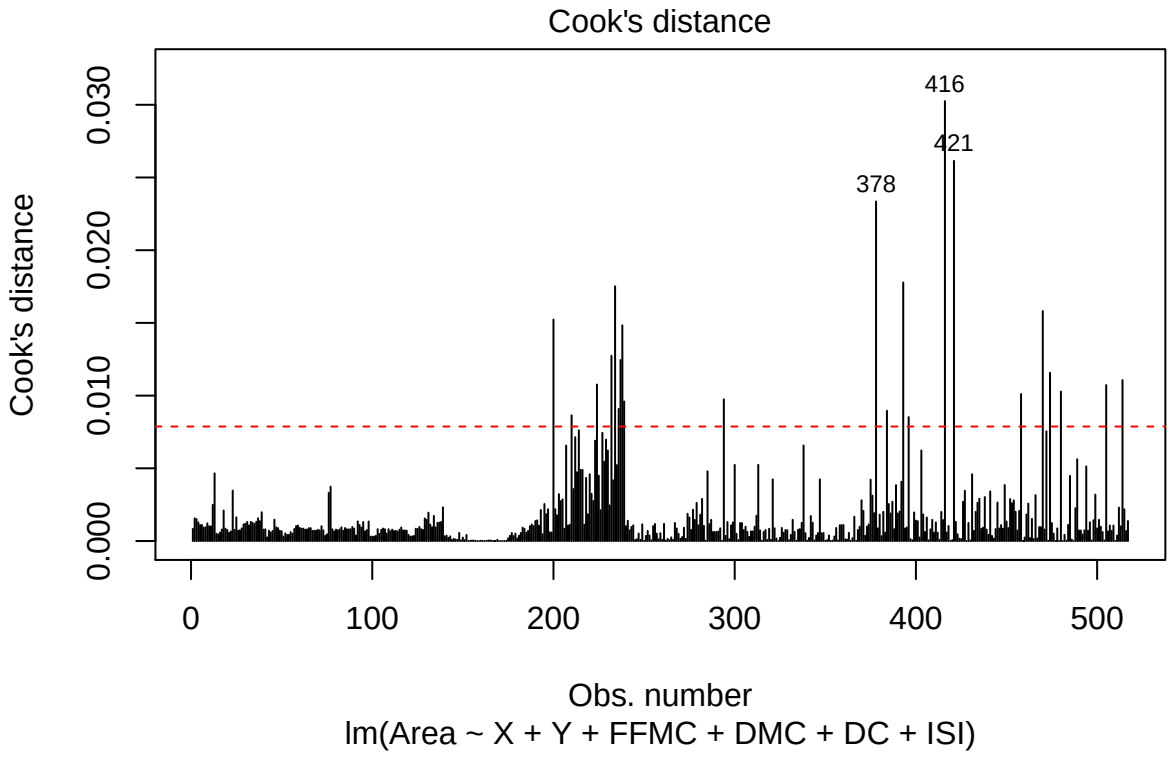
```
##     rstudent unadjusted p-value Bonferroni p
## 239 4.236704         2.6911e-05     0.013913
## 416 3.940781         9.2520e-05     0.047833
```

```
# Testing for Outliers - Model 4
outlierTest(fit_ff4)
```

