IE7275 HW2 Group 5

Group 5

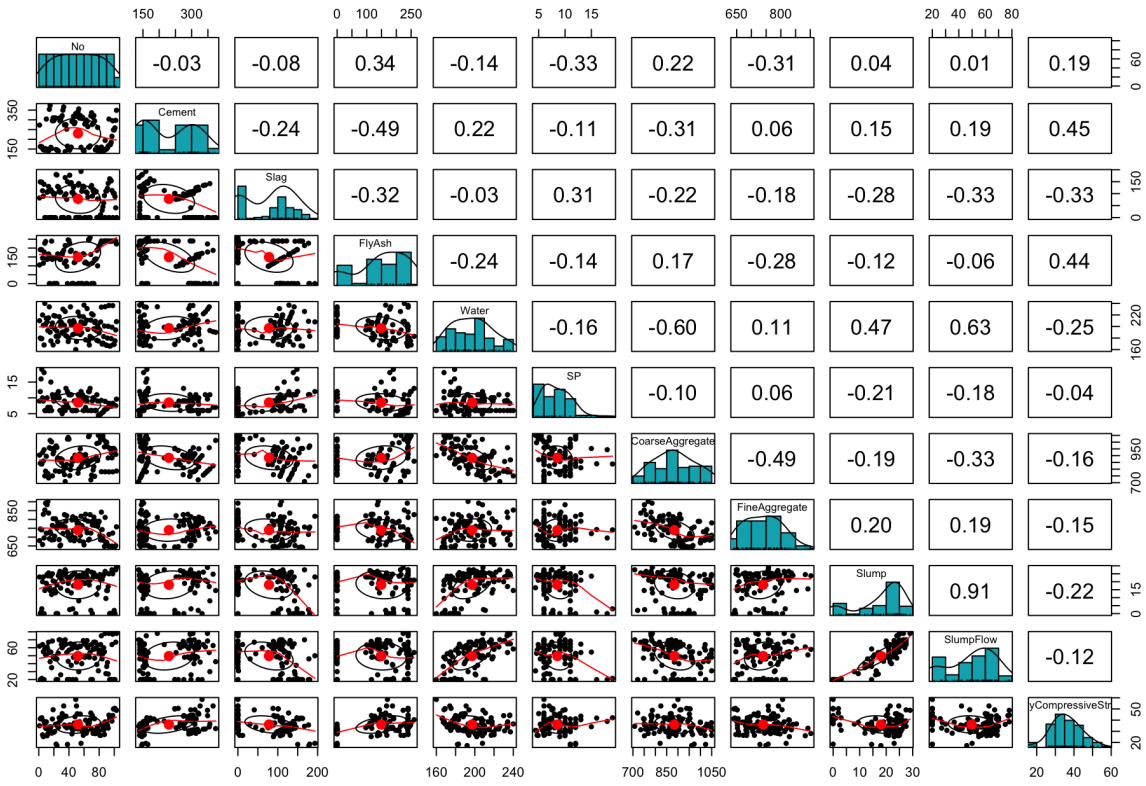
3/14/2020

Problem 1: Concrete Slump Test Data

Create a scatterplot matrix of "Concrete Slump Test Data" and select an initial set of predictor variables

- Predictors Cement, Slag, Fly Ash, Water, SP, Coarse Aggregate and Fine Aggregate
- Response vars: Slump, Slump Flow and 28-day Compressive Strength

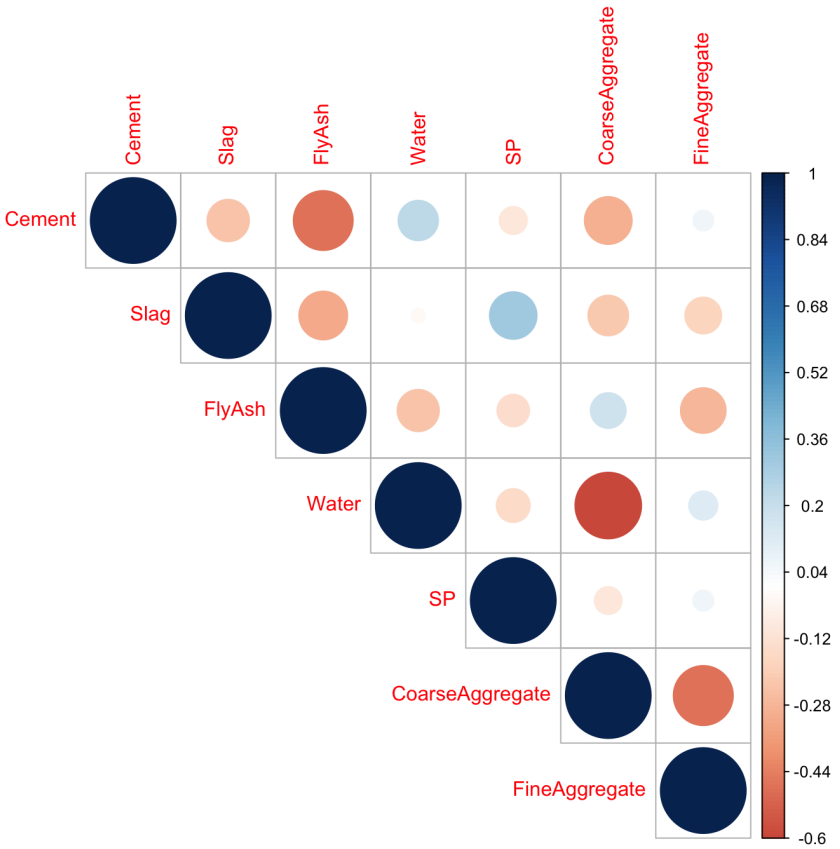
```
#create the scatterplot matrix
library(readxl)
library(forecast)
library(tidyverse)
library(caret)
library(rpart)
library(caret)
library(e1071)
library(data.table)

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

## Remove spaces from column names
Concrete<- setnames(x = Concrete_Slump_Test, old = names(Concrete_Slump_Test),
         new = gsub(" ", "", names(Concrete_Slump_Test)))

library(psych)
pairs.panels(Concrete,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



Interpretation

The correlation coefficient provides information on the effect level and direction of the linear relationship between two variables. The Pearson correlation is used when the dataset has a normal distribution. According to the correlation results of the dataset, Slump and Slump Flow are highy correlated. Slump Flow, as a response variable is also moderately correlated with the predictors water (0.63) as well as Coarse Aggregate and Slag (both with -0.33). Slump Flow will be used as response variable.

```
library(readxl)
library(corrplot)
#correlate all predictors
matrix <- as.matrix(Concrete[ , c(2:8)])
corrplot(cor(matrix), is.corr = FALSE, method = "circle", type = "upper")
```



Interpretation

The calucalted correlation coefficients (R) of all predictor variables are shown on the graph. Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients. In the right side of the correlogram, the legend color shows the correlation coefficients and the corresponding colors.The R value of approximately zero indicates that the points have no direction.

Build a few potential regression models using "Concrete Slump Test Data"

Multiple Linear Regression

```r
#ML:predictive analysis
#multiple regression model

#set the training and testing dataset p=0.7
#seprate the outcome from the predictor varaibles
# Set seed for reproducibility
set.seed(1000)
Concrete_slumpflow <- Concrete$SlumpFlow

idx <- createDataPartition(Concrete_slumpflow, p = 0.7, list = FALSE)

train_set <- Concrete[idx, ]
train_outcome <- Concrete_slumpflow[idx]

test_set <- Concrete[-idx, ]
test_outcome <- Concrete_slumpflow[-idx]


#slump flow is the response variables
#Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` + `Fine Aggregate`) are the initial s
        et of predictor variables

regressorlm <- lm(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate + FineAggregat
        e,data =train_set)
summary(regressorlm)
```

```
##
## Call:
## lm(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP +
##     CoarseAggregate + FineAggregate, data = train_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.9381  -7.9130   0.2774   9.4629  25.1517
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -512.3055   382.1232  -1.341   0.1846
## Cement             0.1483     0.1229   1.207   0.2317
## Slag               0.1183     0.1718   0.689   0.4933
## FlyAsh             0.1492     0.1245   1.198   0.2350
## Water              0.9357     0.3799   2.463   0.0163 *
## SP                -0.1520     0.7043  -0.216   0.8297
## CoarseAggregate    0.1828     0.1489   1.227   0.2240
## FineAggregate      0.2050     0.1556   1.318   0.1921
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.31 on 67 degrees of freedom
## Multiple R-squared:  0.5551, Adjusted R-squared:  0.5086
## F-statistic: 11.94 on 7 and 67 DF,  p-value: 8.897e-10
```

```r
summary(regressorlm)$coefficient
```

```
##                     Estimate  Std. Error    t value   Pr(>|t|)
## (Intercept)     -512.3055474 382.1231627 -1.3406817 0.18455190
## Cement             0.1482835   0.1228524  1.2070055 0.23167417
## Slag               0.1183142   0.1717691  0.6887979 0.49332880
## FlyAsh             0.1492146   0.1245149  1.1983669 0.23499599
## Water              0.9356546   0.3798627  2.4631388 0.01634826
## SP                -0.1520425   0.7042846 -0.2158822 0.82973544
## CoarseAggregate    0.1827820   0.1489397  1.2272211 0.22403420
## FineAggregate      0.2050319   0.1555876  1.3177907 0.19206193
```

```
#make the prediction
predicted_slump1 <- predict(regressorlm, newdata = test_set)
residuls1 <- test_set$SlumpFlow[1:24]-predicted_slump1[1:24]
d1 <- data.frame("Predicted"= predicted_slump1[1:24], "Actual"=test_set$SlumpFlow[1:24], "Residul"
        = residuls1)
#randomly samples 5 rows from the dataframe to display
library(knitr)
kable(sample_n(d1, 5))
```

Predicted	Actual	Residul
51.17699	20	-31.176987
64.60339	51	-13.603394
45.42839	20	-25.428395
61.93097	51	-10.930974
53.32081	62	8.679192

```
accuracy(predicted_slump1,test_set$SlumpFlow)
```

```
##                ME     RMSE      MAE      MPE     MAPE
## Test set -0.12679 14.98764 13.24437 -16.178 37.17616
```

Interpretation

Multiple linear regression is an extension of simple linear regression used to predict an outcome variable (y) on the basis of multiple distinct predictor variables (x). With three predictor variables (x), the prediction of y is expressed by the following equation:

Slump Flow = -512.306 + (0.148)Cement + (0.118)Slag + (0.149)FlyAsh + (0.936)Water + (-0.152)SP + (0.183)CoarseAggregate + (0.205)FineAggregate

It can be seen that p-value of the F-statistic is 0.00000000089, which is highly significant. This means that, at least, one of the predictor variables is significantly related to the response variable. This is likely the Water variable. For a given the predictor, the t-statistic evaluates whether or not there is significant association between the predictor and the outcome variable. It can be seen that, changes in Water input are significantly associated to the change in slump flow while changes in slag input are not significantly associated with slump flow.

The overall quality of the model can be assessed by examining the R-squared (R2). R2 represents the proportion of variance in the response variable, slump flow, that may be predicted by knowing the value of the predictor variables. In the above model, the adjusted R2 = 0.555, meaning that 55.5% of the variance in the measure of slump folw can be predicted by the 7 predictor inputs. In addition, the RMSE for the model is 15.

Support Vector Regression (SVR)

```
#SVM
library(e1071)
svr_regressor <- svm(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate + FineAggre
        gate,data =train_set, type = 'eps-regression')

#make the prediction
predicted_slump2 <- predict(svr_regressor, newdata = test_set)

residuls2 <- test_set$SlumpFlow[1:24]-predicted_slump2[1:24]
d2 <- data.frame("Predicted"= predicted_slump2[1:24], "Actual"=test_set$SlumpFlow[1:24], "Residul"
        = residuls2)
#randomly samples 5 rows from the dataframe to display
library(knitr)
kable(sample_n(d2, 5))
```

Predicted	Actual	Residul
67.94882	67.0	-0.9488242
48.69195	31.0	-17.6919521
51.21094	42.5	-8.7109404

Predicted	Actual	Residul
59.12370	68.5	9.3762985
45.17461	39.0	-6.1746059

```
accuracy(predicted_slump2,test_set$SlumpFlow)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set -2.375384 14.08637 10.40457 -20.07103 33.25076
```

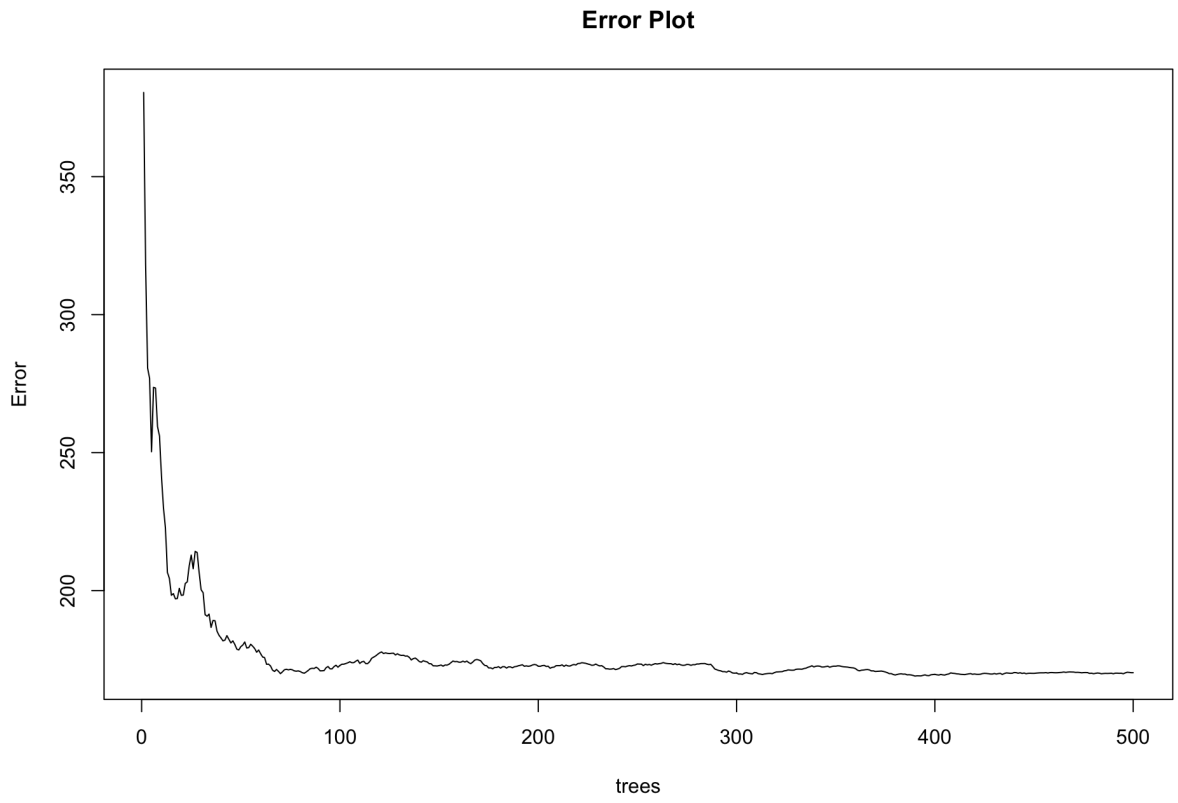
Interpretation

Support Vector Regression (SVR) uses the same basic idea as Support Vector Machine (SVM) which is a classification method that separates data using hyperplanes such that the data space is divided into segments and each segment contains only one kind of data. However, classification model will be fitted only when type of y variable is a factor. As the slump flow, a y variable in this model, is not a labled factor, SVR will be using SVM as a regression analysis predicting real values rather than a class. SVR acknowledges the presence of non-linearity in the data and provides a proficient prediction model.

From the above results, it can be observed that the RMSE for the linear model,15, is larger than the RMSE of the above SVR model, 14.1 This incicates that the implementation of SVR has an accuracy higher than the linear regression model. Therefore, we can conclude that SVR is superior to the linear model as a prediction method. This is because the linear model cannot capture the nonlinearity in a dataset and SVR is the best fit in this Concrete Slump Test Data, an evidently nonlinear dataset.

Random Forest (RF)