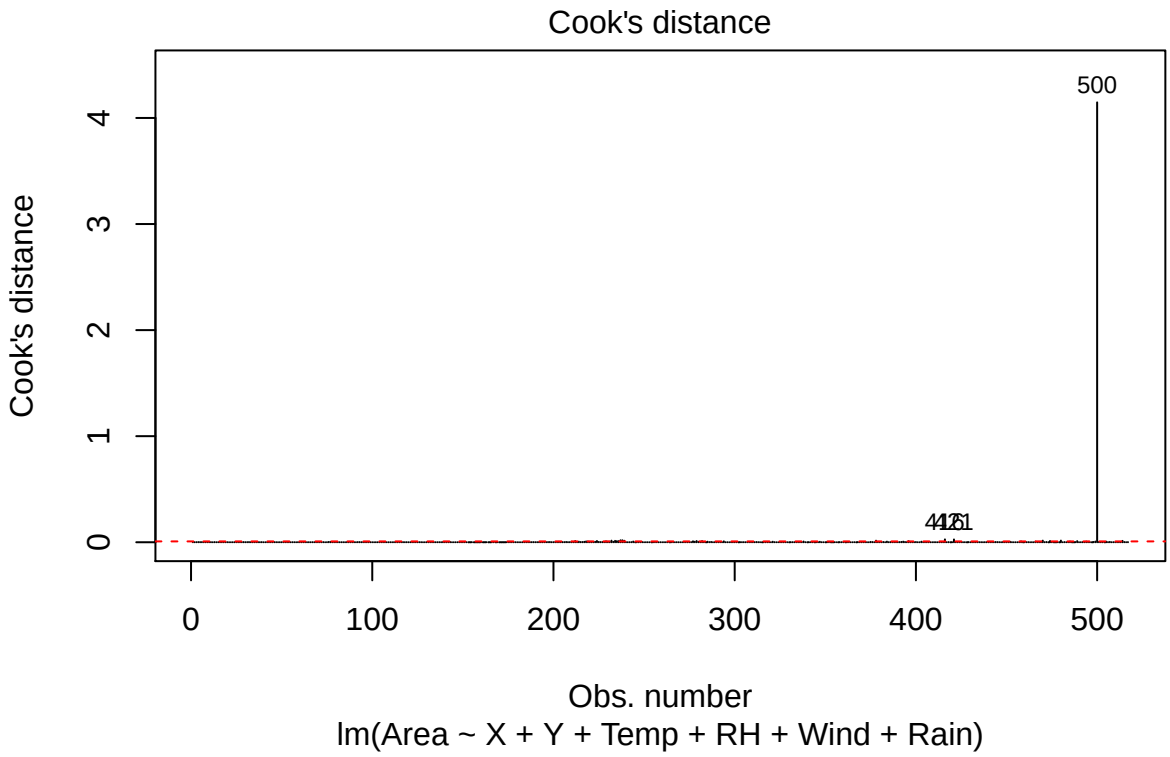
```
##     rstudent unadjusted p-value Bonferroni p
## 239  4.20067         3.1404e-05     0.016236
```

Interpretation for Outlier Test : Outliers are the observations that are not predicted well by the model.On performing the outlierTest we observe that 239 is a sgnificant outlier in all 4 models.

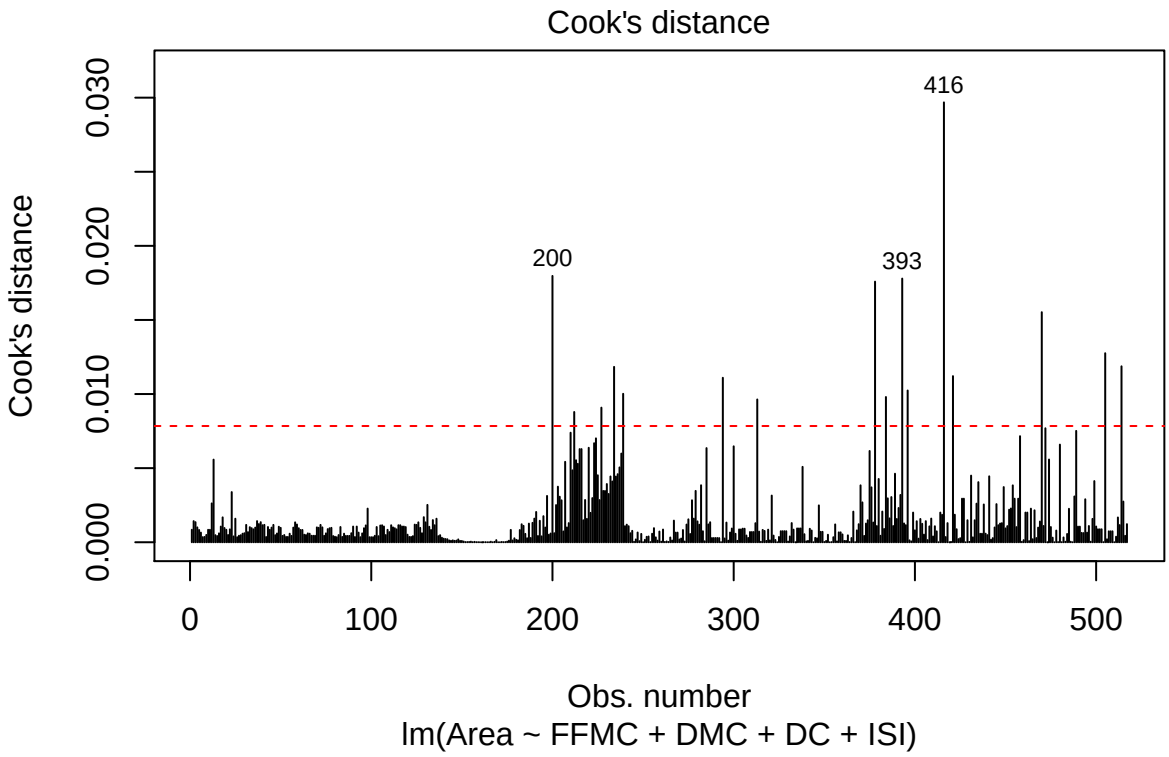
```
# Influential observations
cutoff <- 4/(nrow(forestFire_df)-length(fit_ff1$coefficients)-2)
plot(fit_ff1, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```



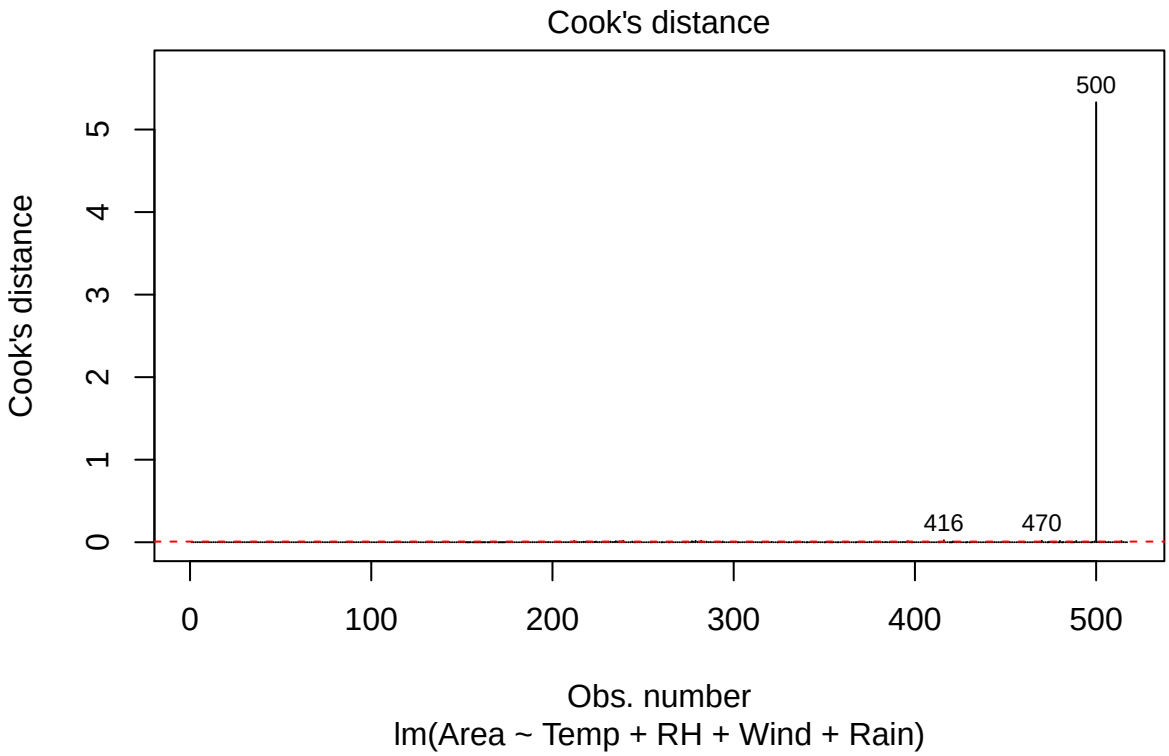