```
#RF
library(randomForest)
library(data.table)

regressorRF <- randomForest(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate + Fi
        neAggregate, data=train_set)

plot(regressorRF,  main="Error Plot")
```



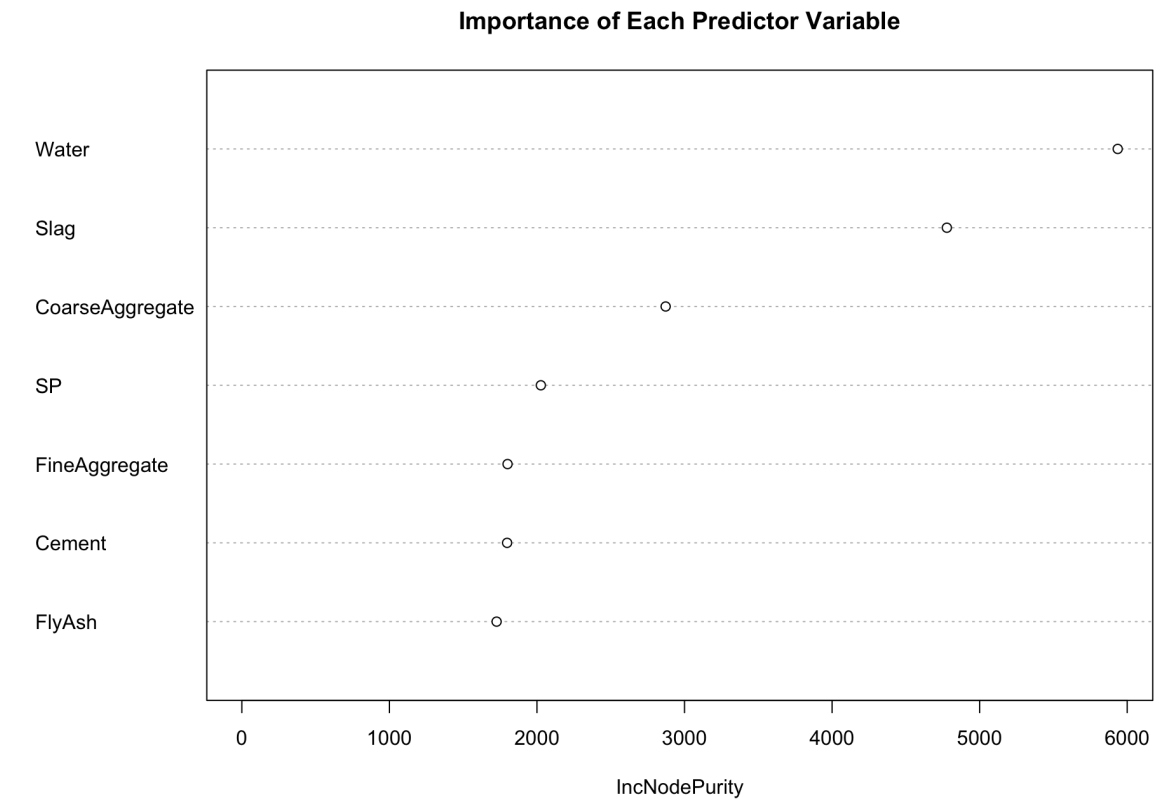

```
summary(regressorRF)
```

```
##                 Length Class  Mode
## call               3    -none- call
## type               1    -none- character
## predicted         75    -none- numeric
## mse              500    -none- numeric
## rsq              500    -none- numeric
## oob.times         75    -none- numeric
## importance         7    -none- numeric
## importanceSD       0    -none- NULL
## localImportance    0    -none- NULL
## proximity          0    -none- NULL
## ntree              1    -none- numeric
## mtry               1    -none- numeric
## forest            11    -none- list
## coefs              0    -none- NULL
## y                 75    -none- numeric
## test               0    -none- NULL
## inbag              0    -none- NULL
## terms              3    terms  call
```

```
print(regressorRF)
```

```
##
## Call:
##  randomForest(formula = SlumpFlow ~ Cement + Slag + FlyAsh + Water +      SP + CoarseAggregate
+ FineAggregate, data = train_set)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 170.2631
##                     % Var explained: 44.06
```

```
## to look at variable importance
varImpPlot(regressorRF,  main="Importance of Each Predictor Variable")
```



```
#make the prediction
predicted_slump3 <- predict(regressorRF, newdata = test_set,type="response")

residuls3 <- test_set$SlumpFlow[1:24]-predicted_slump3[1:24]
d3 <- data.frame("Predicted"= predicted_slump3[1:24], "Actual"=test_set$SlumpFlow[1:24], "Residul"
        = residuls3)
#randomly samples 5 rows from the dataframe to display
library(knitr)
kable(sample_n(d3, 5))
```

Predicted	Actual	Residul
55.38760	67.0	11.612397
51.19912	42.5	-8.699117
46.49790	31.0	-15.497897
39.73570	50.0	10.264303
54.87073	57.0	2.129273

```
accuracy(predicted_slump3,test_set$SlumpFlow)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set -1.372397 14.82353 11.62436 -20.12388 36.49712
```

Interpretation

Random forest is a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of overcoming over-fitting problem of individual decision tree. In the other word, random forest provides a way to remove the weaknesses of one decision tree by averaging the results of many. Although most widely used for classification, RF can also be used for regression model with continuous response variable. The function, randomForest was used to train a forest of B=500 trees. The forest is to predict the median value of slump flow based on the input predictor variables. The resulted variance is moderate at around 44%.

The error plot shows that the error decreases as the number of tree increases.

In the Importance of Each Predictor Variable graph, the higher the IncNodePurity of a predictor variable, the more important the variable is. As seen in the plot, water is most important while FLyAsh is the least important. It can also be observed that the RMSE for the above random forest is 14.8, which is larger than the SVR's RMSE of 14.1, but smaller than the linear model's RMSE of 15.

Select the best ML regression model

```
#comapre the Machine Learning models
postResample(pred=predicted_slump1[1:24],obs=test_set$SlumpFlow[1:24])
```

```
##       RMSE   Rsquared        MAE
## 14.8209831  0.1128988 12.8127840
```

```
postResample(pred=predicted_slump2[1:24],obs=test_set$SlumpFlow[1:24])
```

```
##       RMSE   Rsquared        MAE
## 13.6992132  0.2461696  9.5707825
```

```
postResample(pred=predicted_slump3[1:24],obs=test_set$SlumpFlow[1:24])
```

```
##       RMSE   Rsquared        MAE
## 13.9936078  0.1825403 10.4327567
```

Interpretation

In summary, SVR did the best in prediction with smaller RMSE and larger R^2 when 70% of the dataset is used in training. However, the multiple regression model did the best in prediction when 80% of the dataset is used in training. The assumption is that the more data being hold-out for later testing, the better the performance estimation. In order to avoid bias, significant amount of data is needed for testing/evaluation of the model.

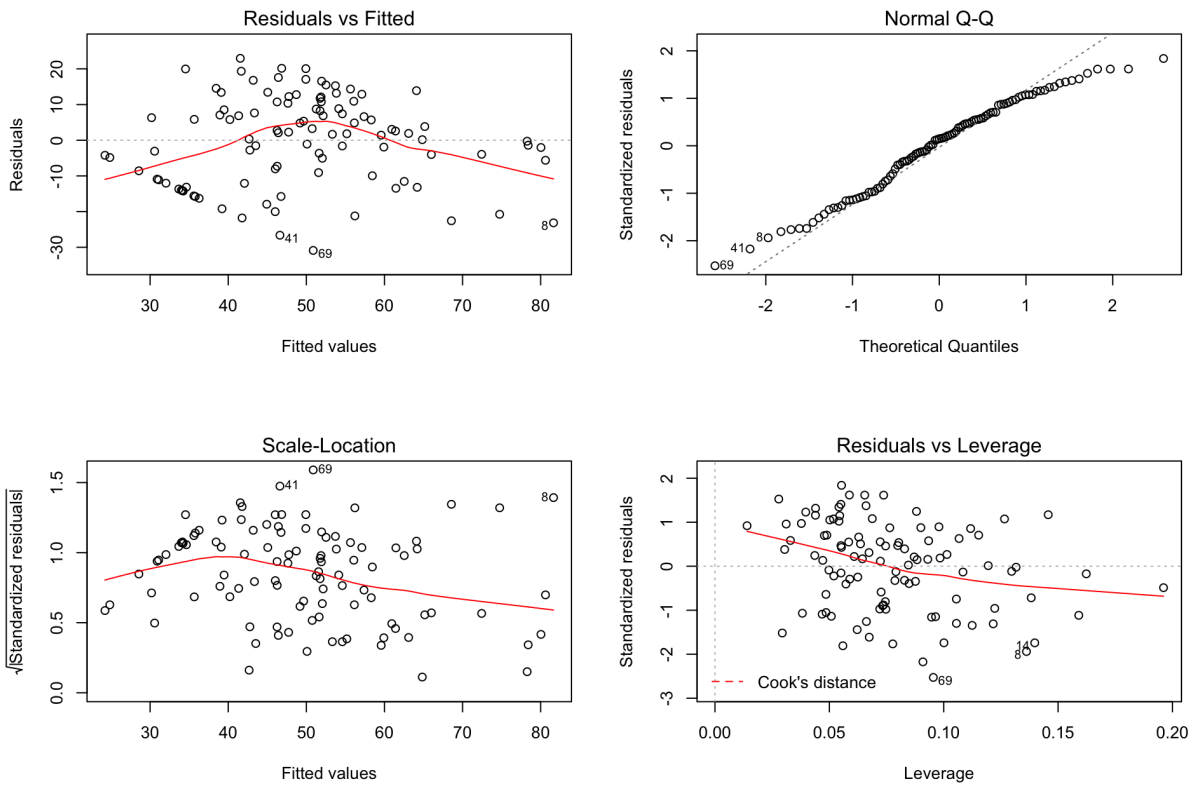
Perform regression diagnostics using both typical approach and enhanced approach

```
library("readxl")
df <- read_excel("Concrete Slump Test Data.xlsx")
```

```
fit1 <- lm(`Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` + `Fine Aggr
        egate`, data = df)
```

Regression diagnostics

```
# A typical approach
par(mfrow=c(2,2))
plot(fit1)
```



Interpretation

In the residual-fitted plot, the residuals bounce randomly around the 0 line, therefore the assumption that the relationship is linear is reasonable.

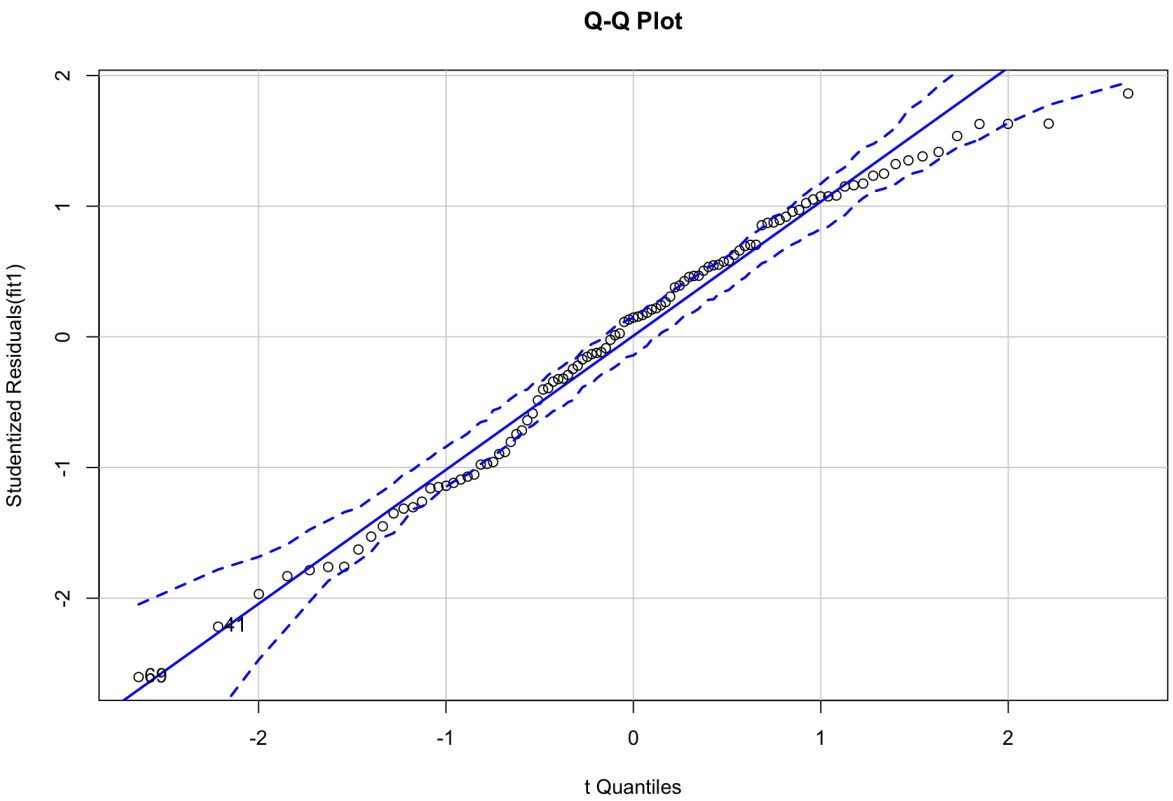
In the Normall Q-Q plot, the points are mostly on the 45-degree line depicting its normality.

In the Scale-Location plot, the points are randomly spread around a horizontal line, hence its homoscedasticity.

In the Residuals-vs-Leverage plot, it can be observe that there are three points labeled, 8, 14 and 69, they are outliers that may be removed.

An enhanced approach

```
#NORMALITY
library(car)
qqPlot(fit1, labels=row.names(df), id.method="identify", simulate=TRUE, main="Q-Q Plot")
```




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

Interpretation

All the points fall close to the line and are within the confidence envelope suggesting that the model meets the normality assumption moderately well.

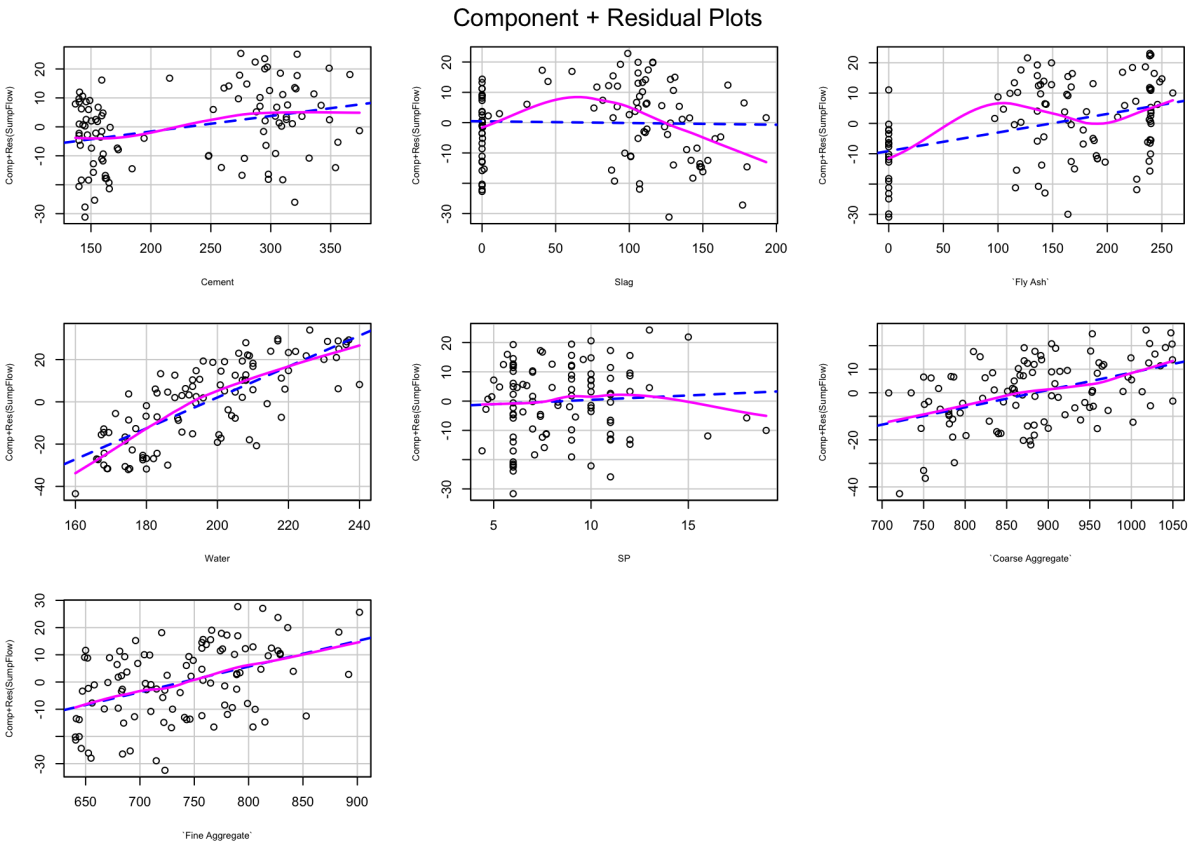
```
#INDEPENDENCE OF ERRORS
durbinWatsonTest(fit1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.01249995      2.009189   0.794
##  Alternative hypothesis: rho != 0
```

Interpretation

The nonsignificant p-value shows a lack of autocorrelation, and conversely an independence of errors. The lag value (lag=1) means each observation is being compared against the one next to it in the dataset.

```
#LINEARITY
library(car)
crPlots(fit1,ylab="Comp+Res(SumpFlow)", cex.lab=0.7)
```



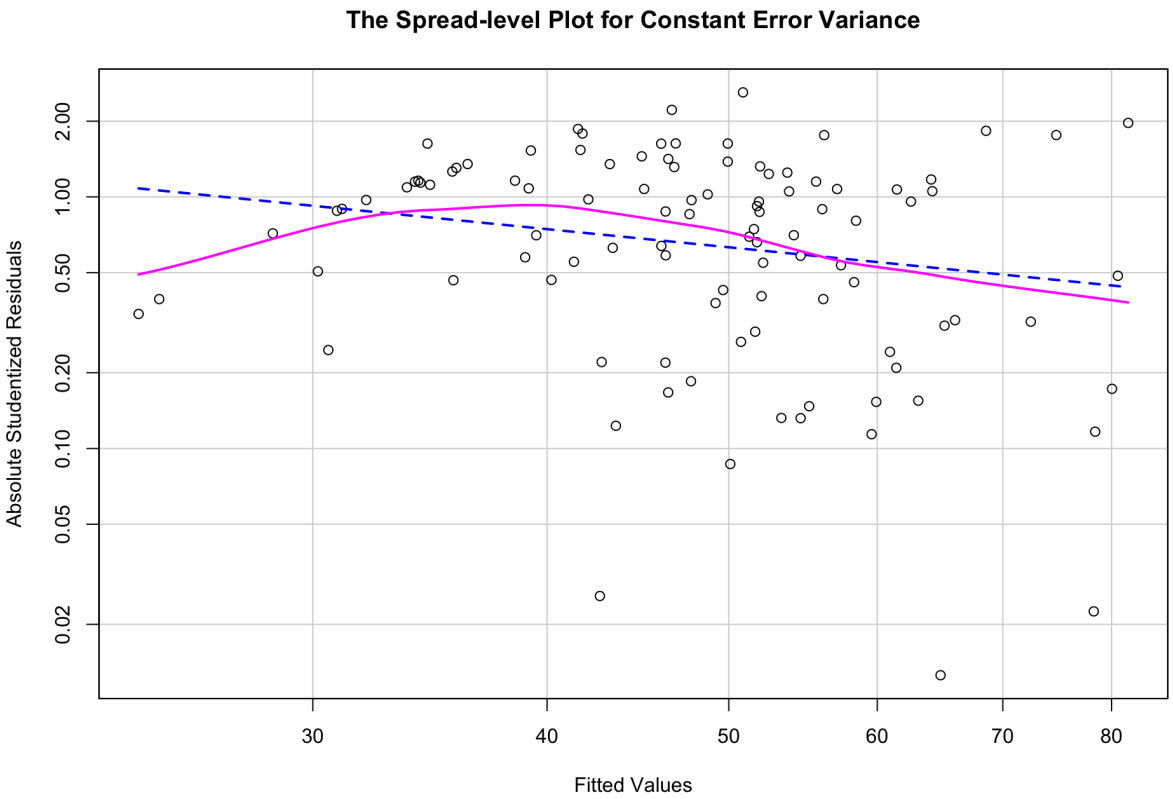
Interpretation

The Component+Residual Plots is used to observe any nonlinearity relationship between the independent variables (predictors) and the dependent variable, slumpflow. Systematic departure from each plot shows linearity. From the plots above, linearity assumption can be confirmed.

```
#HOMOSCEDASTICITY
library(car)
ncvTest(fit1)
```

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

```
spreadLevelPlot(fit1, main ="The Spread-level Plot for Constant Error Variance")
```




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

Interpretation

The Spread-level plot is for assessing constant error variance. The score test of p=0.63 is nonsignificant, suggesting that the constant variance assupmtion has been met. The suggested power transformation is 1.74 which confirms that transformation is not needed.

```
# Multicollinearity
library(car)
vif(fit1)
```

```
##            Cement              Slag          `Fly Ash`             Water
##          48.570807          55.276977          58.649500          31.431899
##                SP `Coarse Aggregate`   `Fine Aggregate`
##           2.139998          88.171895          49.961057
```

```
sqrt(vif(fit1)) >2
```

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

Interpretation

As seen in the result above, by using a statistic called the variance inflation factor(VIF), there may be multicoliinearity problem.

Comparing what were manually calculated to the Global Test of Model Assumptions

```
#install.packages("gvlma")
library(gvlma)
gvmodel <- gvlma(fit1)
summary(gvmodel)
```

```
##
## 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 = fit1)
##
##                      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.
```

Identify unusual observations and take corrective measures

```
#outliers
library(car)
outlierTest(fit1)
```

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

The outlier test shows No 69 is the outlier.

```
#high leverage points
hat.plot <- function(fit1) {
  p <- length(coefficients(fit1))
  n <- length(fitted(fit1))
  plot(hatvalues(fit1), main = "Index Plot of Hat Values")
  abline(h=c(2,3)*p/n, col="red", lty=2)
  identify(1:n, hatvalues(fit1), names(hatvalues(fit1)))
}
hat.plot(fit1)
```

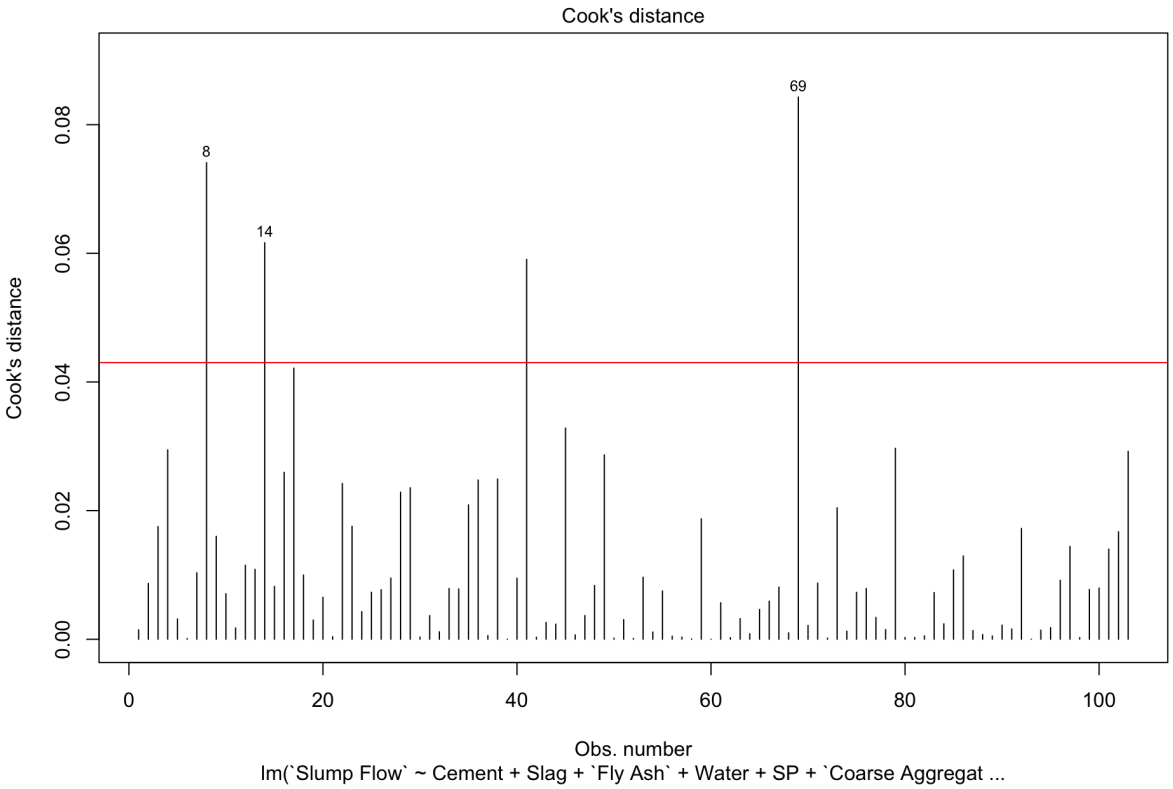

```
## integer(0)
```

Interpretation

The three points are above the red line suggest the presence of outliers.

```
#influential observations
cutoff <- 4/(nrow(df)-length(fit1$coefficients)-2)
plot(fit1, which=4, cook.levels=cutoff)
abline(h=cutoff, hty=2, col="red")
```

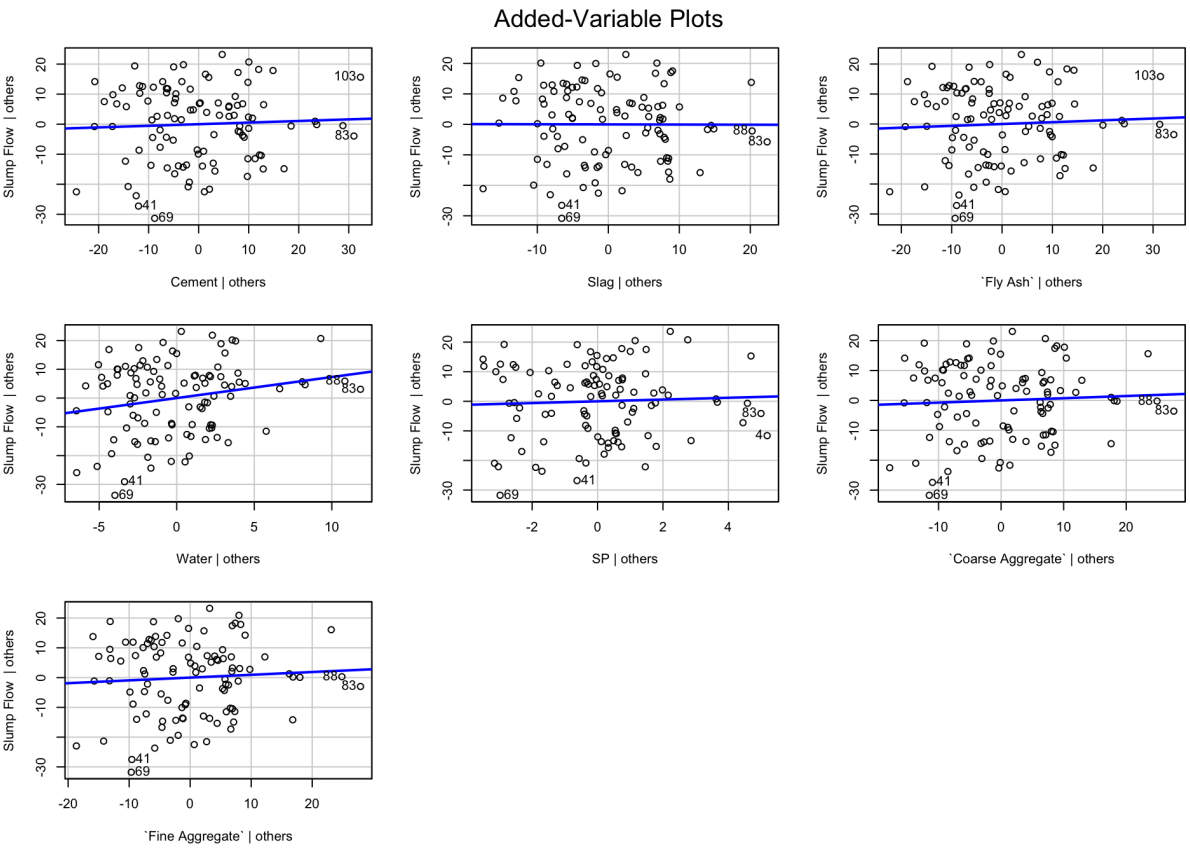
```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "hty" is
## not a graphical parameter
```



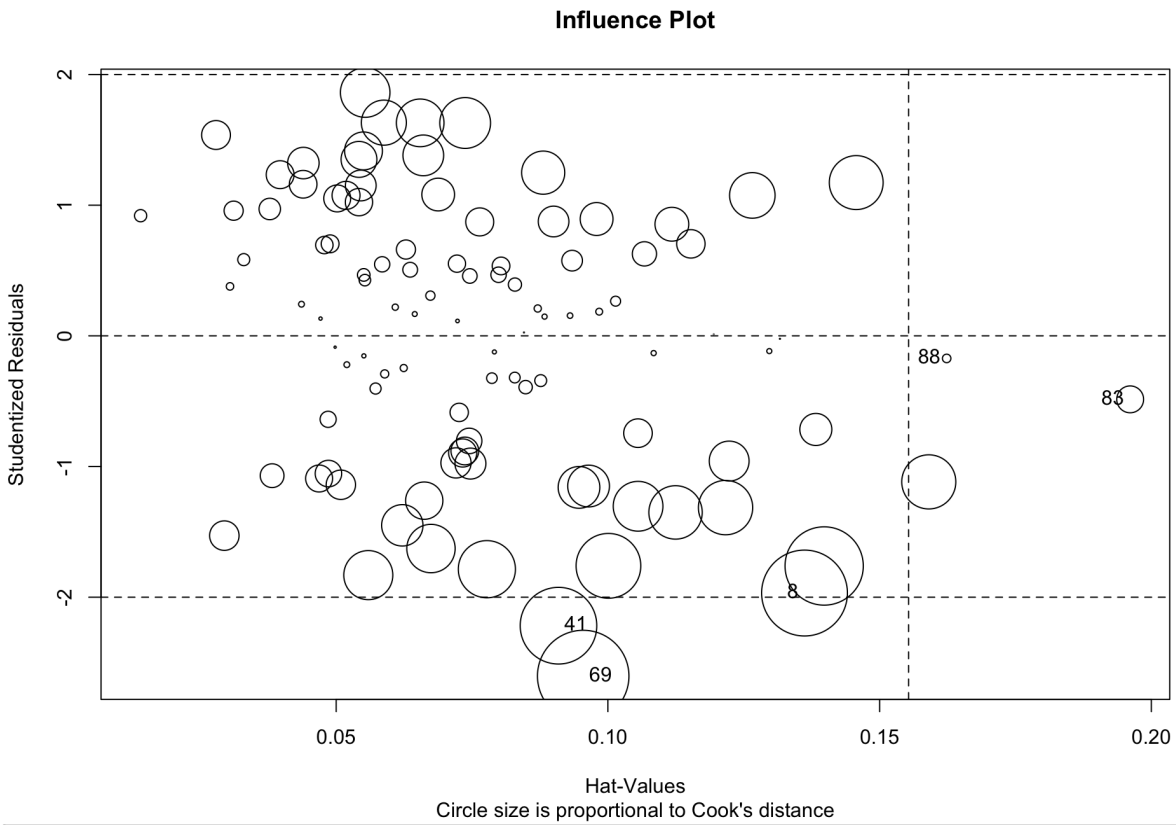
Interpretation

According to Cook's distance, No 8, 14 and 69 are outliers.

```
avPlots(fit1, ask=FALSE, onepage=TRUE, id.method="identify")
```



```
influencePlot(fit1, id.method="identify", main="Influence Plot",
              sub="Circle size is proportional to Cook's distance")
```



	StudRes <dbl>	Hat <dbl>	CookD <dbl>
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

5 rows

Interpretation

According to the influence plot, 8, 41 and 69 are outliers; 83 and 88 are possible influential observations.

Corrective measures

Deleting Observations

Outliers are observations that are poorly fit by the regression model. If outliers are determined influential, they can be removed to avoid serious distortions in the regression calculations. This model only has 3 outliers, so they are likely not significantly influencial to the learning of the model.

Adding or deleting variables

Depending on the current number of predictor variables and their contributions to the model, the adding or removing variables can increase or decrease the accuracy. If the model is currently underfit, then introducing the right variable will increase the accuracy. On the other hand, if the current model has the perfect number of features, introducing new variable will lead to an over-fitting condition, resulting in the decrease in accuracy.

In the “Fine tune the selection of predictor variables” section (later part of this document), it will be shown that the chosen linear regression model is over-fitted with 7 predictors. Improvement could be made to increase acuracy by reducing the number of the predictors to 5.

Using another regression approach

Support Vector Regression (SVR) which acknowledges the presence of non-linearity in the data and provides a proficient prediction model was used ealier in addition to the linear model to form the prodictive model. From the predicted results, it was observed that the RMSE for the linear model,15, was larger than the RMSE of the above SVR model, 14.1 This incicates that the implementation of SVR has an accuracy higher than the linear regression model. Again, this is because the linear model cannot capture the nonlinearity in a dataset and SVR is the best fit in this Concrete Slump Test Data, an evidently nonlinear dataset.