```
# Influential observations - Model2
cutoff <- 4/(nrow(forestFire_df)-length(fit_ff2$coefficients)-2)
plot(fit_ff2, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```




```r
# Influential observations - Model3
cutoff <- 4/(nrow(forestFire_df)-length(fit_ff3$coefficients)-2)
plot(fit_ff3, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```

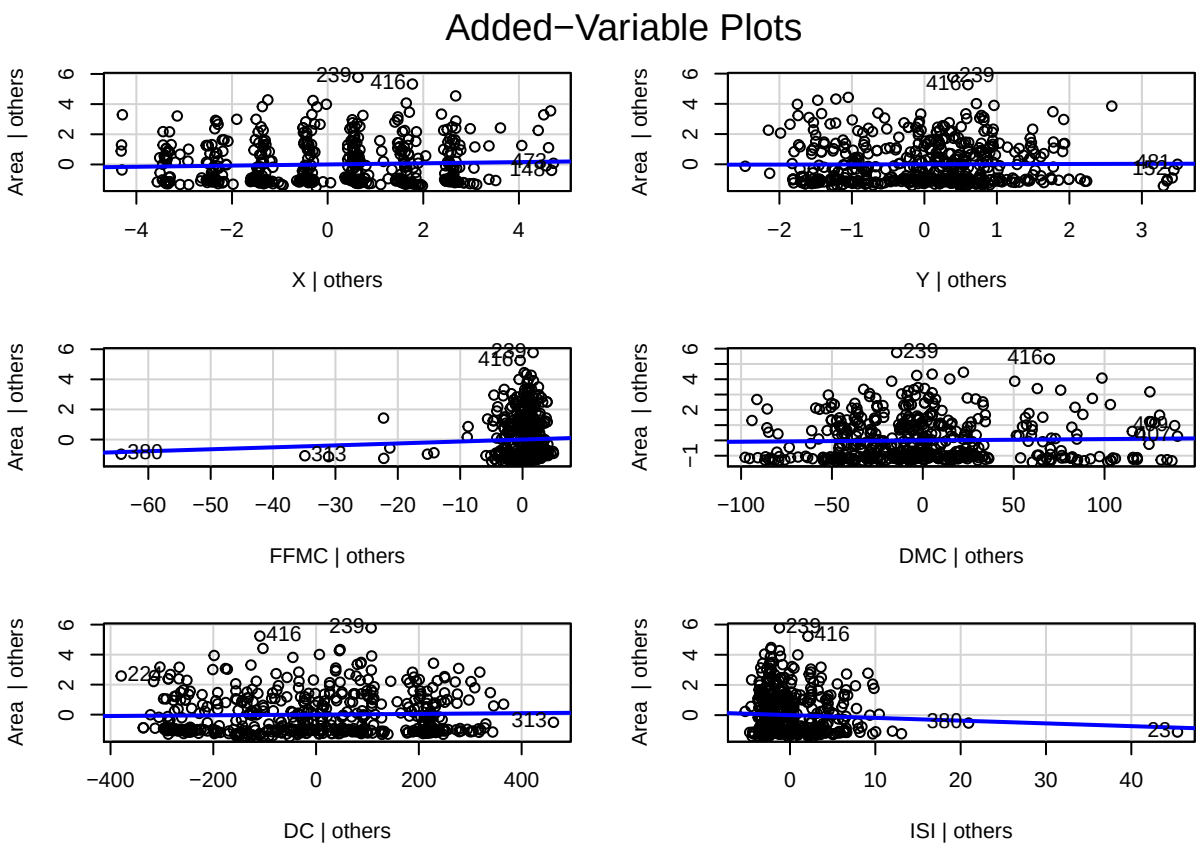