```
# Transforming Variables
library(car)
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

From the results above, there is no transformation needed in this case.

Fine tune the selection of predictor variables

Stepwise Regression - Backward Selection

```
#comparing models
#variable selection- Stepwise Regression
library(MASS)
fit_1 <- lm(`Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` + `Fine Agg
        regate`, data = df)
stepAIC(fit_1, direction="backward")
```

```
## Start:  AIC=533.56
## `Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate`
##
##                       Df Sum of Sq   RSS    AIC
## - Slag                 1      0.22 15672 531.56
## - SP                   1     33.44 15705 531.78
## - Cement               1     37.60 15709 531.81
## - `Fly Ash`            1     47.45 15719 531.87
## - `Coarse Aggregate`   1     49.04 15720 531.88
## - `Fine Aggregate`     1     72.40 15744 532.03
## <none>                             15671 533.56
## - Water                1    709.69 16381 536.12
##
## Step:  AIC=531.56
## `Slump Flow` ~ Cement + `Fly Ash` + Water + SP + `Coarse Aggregate` +
##     `Fine Aggregate`
##
##                       Df Sum of Sq   RSS    AIC
## - SP                   1      62.1 15734 529.97
## <none>                             15672 531.56
## - Cement               1    1244.7 16916 537.43
## - `Coarse Aggregate`   1    1679.4 17351 540.05
## - `Fly Ash`            1    1759.2 17431 540.52
## - `Fine Aggregate`     1    2292.3 17964 543.62
## - Water                1   10877.0 26548 583.86
##
## Step:  AIC=529.97
## `Slump Flow` ~ Cement + `Fly Ash` + Water + `Coarse Aggregate` +
##     `Fine Aggregate`
##
##                       Df Sum of Sq   RSS    AIC
## <none>                             15734 529.97
## - Cement               1    1193.1 16927 535.50
## - `Coarse Aggregate`   1    1678.8 17412 538.41
## - `Fly Ash`            1    1746.5 17480 538.81
## - `Fine Aggregate`     1    2237.1 17971 541.66
## - Water                1   11947.4 27681 586.16
```

```
##
## Call:
## lm(formula = `Slump Flow` ~ Cement + `Fly Ash` + Water + `Coarse Aggregate` +
##     `Fine Aggregate`, data = df)
##
## Coefficients:
##        (Intercept)              Cement           `Fly Ash`               Water
##         -249.50866             0.05366             0.06101             0.72313
## `Coarse Aggregate`    `Fine Aggregate`
##            0.07291             0.09554
```

Interpretation

The model started with 7 predictors in the model. For each step, the AIC column provides the model AIC resulting from the deletion of the variable listed in that row. In the first step, Slag is removed, decreasing the AIC from 534 to 532. In the second step, SP is removed, decreasing the AIC to 530. Deleting any more variables would increase the AIC, so the reduction process stopped.

Another Approach for Stepwise Regression - Both Direction

```
#stat: explanatory model/ best-fit purpose
#For multiple regression model (lm)
#Backward stepwise selection to find the most significant Factors

library(tidyverse)
library(caret)
library(leaps)
library(MASS)

# Set seed for reproducibility
set.seed(505)
#fit the full model
model <- lm (SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate + FineAggregate, da
        ta=Concrete)

# Stepwise regression model
step.model1 <- stepAIC(model, direction = "both",
                      trace = FALSE)
summary(step.model1) #return to the best final model
```

```
##
## Call:
## lm(formula = SlumpFlow ~ Cement + FlyAsh + Water + CoarseAggregate +
##     FineAggregate, data = Concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.893 -10.125   1.773   9.559  23.914
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -249.50866   48.90884  -5.102 1.67e-06 ***
## Cement             0.05366    0.01979   2.712 0.007909 **
## FlyAsh             0.06101    0.01859   3.281 0.001436 **
## Water              0.72313    0.08426   8.582 1.53e-13 ***
## CoarseAggregate    0.07291    0.02266   3.217 0.001760 **
## FineAggregate      0.09554    0.02573   3.714 0.000341 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.74 on 97 degrees of freedom
## Multiple R-squared:  0.5003, Adjusted R-squared:  0.4745
## F-statistic: 19.42 on 5 and 97 DF,  p-value: 2.36e-13
```

```
accuracy(step.model1)
```

```
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -4.137821e-16 12.35932 10.28612 -10.75286 27.77644 0.7061536
```

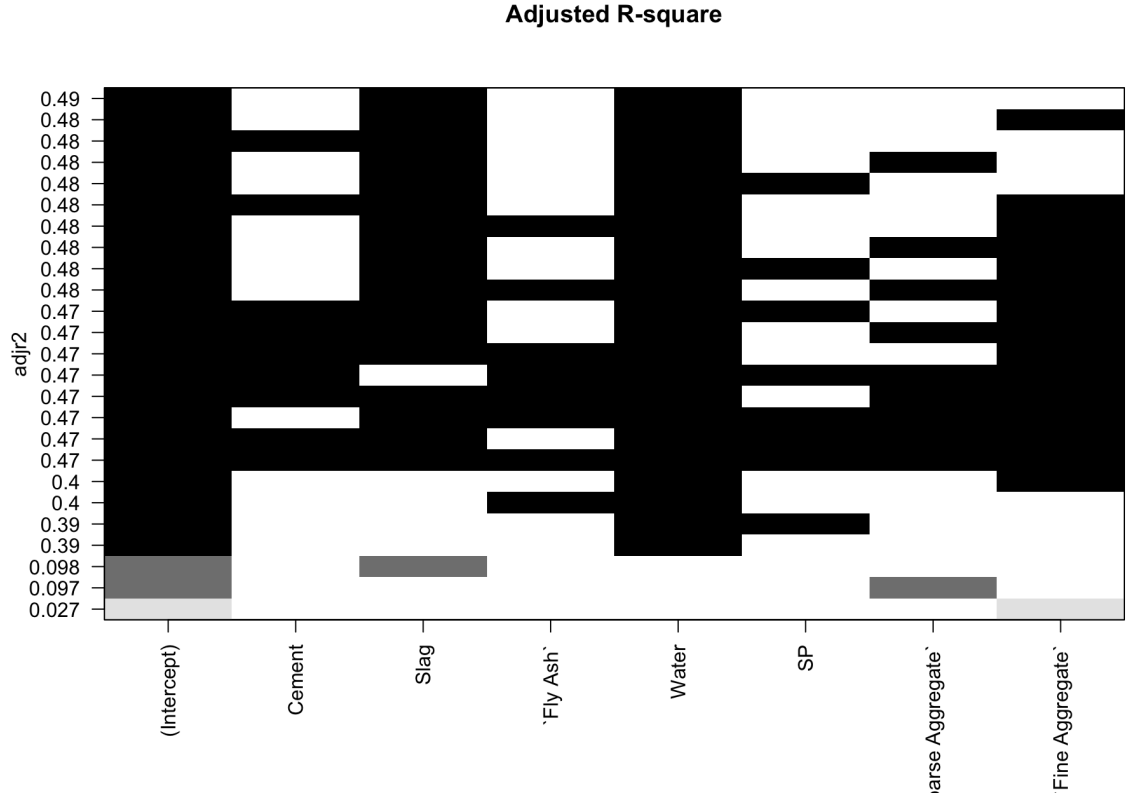
Interpretation

In this approach, the model also started with 7 predictors and ended with 5 predictors. Note that these 5 predictors are exactly the same with those of Stepwise Regression - Backward Selection.

Also, note that the coefficients of 5 predictors in both models are mostly identical.

All Subsets Regression

```
#variable selection- all subsets regression
#install.packages("leaps")
library(leaps)
leaps <- regsubsets(`Slump Flow` ~ Cement + Slag + `Fly Ash` + Water + SP + `Coarse Aggregate` + `
        Fine Aggregate`, data = df, nbest=4)
plot(leaps, scale="adjr2", main = "Adjusted R-square")
```



Interpretation

It can be observed that a model with the intercept and Fine Aggregate has the adjusted R-square of 0.027. A model with Intercept, Water, and SP has the adjusted R-square of about 0.39. The higher the R-squared, the better the model fits your data. This concludes that if we were to pick between these 2 models, the later one with more predictor variables would be a better fit.

Cross-validation

```
# use 10-fold cross-validation to estimate the average prediction error (RMSE) of each of the 7 mo
        dels

# Set seed for reproducibility
set.seed(1000)
# Set up repeated k-fold cross-validation
#use the 10-fold cross-validation
train.control<- trainControl(method = "cv", number = 10)
# Train the model
step.model2 <- train(SlumpFlow ~ Cement + Slag + FlyAsh + Water + SP + CoarseAggregate + FineAggre
        gate, data =Concrete,
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax = 1:7),
                    trControl = train.control
                    )
step.model2$results
```

nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	13.53468	0.4386924	11.54070	2.124631	0.2085134	1.855373
2	2	12.47226	0.5257535	10.49224	2.029738	0.2104619	2.020491
3	3	12.93169	0.4912365	11.00424	2.162058	0.2420788	2.333529
4	4	12.86315	0.5001346	10.77131	1.796669	0.1990204	1.817584
5	5	13.08838	0.4849560	10.97465	1.728781	0.1931265	1.791776
6	6	13.09559	0.4857257	11.01709	1.709990	0.1928892	1.811763
7	7	13.12814	0.4851796	11.06444	1.730804	0.1891807	1.846877

7 rows

```
 #Fine tune the selection of predictor variables
step.model2$bestTune
```

	nvmax <int>
2	2

1 row

```
 #therefore, the best prediction model with 5 predictor variables

#get the final model
summary(step.model2$finalModel)
```

```
## Subset selection object
## 7 Variables  (and intercept)
##                 Forced in Forced out
## Cement              FALSE      FALSE
## Slag                FALSE      FALSE
## FlyAsh              FALSE      FALSE
## Water               FALSE      FALSE
## SP                  FALSE      FALSE
## CoarseAggregate     FALSE      FALSE
## FineAggregate       FALSE      FALSE
## 1 subsets of each size up to 2
## Selection Algorithm: 'sequential replacement'
##          Cement Slag FlyAsh Water SP  CoarseAggregate FineAggregate
## 1  ( 1 ) " "    " "  " "    "*"   " " " " " "             " "
## 2  ( 1 ) " "    "*"  " "    "*"   " " " " " "             " "
```

```
modelaa <- lm(SlumpFlow ~ Cement +FlyAsh + Water + CoarseAggregate + FineAggregate, data =Concret
       e)
summary(modelaa)
```

```
##
## Call:
## lm(formula = SlumpFlow ~ Cement + FlyAsh + Water + CoarseAggregate +
##     FineAggregate, data = Concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.893 -10.125   1.773   9.559  23.914
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -249.50866   48.90884  -5.102 1.67e-06 ***
## Cement             0.05366    0.01979   2.712 0.007909 **
## FlyAsh             0.06101    0.01859   3.281 0.001436 **
## Water              0.72313    0.08426   8.582 1.53e-13 ***
## CoarseAggregate    0.07291    0.02266   3.217 0.001760 **
## FineAggregate      0.09554    0.02573   3.714 0.000341 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.74 on 97 degrees of freedom
## Multiple R-squared:  0.5003, Adjusted R-squared:  0.4745
## F-statistic: 19.42 on 5 and 97 DF,  p-value: 2.36e-13
```

```
accuracy(modelaa)
```

```
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -4.137821e-16 12.35932 10.28612 -10.75286 27.77644 0.7061536
```

Interpretation

In the 10-fold cross-validation process, the best fitted model contains five variables: Cement + FlyAsh + Water + CoarseAggregate + FineAggregate. This matches with the result from the stepwise selection. The RMSE drops to 12.36

Interpret the prediction results

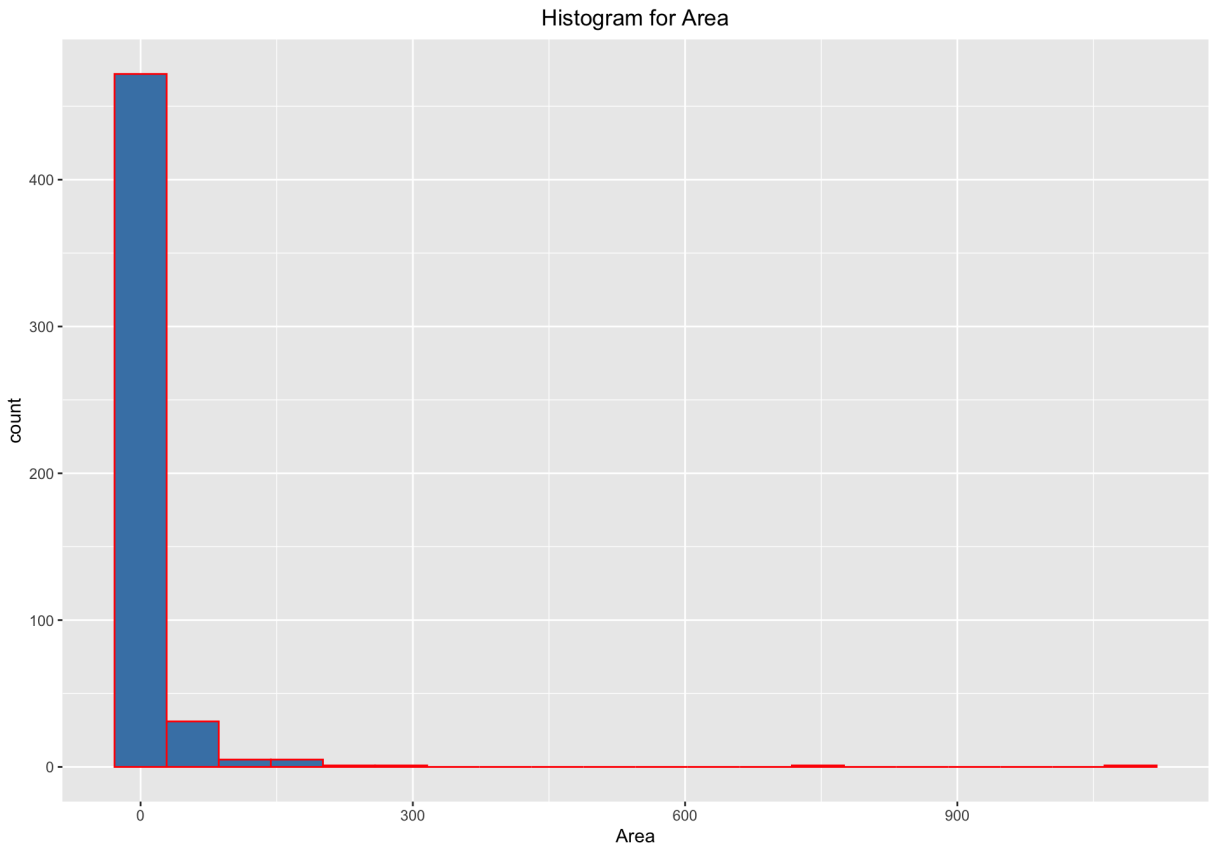
After the fine-tuning processes along with the 10-fold cross-validation, it has been confrimed that the best regression model is:

Slump Flow = -249.5087 + (0.0537)Cement + (0.0610)FlyAsh + (0.7231)Water + (0.0729)CoarseAggregate + (0.0955)FineAggregate

Problem 2: Forest Fire Data

```
library(ggplot2)
library(forecast)
library(readxl)
library(leaps)
library(tidyverse)
mydata <- read_excel("Forest Fires Data.xlsx")
#head(mydata)
#summary(mydata)
```

```
##Data exploration of the available data-set for forest fires.
##First, we will plot the histogram for the output/response variable, that is, the Area.
ggplot(mydata, aes(x=Area)) +
        geom_histogram(bins = 20,color="red", fill="steelblue") +
        ggtitle('Histogram for Area') +
        theme(plot.title = element_text(hjust = 0.5))
```