```r
# Influential observations - Model2
cutoff <- 4/(nrow(forestFire_df)-length(fit_ff4$coefficients)-2)
plot(fit_ff4, which = 4, cook.levels = cutoff)
abline(h=cutoff, lty=2, col="red")
```

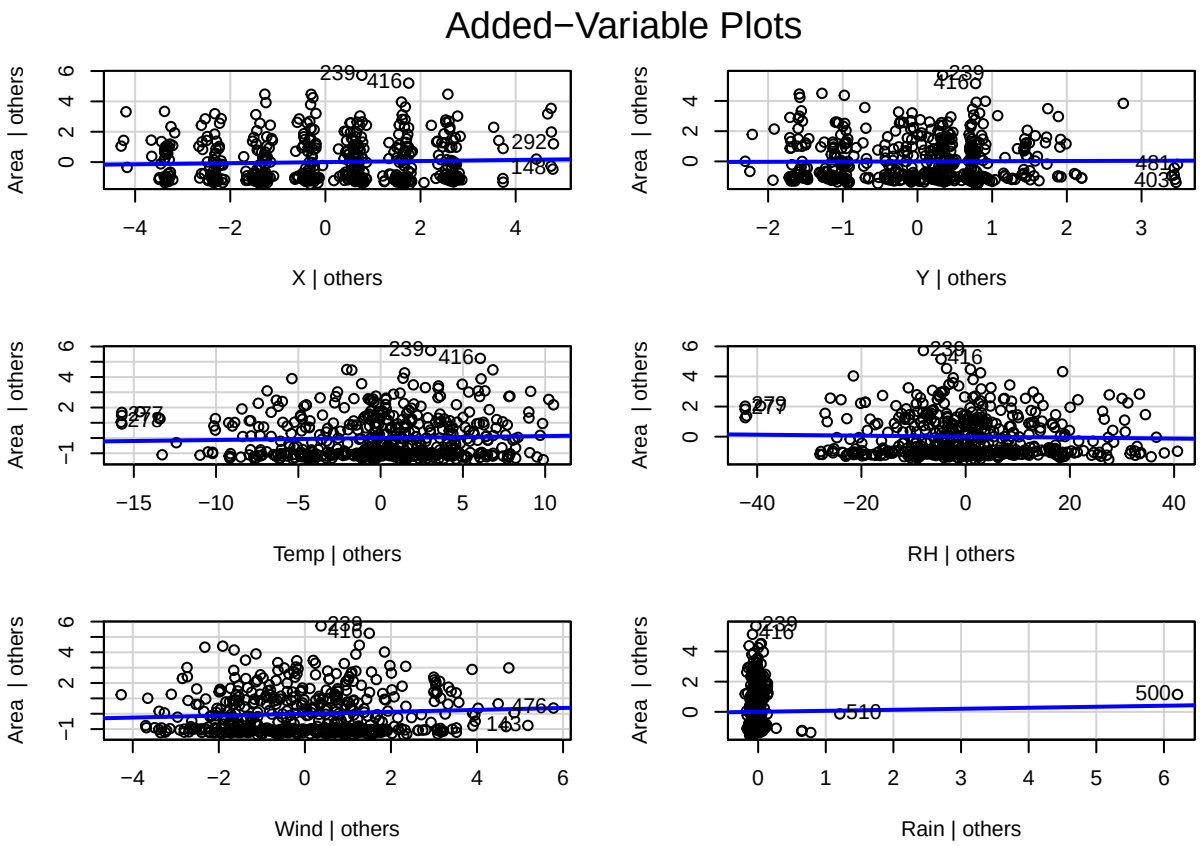


Interpretation for Influential Observation : Influential observations are observations that have a disproportionate impact on the model parameters. For the first model 416 and 421 are the influential observations. In the second and fourth model 500 is the observational influencer and in the third model 416.

```
# Added Variable Plot Model 1
avPlots(fit_ff1, ask=FALSE)
```

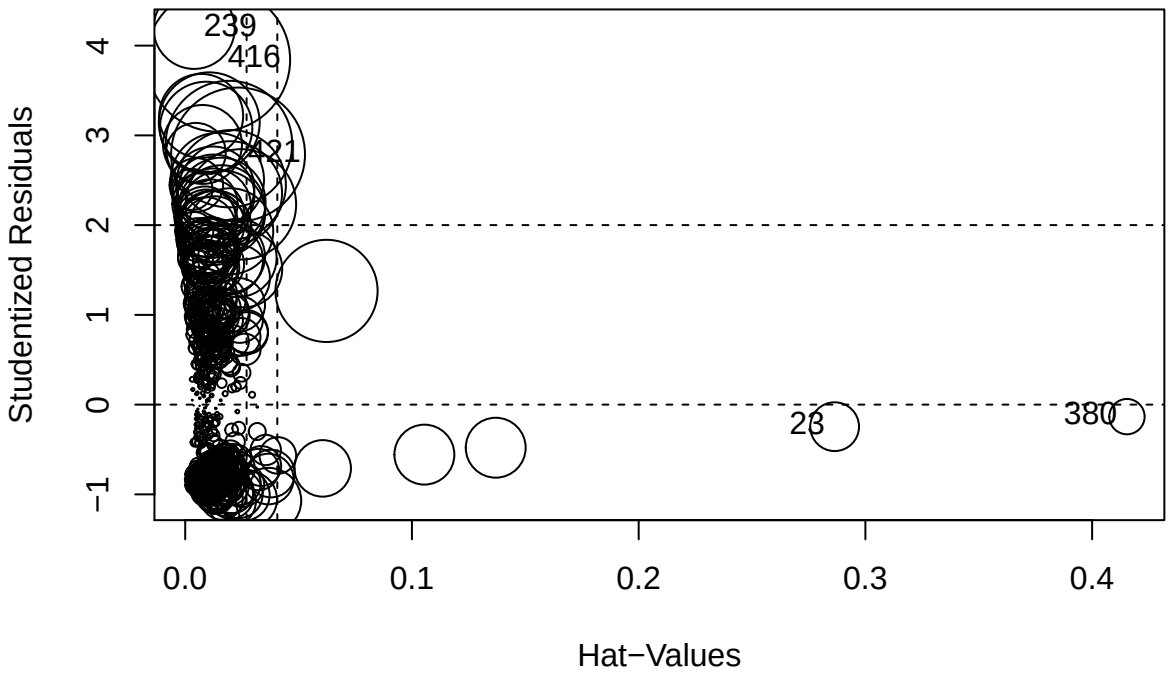



```
# Added Variable Plot Model 1
avPlots(fit_ff2, ask=FALSE)
```



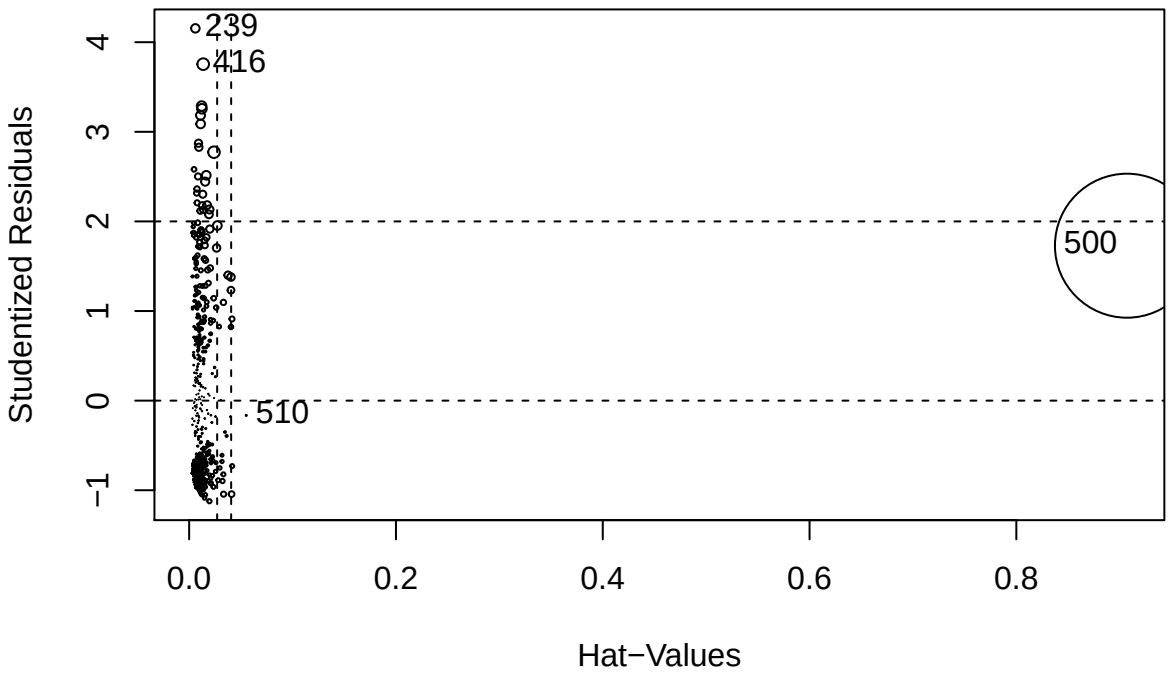
Interpretation of avPlots : avPlots not only help identify influential observations but also the affect of these observations on the model.