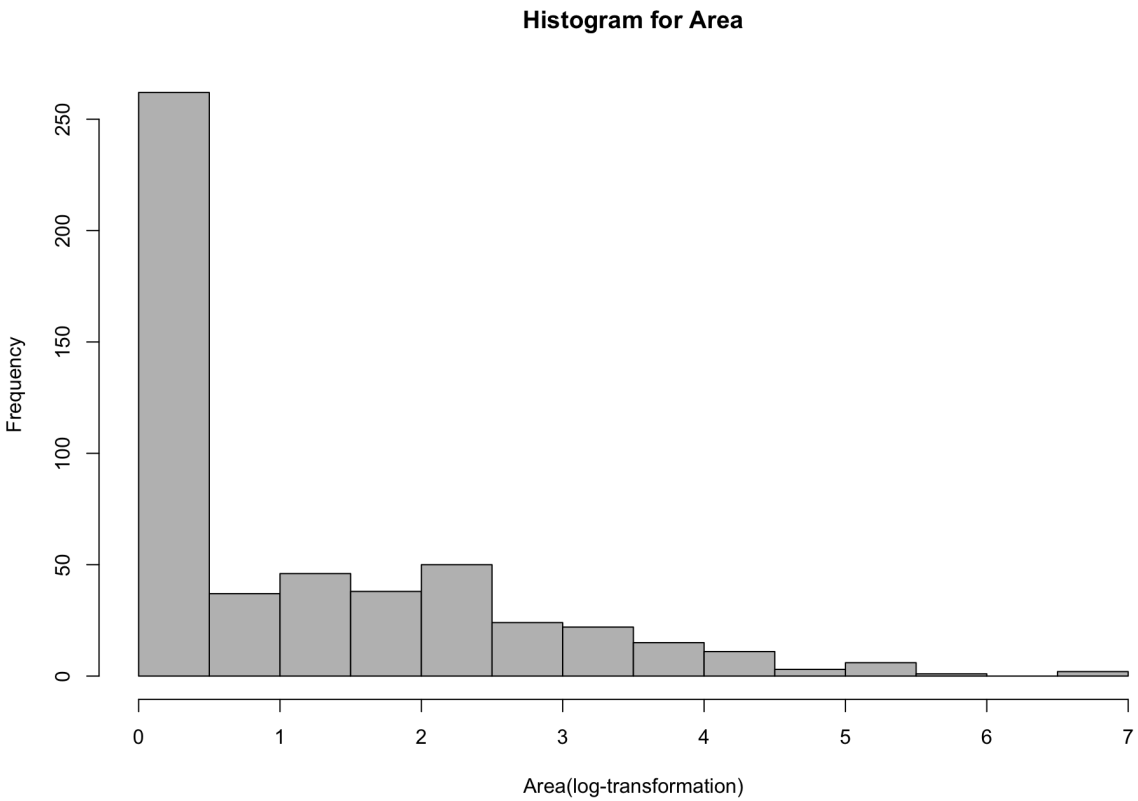


Interpretation

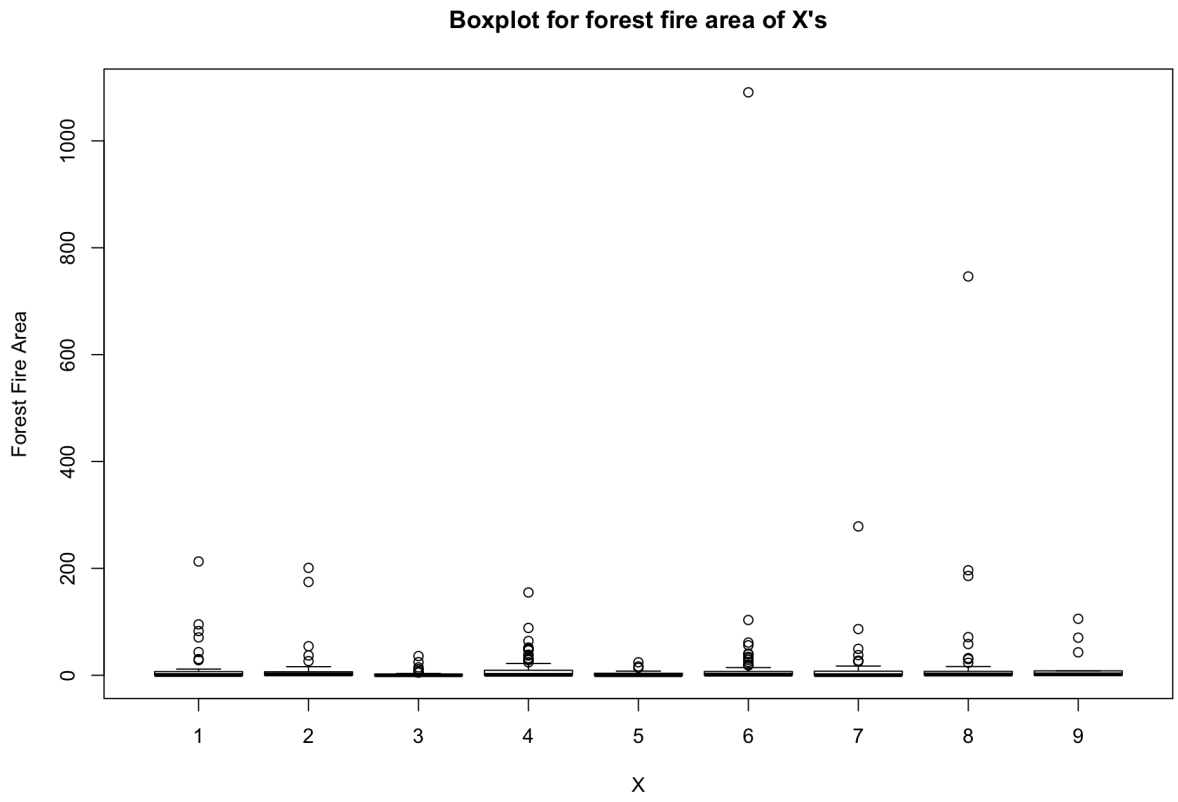
It can be observed from the above histogram that the data distribution for Area is skewed towards 0, so we have to apply log tranformation to the data and visualise the data.

After applying the log transformation to the data distribution of Area variable:

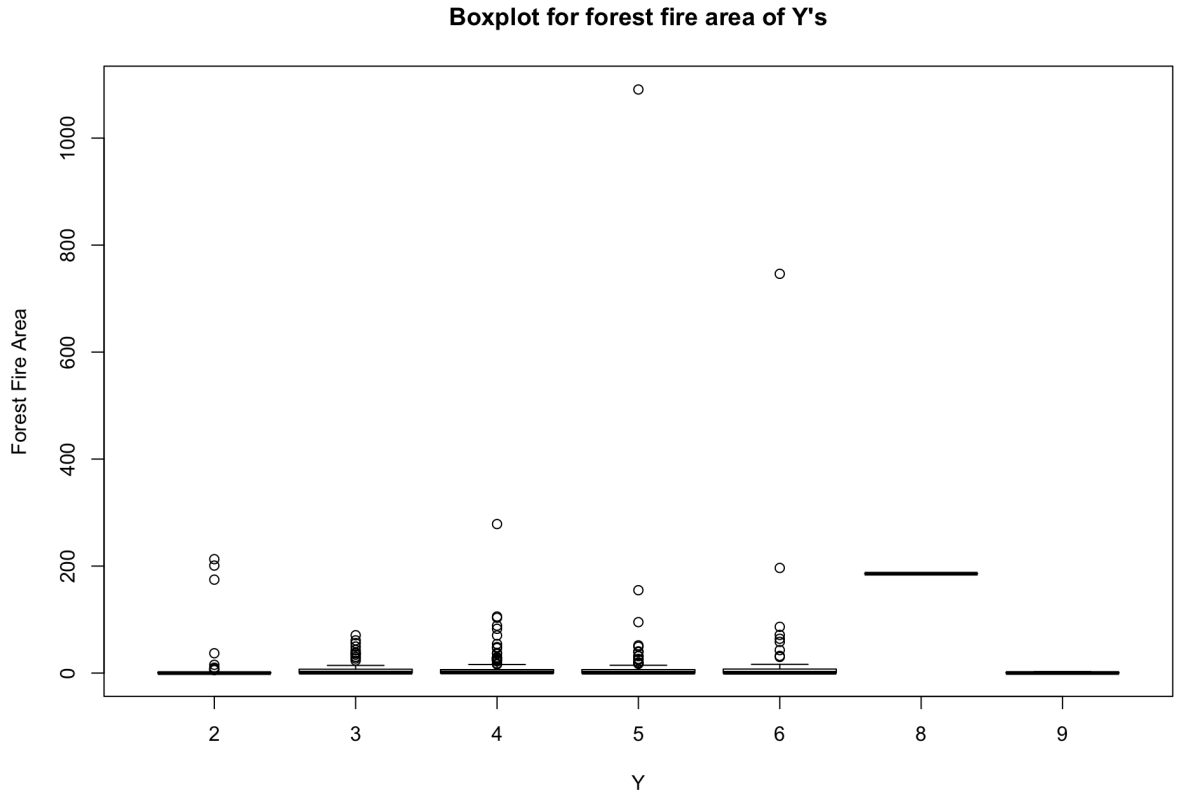
```
hist(log1p(mydata$Area),
     main = "Histogram for Area",
     xlab = "Area(log-transformation)",
     col = "grey")
```




```
##Now using the boxplot, to show the connection between the forest fire area and the Spatial Varia
        bles-X&Y.
par(mfrow = c(1,1))
boxplot(mydata$Area ~ as.factor(X), data = mydata,
        xlab = "X", ylab = "Forest Fire Area",
        main = "Boxplot for forest fire area of X's")
```



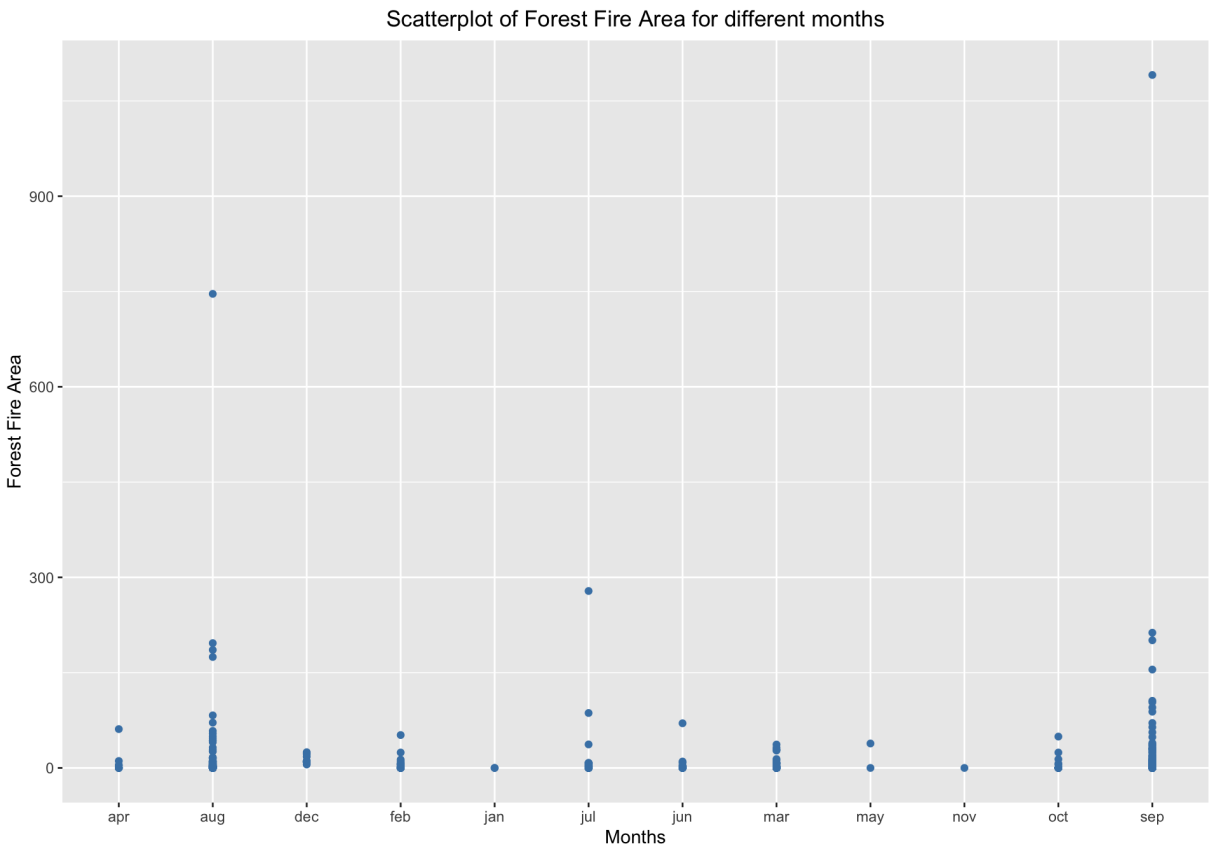

```
boxplot(mydata$Area ~ as.factor(Y), data = mydata,
        xlab = "Y", ylab = "Forest Fire Area",
        main = "Boxplot for forest fire area of Y's")
```



Interpretation

It can be interpreted from the above boxplots that there is no obvious kind of relationship among the Spatial Variables(X&Y) and Forest Fire Area.

```
ggplot(mydata, aes(x=Month, y=Area)) +
        geom_point(color='steelblue', fill='yellow') +
        ggtitle('Scatterplot of Forest Fire Area for different months') +
        theme(plot.title = element_text(hjust = 0.5)) +
        xlab('Months') + ylab('Forest Fire Area')
```

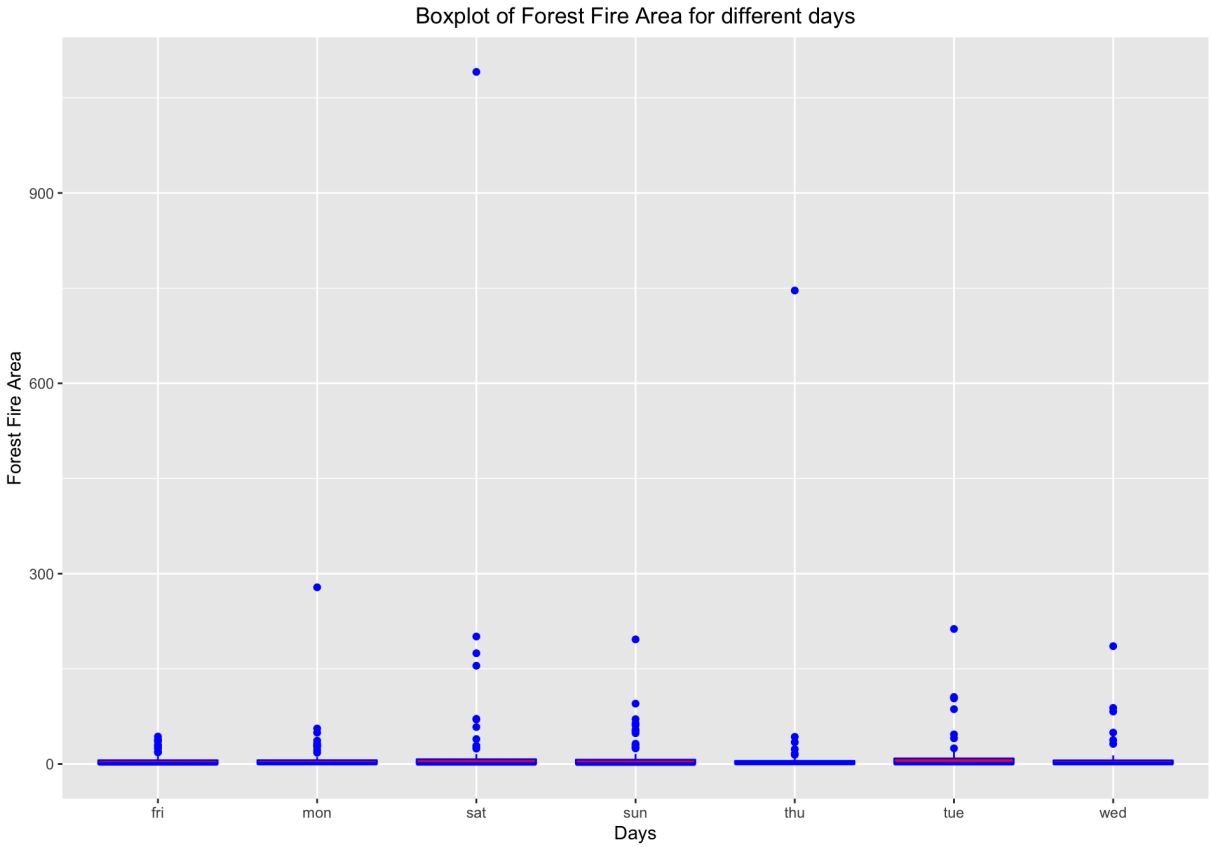


Interpretation

It can be observed from the above scatterplot that the observations in each month are unbalanced, and the observations have a higher risk of getting overfitted.