```
# Influence plot for model 1
influencePlot(fit_ff1)
```




```
##        StudRes         Hat       CookD
## 23  -0.2457834 0.286431834 0.003470514
## 239  4.1924718 0.003932418 0.009601113
## 380 -0.1338113 0.415362771 0.001820815
## 416  3.8426887 0.014514033 0.030251205
## 421  2.7870279 0.023315020 0.026142121
```

```
influencePlot(fit_ff2)
```




```
##         StudRes         Hat        CookD
## 239  4.1545105 0.005991275 0.0144025904
## 416  3.7550422 0.013568457 0.0270134648
## 500  1.7289317 0.906975656 4.1473055248
## 510 -0.1645355 0.055260763 0.0002266499
```

Interpreatation Influence Plots : Model 1 - 1) Outliers : 23 and 380 2) High leverage values : 23 and 380 3) Influentail Obseravtaion : 416

Model 1 - 1) Outliers : 239 and 416 2) High leverage values : 500 and 239 3) Influentail Obseravtaion : 500

```
# Adding and Deleting Variable
sqrt(vif(fit_ff1))>2
```

```
##     X     Y  FFMC   DMC    DC   ISI
## FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sqrt(vif(fit_ff2))>2
```

```
##     X     Y  Temp    RH  Wind  Rain
## FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sqrt(vif(fit_ff3))>2
```

```
##  FFMC   DMC    DC   ISI
## FALSE FALSE FALSE FALSE
```

```
sqrt(vif(fit_ff4))>2
```

```
##  Temp    RH  Wind  Rain
## FALSE FALSE FALSE FALSE
```

Deleting a variable for all the models is unneccessary.

Best Regression Model

```
# Comparing models using anova
anova(fit_ff1,fit_ff2,fit_ff3,fit_ff4,fit_ff5)
```

```
## Analysis of Variance Table
##
## Model 1: Area ~ X + Y + FFMC + DMC + DC + ISI
## Model 2: Area ~ X + Y + Temp + RH + Wind + Rain
## Model 3: Area ~ FFMC + DMC + DC + ISI
## Model 4: Area ~ Temp + RH + Wind + Rain
## Model 5: Area ~ FFMC + DMC + DC + ISI + Temp + RH + Wind + Rain
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    510  996.17
## 2    510  994.22  0    1.9474
## 3    512 1000.98 -2   -6.7647 1.7373 0.1770
## 4    512  998.61  0    2.3777
## 5    508  989.04  4    9.5657 1.2283 0.2978
```

```
AIC(fit_ff1,fit_ff2,fit_ff3,fit_ff4,fit_ff5)
```

```
##         df      AIC
## fit_ff1  8 1822.267
## fit_ff2  8 1821.256
## fit_ff3  6 1820.762
## fit_ff4  6 1819.532
## fit_ff5 10 1822.556
```

###Interpretation for best fit model : As per anova, Model 5 has the least RSS value,hence is the best fit model. But as per AIC Model 4 is the best fit model.

Fine tune the variable selection

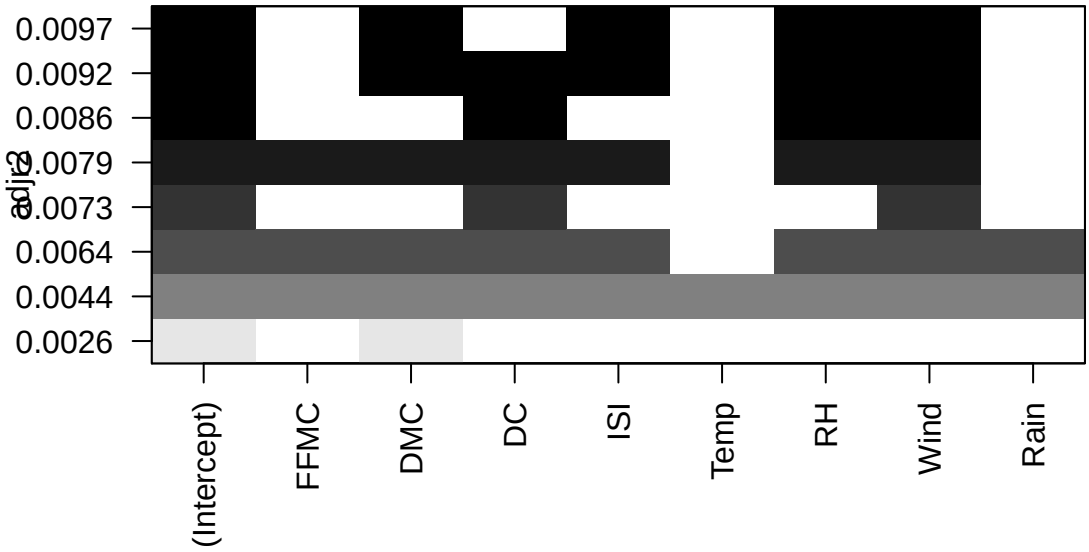
```r
library(MASS)
stepAIC(fit_ff5,direction = "backward")
```