Now, we will visualize the dataset to explore the relationship between the days and the forest fire area.

```
ggplot(mydata, aes(x=Day, y=Area)) +
        geom_boxplot(color='blue', fill='red') +
        ggtitle('Boxplot of Forest Fire Area for different days') +
        theme(plot.title = element_text(hjust = 0.5)) +
        xlab('Days') + ylab('Forest Fire Area')
```



Interpretation

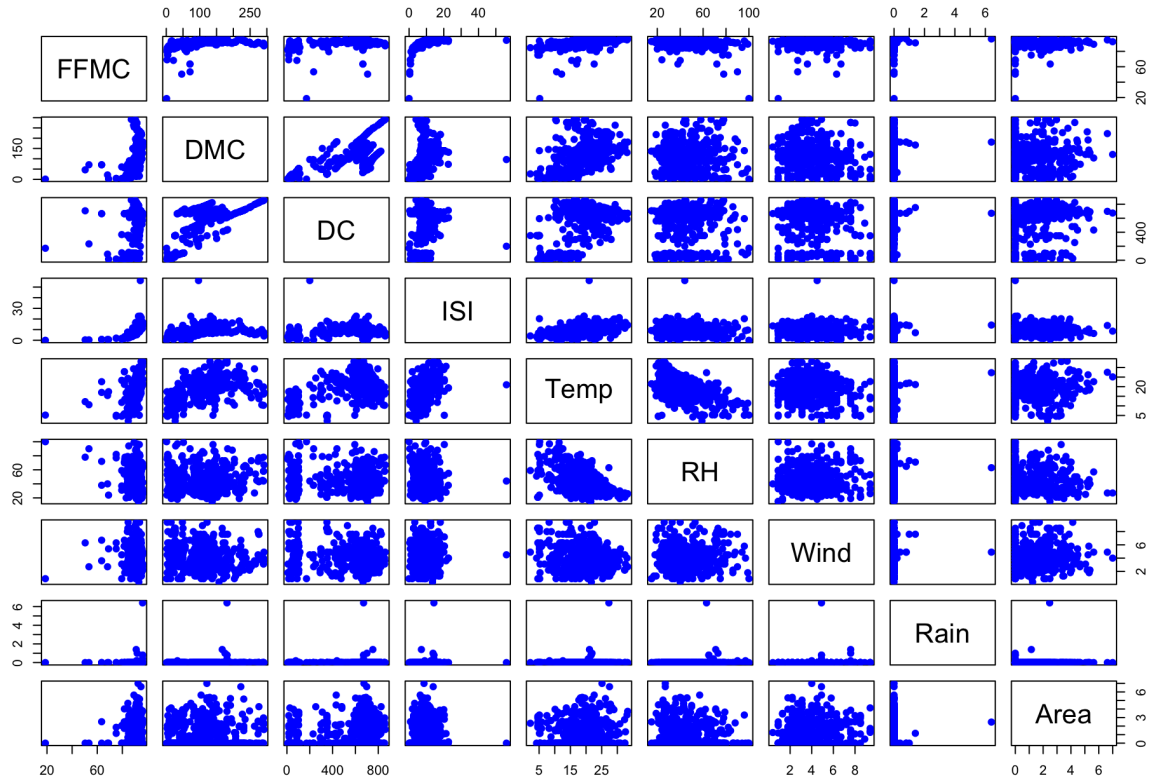
It can be interpreted from the above boxplot that Saturday has higher chance of getting the forest fire.

```
mydata$Month <- as.numeric(as.factor(mydata$Month))
mydata$Day <- as.numeric(as.factor(mydata$Day))
mydata$Area<-log1p(mydata$Area)
glimpse(mydata)
```

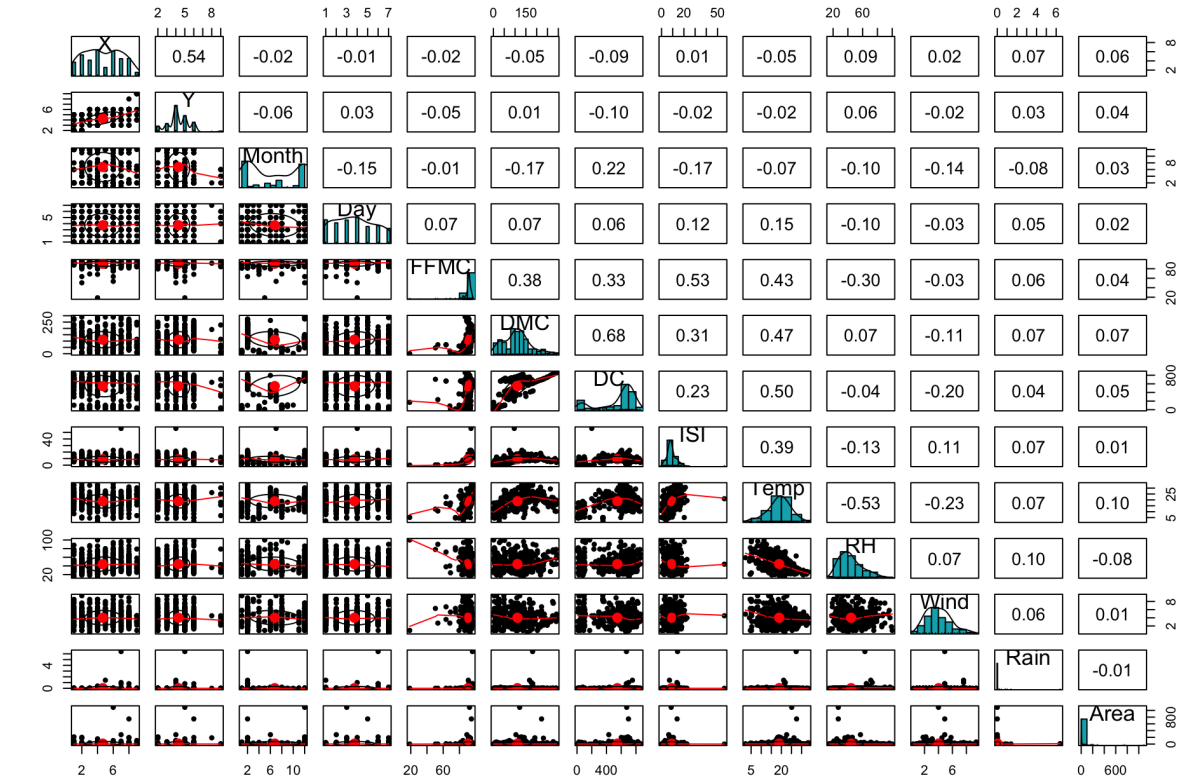
```
## Observations: 517
## Variables: 13
## $ X     <dbl> 7, 7, 7, 8, 8, 8, 8, 8, 8, 7, 7, 7, 6, 6, 6, 6, 5, 8, 6, 6, 6, …
## $ Y     <dbl> 5, 4, 4, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, …
## $ Month <dbl> 8, 11, 11, 8, 8, 2, 2, 2, 12, 12, 12, 12, 2, 12, 12, 12, 8, 11,…
## $ Day   <dbl> 1, 6, 3, 1, 4, 4, 2, 2, 6, 3, 3, 3, 1, 2, 7, 1, 3, 2, 7, 3, 6, …
## $ FFMC  <dbl> 86.2, 90.6, 90.6, 91.7, 89.3, 92.3, 92.3, 91.5, 91.0, 92.5, 92.…
## $ DMC   <dbl> 26.2, 35.4, 43.7, 33.3, 51.3, 85.3, 88.9, 145.4, 129.5, 88.0, 8…
## $ DC    <dbl> 94.3, 669.1, 686.9, 77.5, 102.2, 488.0, 495.6, 608.2, 692.6, 69…
## $ ISI   <dbl> 5.1, 6.7, 6.7, 9.0, 9.6, 14.7, 8.5, 10.7, 7.0, 7.1, 7.1, 22.6, …
## $ Temp  <dbl> 8.2, 18.0, 14.6, 8.3, 11.4, 22.2, 24.1, 8.0, 13.1, 22.8, 17.8, …
## $ RH    <dbl> 51, 33, 33, 97, 99, 29, 27, 86, 63, 40, 51, 38, 72, 42, 21, 44,…
## $ Wind  <dbl> 6.7, 0.9, 1.3, 4.0, 1.8, 5.4, 3.1, 2.2, 5.4, 4.0, 7.2, 4.0, 6.7…
## $ Rain  <dbl> 0.0, 0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0…
## $ Area  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
```

Create a scatterplot matrix of "Forest Fires Data" and select an initial set of predictor variables

```
pairs(mydata[,5:13], pch = 19,col='blue')
```



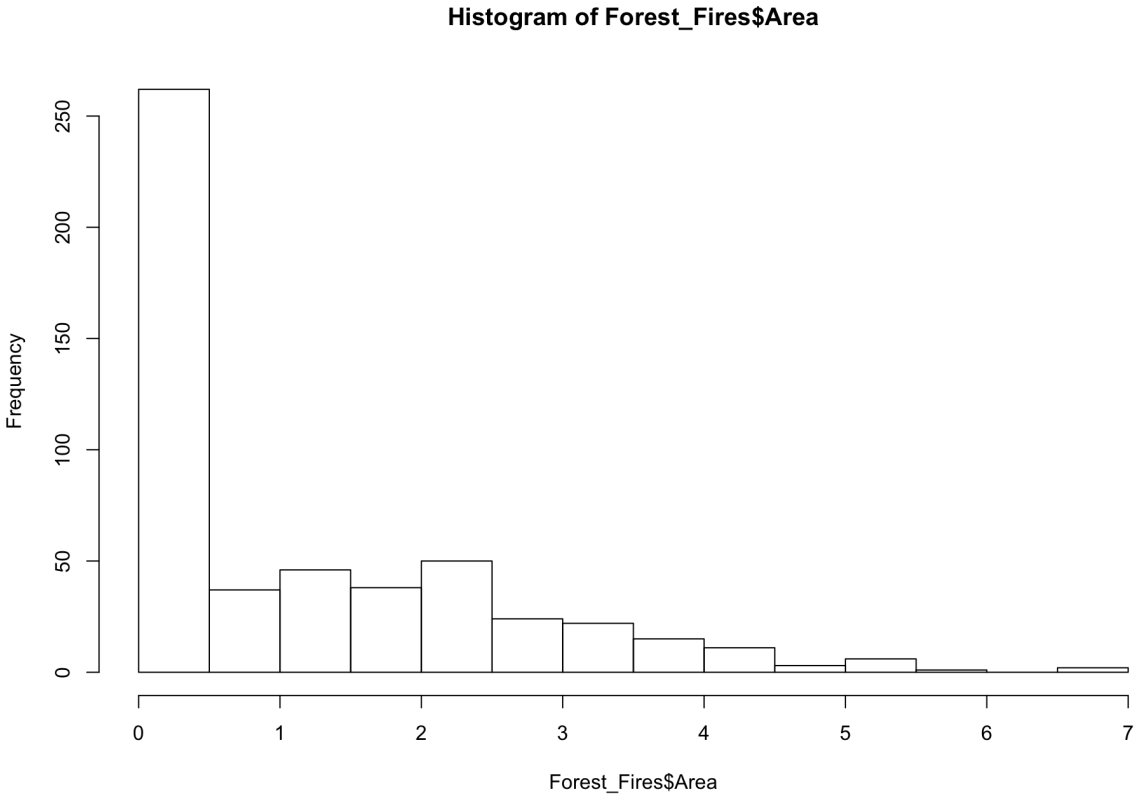

```r
# another way of scatterplot
library(readxl)
library(forecast)
library(tidyverse)
library(caret)
library(rpart)
library(caret)
library(e1071)
library(data.table)

#upload the dataset
Forest_Fires <- read_excel("./Forest Fires Data.xlsx")
Forest_Fires$Month <- as.numeric(as.factor(Forest_Fires$Month))
Forest_Fires$Day <- as.numeric(as.factor(Forest_Fires$Day))

#get the scatter matrix
library(psych)
pairs.panels(Forest_Fires,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



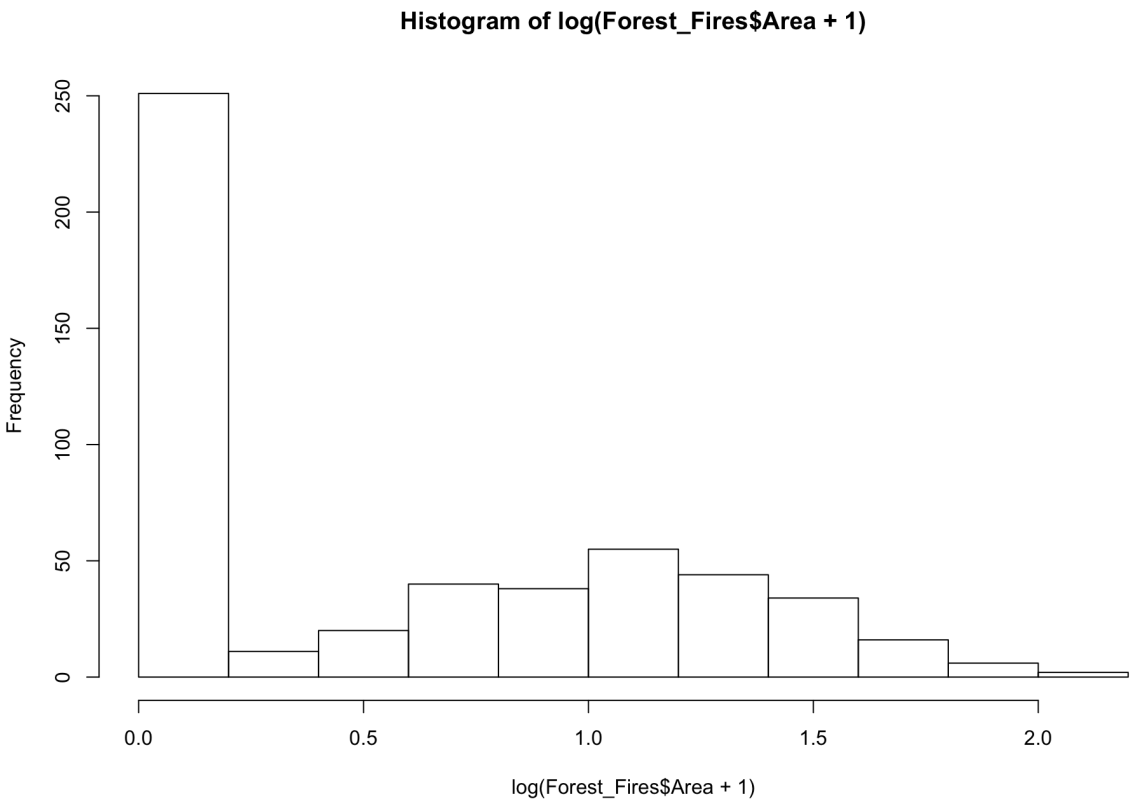
Interpretation

The Pearson correlation is used and according to the results: Relative humidity(RH) and Wind Speed are highly correlated. We will be using area as the response variable.

```
#log transformation the dataset
Forest_Fires$Area <- log1p(Forest_Fires$Area)
hist(Forest_Fires$Area)
```



```
hist(log(Forest_Fires$Area+1))
```



Interpretation

The logarithm function is applied to the Area variable transforming a highly skewed variable into a more normalized one.

However, for this specific dataset, there are many data points with zero values in Area column after transformation. This simple transformation has distinct drawbacks, affecting regression accuracy.

Build a few potential regression models using "ForestFireData"

Starting with building a very simple model:

```
mod2 <- lm(Area ~., data = mydata)
summary(mod2)
```

```
##
## Call:
## lm(formula = Area ~ ., data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6785 -1.0875 -0.5743  0.8840  5.5821
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1571321  1.3930842  -0.113   0.9102
## X            0.0401684  0.0317946   1.263   0.2070
## Y            0.0136655  0.0600928   0.227   0.8202
## Month        0.0173175  0.0170180   1.018   0.3094
## Day          0.0226868  0.0328330   0.691   0.4899
## FFMC         0.0063487  0.0145561   0.436   0.6629
## DMC          0.0017682  0.0015803   1.119   0.2637
## DC           0.0001277  0.0004120   0.310   0.7567
## ISI         -0.0229718  0.0170871  -1.344   0.1794
## Temp         0.0042540  0.0175928   0.242   0.8090
## RH          -0.0049939  0.0052697  -0.948   0.3438
## Wind         0.0808763  0.0368542   2.194   0.0287 *
## Rain         0.0760935  0.2127150   0.358   0.7207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.395 on 504 degrees of freedom
## Multiple R-squared:  0.02793,    Adjusted R-squared:  0.004783
## F-statistic: 1.207 on 12 and 504 DF,  p-value: 0.2749
```

Interpretation

It can be observed that, the model has low accuracy as the R-square is very small. The reason for this might be that the predictors might not have sufficient information to justify the response.Thus, to fix this, we will apply quadratic terms among the four FWI system indices which are- FFMC,DMCM,DC,ISI

```
finalmod <- lm(Area ~ X+Y+Month + Day + (FFMC + DMC + DC + ISI)^2 + Temp + RH + Wind + Rain, data
         = mydata)
summary(finalmod)
```

```
##
## Call:
## lm(formula = Area ~ X + Y + Month + Day + (FFMC + DMC + DC +
##     ISI)^2 + Temp + RH + Wind + Rain, data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8554 -1.0953 -0.5259  0.8757  5.5569
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.073e-02  2.099e+00  -0.015    0.988
## X            4.300e-02  3.208e-02   1.340    0.181
## Y            6.430e-03  6.060e-02   0.106    0.916
## Month        2.602e-02  1.841e-02   1.414    0.158
## Day          2.042e-02  3.327e-02   0.614    0.540
## FFMC         5.621e-03  2.464e-02   0.228    0.820
## DMC         -3.432e-02  4.329e-02  -0.793    0.428
## DC           2.919e-03  6.460e-03   0.452    0.652
## ISI          9.461e-02  5.701e-01   0.166    0.868
## Temp         4.503e-04  1.927e-02   0.023    0.981
## RH          -5.048e-03  5.397e-03  -0.935    0.350
## Wind         8.301e-02  3.791e-02   2.190    0.029 *
## Rain         4.652e-02  2.165e-01   0.215    0.830
## FFMC:DMC     4.064e-04  4.860e-04   0.836    0.403
## FFMC:DC     -2.811e-05  7.534e-05  -0.373    0.709
## FFMC:ISI    -1.539e-03  6.045e-03  -0.255    0.799
## DMC:DC      -3.722e-06  6.773e-06  -0.549    0.583
## DMC:ISI      2.334e-04  4.263e-04   0.547    0.584
## DC:ISI      -1.048e-05  8.478e-05  -0.124    0.902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.4 on 498 degrees of freedom
## Multiple R-squared:  0.03221,    Adjusted R-squared:  -0.002774
## F-statistic: 0.9207 on 18 and 498 DF,  p-value: 0.5533
```

Interpretation

Now this model looks much better than the previous model and now we have an acceptable F-test.

```
#multiple regression (ML)
# Set seed for reproducibility
set.seed(1000)

#set the training and testing dataset p=0.7
#seprate the outcome from the predictor varaibles
Forest_Fires_Area <- Forest_Fires$Area
idx <- createDataPartition(Forest_Fires_Area, p = 0.7, list = FALSE)
train_set <- Forest_Fires[idx, ]
test_set <- Forest_Fires[-idx, ]

regressorlm <- lm(Area ~.,data =train_set)
summary(regressorlm)
```

```
##
## Call:
## lm(formula = Area ~ ., data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6368 -1.0397 -0.5465  0.8223  4.6217
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.7900684  1.5360150  -0.514   0.6073
## X            0.0245785  0.0375316   0.655   0.5130
## Y            0.0508594  0.0730078   0.697   0.4865
## Month        0.0282936  0.0200460   1.411   0.1590
## Day          0.0579419  0.0395127   1.466   0.1434
## FFMC         0.0141774  0.0156687   0.905   0.3662
## DMC          0.0036445  0.0018760   1.943   0.0529 .
## DC          -0.0002920  0.0004839  -0.603   0.5466
## ISI         -0.0202791  0.0183029  -1.108   0.2686
## Temp        -0.0142433  0.0211526  -0.673   0.5012
## RH          -0.0044178  0.0062519  -0.707   0.4803
## Wind         0.0666469  0.0426236   1.564   0.1188
## Rain         0.1195827  0.2129104   0.562   0.5747
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.351 on 351 degrees of freedom
## Multiple R-squared:  0.03713,    Adjusted R-squared:  0.004213
## F-statistic: 1.128 on 12 and 351 DF,  p-value: 0.3359
```

```
summary(regressorlm)$coefficient
```

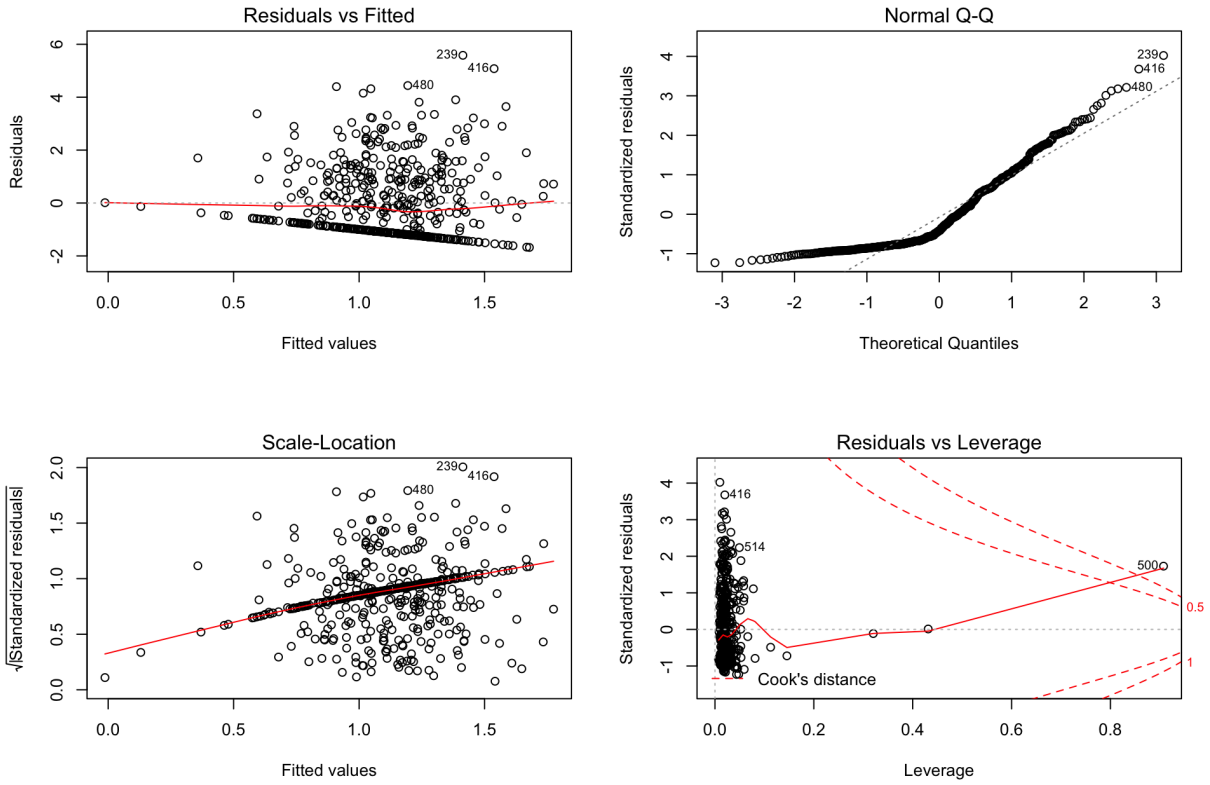
```
##                 Estimate  Std. Error    t value   Pr(>|t|)
## (Intercept) -0.7900683952 1.536015010 -0.5143624 0.60732233
## X            0.0245784942 0.037531558  0.6548754 0.51297689
## Y            0.0508593614 0.073007832  0.6966288 0.48649618
## Month        0.0282935987 0.020046017  1.4114324 0.15900294
## Day          0.0579419003 0.039512748  1.4664103 0.14343186
## FFMC         0.0141774468 0.015668739  0.9048238 0.36617940
## DMC          0.0036444791 0.001876008  1.9426780 0.05285448
## DC          -0.0002919732 0.000483860 -0.6034249 0.54661578
## ISI         -0.0202790904 0.018302873 -1.1079731 0.26863210
## Temp        -0.0142432644 0.021152566 -0.6733587 0.50116224
## RH          -0.0044177868 0.006251889 -0.7066323 0.48026381
## Wind         0.0666469247 0.042623613  1.5636151 0.11880910
## Rain         0.1195827031 0.212910388  0.5616574 0.57470788
```

```
#make the prediction
predicted_area1 <- predict(regressorlm, newdata = test_set)
residuls1 <- test_set$Area[1:153]-predicted_area1[1:153]
d1 <- data.frame("Predicted"= predicted_area1[1:153], "Actual"=test_set$Area[1:153], "Residul"= re
        siduls1)

accuracy(predicted_area1,test_set$Area)
```

```
##                  ME     RMSE      MAE  MPE MAPE
## Test set 0.02705449 1.515455 1.223482 -Inf  Inf
```

Perform regression diagnostics using both typical approach and enhanced approach

```
# a. Typical Approach
par(mfrow = c(2,2))
plot(mod2)
```



Interpretation

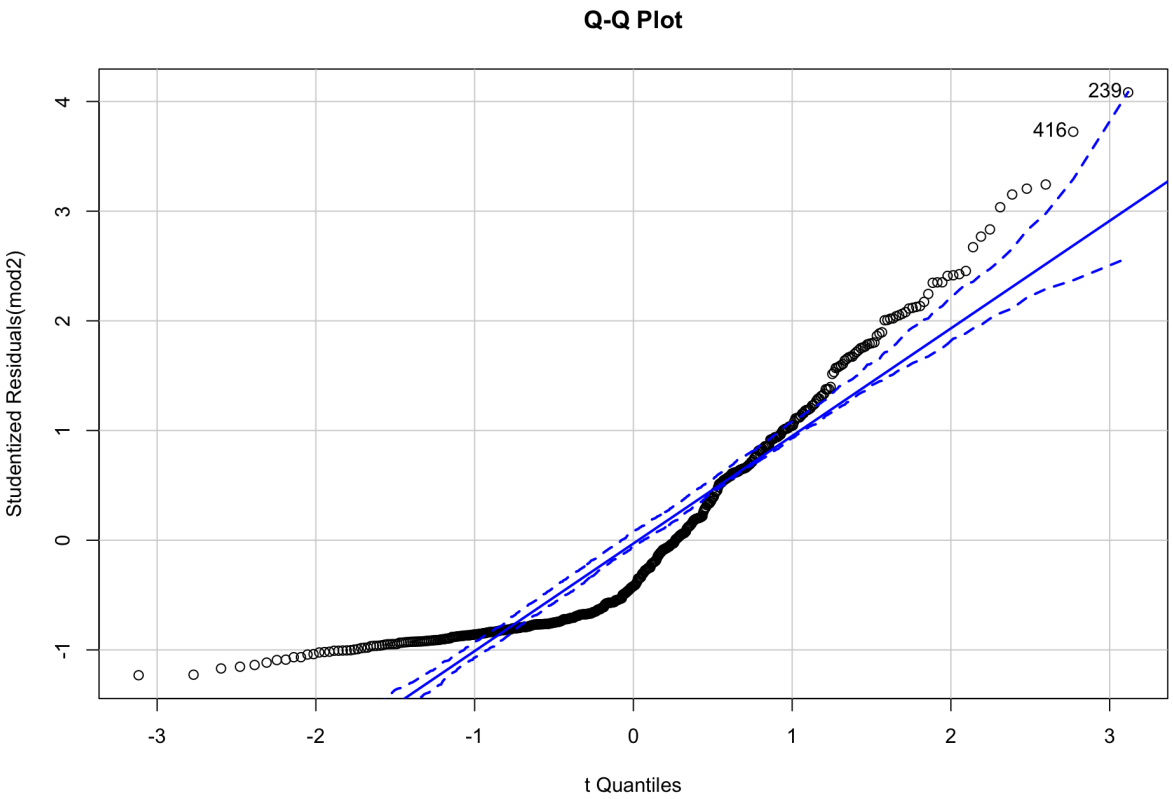
In the residual-fitted plot, the residuals bounce randomly around the 0 line, therefore the assumption that the relationship is linear is reasonable.

In the Normall Q-Q plot, after zero on the x-axis, the points are mostly on the 45-degree line depicting its normality.

In the Scale-Location plot, the points are randomly spread around a horizontal line, hence its homoscedasticity.

In the Residuals-vs-Leverage plot, it can be observe that there are outliers that may be removed.

```
# b. Enhanced Approach
# To determine the Normality
library(ggplot2)
library(car)
par(mfrow = c(1,1))
qqPlot(mod2, labels = row.names(mydata), id.method = "identify", simulate = TRUE, plot.it=TRUE,
 main = "Q-Q Plot")
```




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

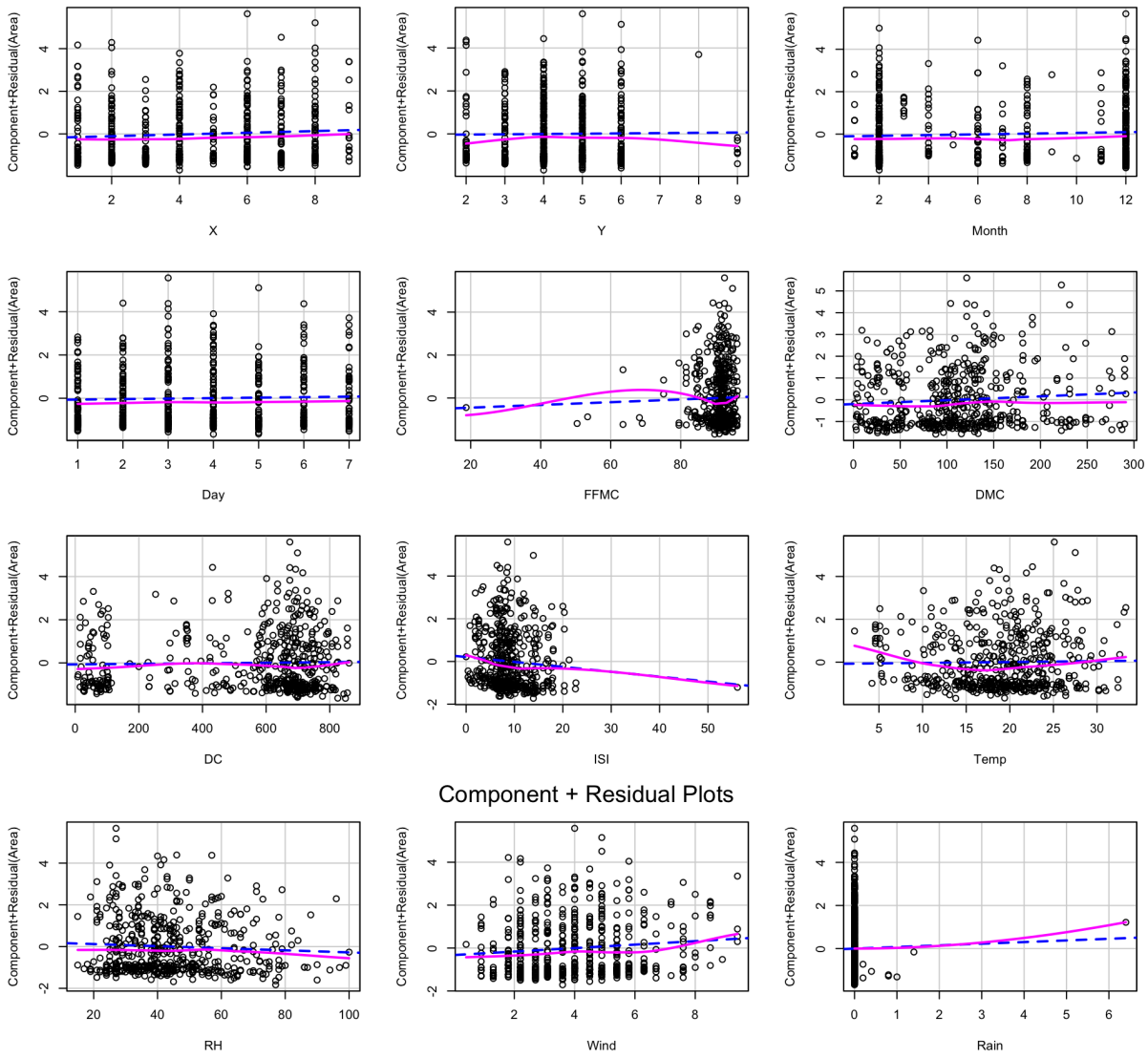
Interpretation

There are a few outliers towards the end of the plot and the model doesn't meet the normality assumption well.

```
#To check the Independence
durbinWatsonTest(mod2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5282327     0.9407048       0
##  Alternative hypothesis: rho != 0
```

```
# To check the Linearity
library(car)
crPlots(mod2)
```



Interpretation

The Component+Residual Plots shows the nonlinearity of the relationship between the independent variables (predictors) and the dependent variable, area. From the plots above, linearity assumption may not be confirmed.