```
## Start:  AIC=353.37
## Area ~ FFMC + DMC + DC + ISI + Temp + RH + Wind + Rain
##
##        Df Sum of Sq    RSS    AIC
## - Temp  1    0.0396 989.08 351.39
## - Rain  1    0.4029 989.44 351.58
## - FFMC  1    0.5511 989.59 351.66
## - DC    1    1.1446 990.18 351.97
## - DMC   1    1.2892 990.33 352.05
## - RH    1    1.9350 990.97 352.38
## <none>               989.04 353.37
## - ISI   1    3.8984 992.94 353.41
## - Wind  1    8.3364 997.38 355.71
##
## Step:  AIC=351.39
## Area ~ FFMC + DMC + DC + ISI + RH + Wind + Rain
##
##        Df Sum of Sq    RSS    AIC
## - Rain  1    0.4482 989.53 349.63
## - FFMC  1    0.5419 989.62 349.68
## - DC    1    1.3140 990.39 350.08
## - DMC   1    1.5889 990.67 350.22
## - RH    1    3.6797 992.76 351.31
## <none>               989.08 351.39
## - ISI   1    3.9909 993.07 351.48
## - Wind  1    8.5137 997.59 353.83
##
## Step:  AIC=349.63
## Area ~ FFMC + DMC + DC + ISI + RH + Wind
##
##        Df Sum of Sq    RSS    AIC
## - FFMC  1    0.5910 990.12 347.94
## - DC    1    1.3088 990.84 348.31
## - DMC   1    1.6400 991.17 348.48
## - RH    1    3.4488 992.98 349.43
## <none>               989.53 349.63
## - ISI   1    3.9203 993.45 349.67
## - Wind  1    8.7479 998.28 352.18
##
## Step:  AIC=347.94
## Area ~ DMC + DC + ISI + RH + Wind
##
##        Df Sum of Sq    RSS    AIC
## - DC    1    1.4240 991.54 346.68
## - DMC   1    2.1585 992.28 347.06
## - ISI   1    3.3459 993.46 347.68
## <none>               990.12 347.94
## - RH    1    4.8684 994.99 348.47
## - Wind  1    8.6539 998.77 350.44
##
```

```
## Step:  AIC=346.68
## Area ~ DMC + ISI + RH + Wind
##
##        Df Sum of Sq     RSS    AIC
## - ISI   1    3.1805  994.72 346.34
## <none>               991.54 346.68
## - RH    1    5.4795  997.02 347.53
## - Wind  1    7.6958  999.24 348.68
## - DMC   1    8.8975 1000.44 349.30
##
## Step:  AIC=346.34
## Area ~ DMC + RH + Wind
##
##        Df Sum of Sq     RSS    AIC
## <none>               994.72 346.34
## - RH    1    4.2333  998.96 346.53
## - Wind  1    6.3420 1001.06 347.62
## - DMC   1    6.4049 1001.13 347.65


##
## Call:
## lm(formula = Area ~ DMC + RH + Wind, data = ff_df)
##
## Coefficients:
## (Intercept)          DMC           RH         Wind
##    0.912936     0.001755    -0.005583     0.062413
```

```
# All subset regression
library(leaps)
leaps <- regsubsets(Area ~ FFMC + DMC + DC + ISI + Temp + RH + Wind + Rain
,data = ff_df)
plot(leaps, scale = "adjr2")
```



From the above results we can conclude that the below model is the best model - Area ~ DMC + RH + Wind