```
#To find the Homoscedasticity
library(car)
ncvTest(finalmod)
```

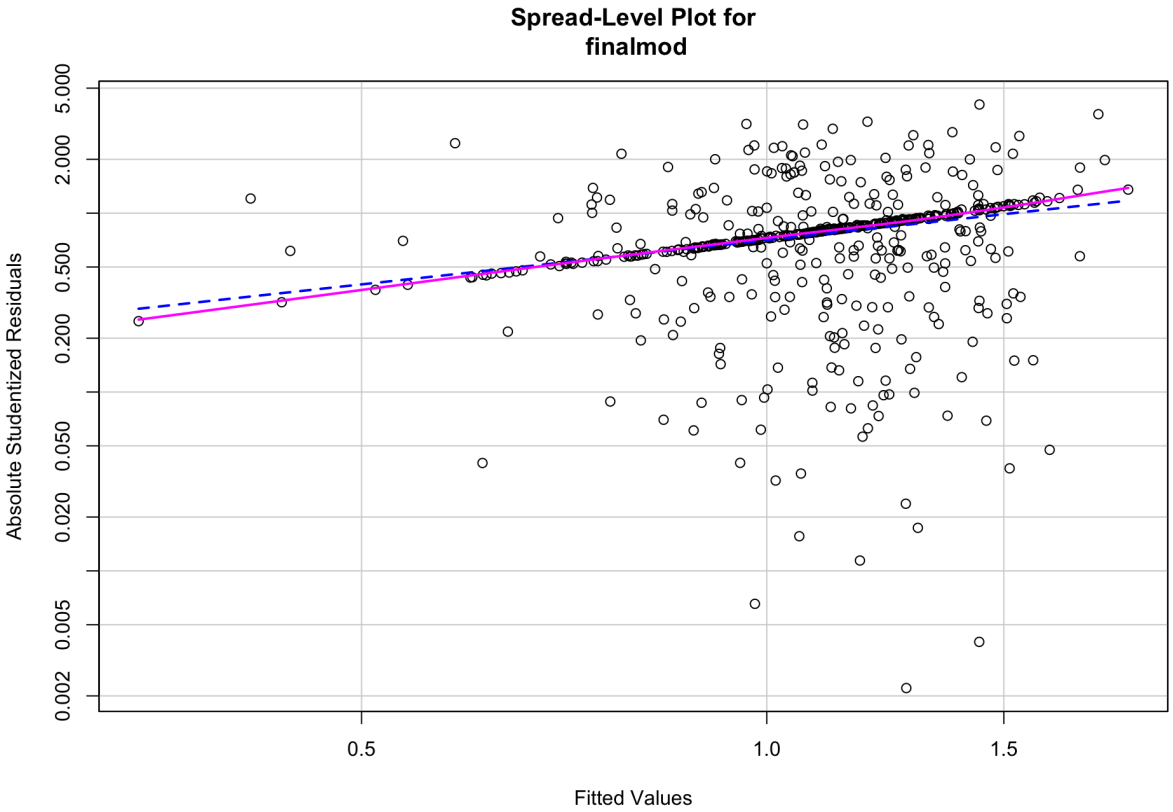
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 21.12886, Df = 1, p = 4.2941e-06
```

Interpretation

p-value is significant suggesting that the constant variance assumption is not satisfied with the non-horizontal line.

```
par(mfrow = c(1,1))
spreadLevelPlot(finalmod)
```

```
## Warning in spreadLevelPlot.lm(finalmod):
## 2 negative fitted values removed
```




```
##
## Suggested power transformation:  0.177467
```

Interpretation

The Spread-level plot is for assessing constant error variance. The p-value is significant, suggesting that the constant variance assupmtion has not been met. The suggested power transformation is 0.177467 which confirms that transformation is needed as the value is moderately far from 1.

```
# Multicollinearity
library(car)
vif(mod2)
```

```
##        X        Y    Month      Day     FFMC      DMC       DC      ISI
## 1.434815 1.448213 1.468514 1.059148 1.711713 2.715938 2.769188 1.609213
##     Temp       RH     Wind     Rain
## 2.766729 1.960343 1.155924 1.050768
```

```
library(car)
sqrt(vif(mod2))>2
```

```
##     X     Y Month   Day  FFMC   DMC    DC   ISI  Temp    RH  Wind  Rain
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Interpretation

As seen in the result above, all show FALSE, therefore, it suggests that there is no multicollinearity problem.

Identify unusual observations and take corrective measures.

Unusual Observations

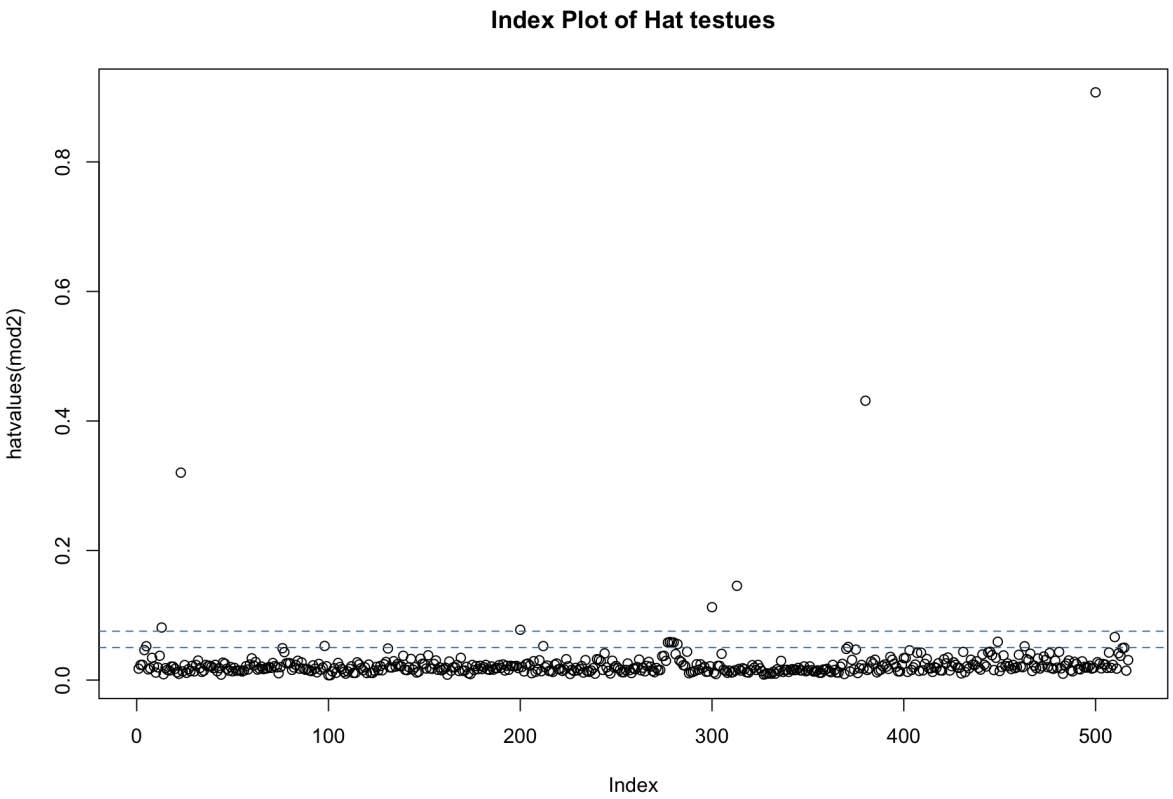
```
# a. Outlier Test
outlierTest(mod2)
```

```
##     rstudent unadjusted p-value Bonferroni p
## 239 4.083257         5.1655e-05     0.026706
```

Interpretation

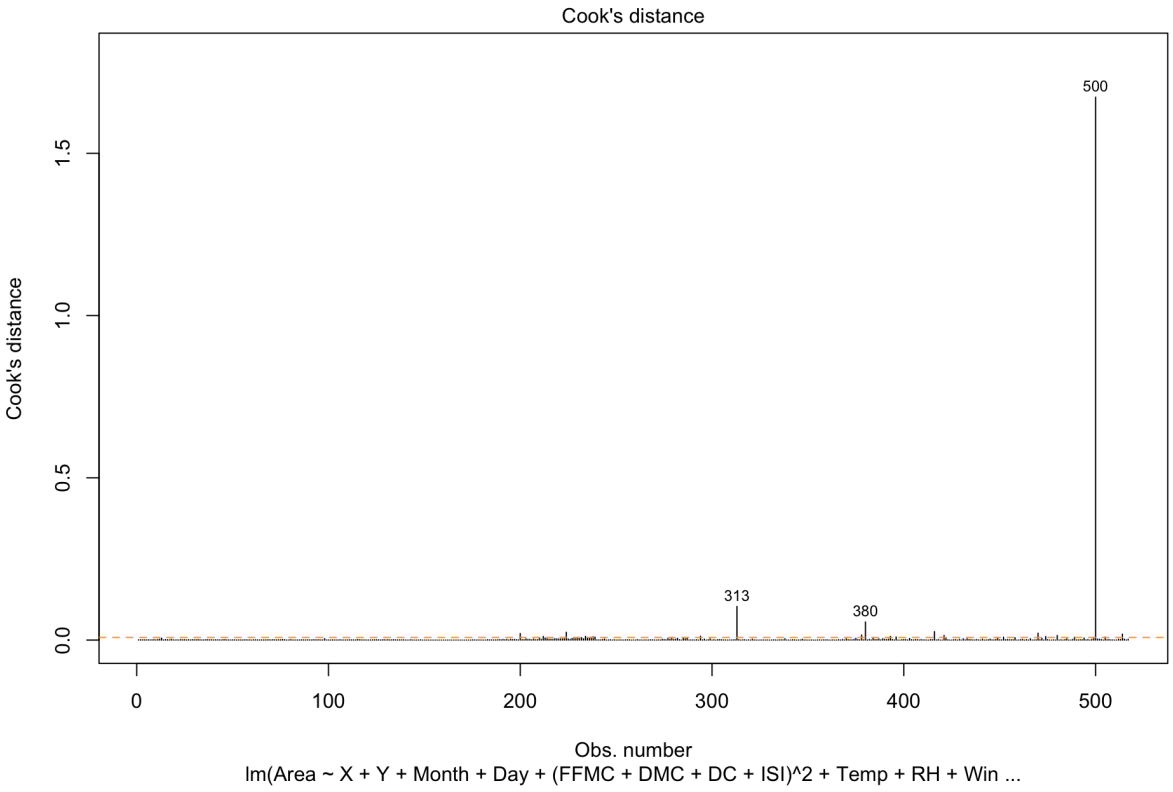
The outlier test shows No 239 is the outlier.

```
# b. High leverage points
hat.plot<-function(mod2)
 {
 co<-length(coefficients(mod2))
 fi<-length(fitted(mod2))
 plot(hatvalues(mod2), main="Index Plot of Hat testues")
 abline(h=c(2,3)*co/fi,col="steelblue", lty=2)
 identify(1:fi,hatvalues(mod2), names(hatvalues(mod2)))
}
hat.plot(mod2)
```




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

```
# c. Influential observation
inf<-4/(nrow(mydata)-length(finalmod$coefficients)-2)
plot(finalmod,which=4,cook.levels=inf)
abline(h=inf,lty=2,col="orange")
```



Interpretation

According to the influence plot, 313,380,416,239 are outliers.

Corrective measures Deleting Observations Outliers are observations that are poorly fit by the regression model. If outliers are determined influential, they can be removed to avoid serious distortions in the regression calculations.This model only has 4 outliers, so they are not significantly influencial to the learning of the model.

Corrective Measures

```
# Deleting the outliers:
mydata <- mydata[-c(313,380,416,239),]
mydata
```

X <dbl>	Y <dbl>	Month <dbl>	Day <dbl>	FFMC <dbl>	DMC <dbl>	DC <dbl>	ISI <dbl>	Temp <dbl>	RH <dbl>
7	5	8	1	86.2	26.2	94.3	5.1	8.2	51
7	4	11	6	90.6	35.4	669.1	6.7	18.0	33
7	4	11	3	90.6	43.7	686.9	6.7	14.6	33
8	6	8	1	91.7	33.3	77.5	9.0	8.3	97
8	6	8	4	89.3	51.3	102.2	9.6	11.4	99
8	6	2	4	92.3	85.3	488.0	14.7	22.2	29
8	6	2	2	92.3	88.9	495.6	8.5	24.1	27
8	6	2	2	91.5	145.4	608.2	10.7	8.0	86
8	6	12	6	91.0	129.5	692.6	7.0	13.1	63
7	5	12	3	92.5	88.0	698.6	7.1	22.8	40

1-10 of 513 rows | 1-10 of 13 columns Previous **1** 2 3 4 5 6 … 52 Next

Select the best regression model

```
# Using anova() function to compare the models
anova(mod2, finalmod)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	504	980.9194	*NA*	*NA*	*NA*	*NA*
2	498	976.6015	6	4.317864	0.3669692	0.8998346

2 rows

- We decide to go ahead with model 2 based on their AIC values and variance table.

```
# Compare the models and select the best model
AIC(mod2,finalmod)
```

	df <dbl>	AIC <dbl>
mod2	14	1826.294
finalmod	20	1836.013

2 rows

Fine tune the selection of predictor variables

Will use the stepwise selection algorithm to find the most significant Factors in our model

```r
library(tidyverse)
library(caret)
library(leaps)
library(MASS)
library(forecast)
#stat: explanatory model/ best-fit purpose
#For multiple regression model (lm)

# Set seed for reproducibility
set.seed(1000)

#get the model with all predictor variables
model <- lm(Area ~.,data = Forest_Fires)
summary(model)
```

```
##
## Call:
## lm(formula = Area ~ ., data = Forest_Fires)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6785 -1.0875 -0.5743  0.8840  5.5821
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1571321  1.3930842  -0.113   0.9102
## X            0.0401684  0.0317946   1.263   0.2070
## Y            0.0136655  0.0600928   0.227   0.8202
## Month        0.0173175  0.0170180   1.018   0.3094
## Day          0.0226868  0.0328330   0.691   0.4899
## FFMC         0.0063487  0.0145561   0.436   0.6629
## DMC          0.0017682  0.0015803   1.119   0.2637
## DC           0.0001277  0.0004120   0.310   0.7567
## ISI         -0.0229718  0.0170871  -1.344   0.1794
## Temp         0.0042540  0.0175928   0.242   0.8090
## RH          -0.0049939  0.0052697  -0.948   0.3438
## Wind         0.0808763  0.0368542   2.194   0.0287 *
## Rain         0.0760935  0.2127150   0.358   0.7207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.395 on 504 degrees of freedom
## Multiple R-squared:  0.02793,    Adjusted R-squared:  0.004783
## F-statistic: 1.207 on 12 and 504 DF,  p-value: 0.2749
```

```r
summary(model)$coefficient
```

```
##                  Estimate   Std. Error    t value   Pr(>|t|)
## (Intercept) -0.1571321423 1.3930841914 -0.1127944 0.91023844
## X            0.0401684145 0.0317945564  1.2633740 0.20703891
## Y            0.0136655495 0.0600928421  0.2274073 0.82019928
## Month        0.0173174895 0.0170180111  1.0175977 0.30935739
## Day          0.0226867989 0.0328329818  0.6909759 0.48989881
## FFMC         0.0063486790 0.0145560723  0.4361533 0.66291227
## DMC          0.0017682334 0.0015803060  1.1189183 0.26370797
## DC           0.0001277343 0.0004119886  0.3100434 0.75665611
## ISI         -0.0229718012 0.0170870944 -1.3443948 0.17942552
## Temp         0.0042539883 0.0175928439  0.2418022 0.80903179
## RH          -0.0049939403 0.0052697465 -0.9476623 0.34375553
## Wind         0.0808762663 0.0368542298  2.1944907 0.02865571
## Rain         0.0760934970 0.2127150329  0.3577251 0.72069884
```

```r
accuracy(model)
```

```
##                        ME     RMSE      MAE MPE MAPE      MASE
## Training set -6.51788e-17 1.377436 1.135532 NaN  Inf 0.9812005
```

```
# Stepwise regression model
step.model1 <- stepAIC(model, direction = "backward", #can change the direction
                       trace = FALSE)
summary(step.model1) #return to the best final model
```

```
##
## Call:
## lm(formula = Area ~ X + Month + DMC + RH + Wind, data = Forest_Fires)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6551 -1.0850 -0.6000  0.9061  5.6014
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5112159  0.3096796   1.651   0.0994 .
## X            0.0433550  0.0265691   1.632   0.1033
## Month        0.0203525  0.0144363   1.410   0.1592
## DMC          0.0020954  0.0009817   2.134   0.0333 *
## RH          -0.0057045  0.0037953  -1.503   0.1334
## Wind         0.0697302  0.0348560   2.001   0.0460 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.389 on 511 degrees of freedom
## Multiple R-squared:  0.02299,    Adjusted R-squared:  0.01343
## F-statistic: 2.405 on 5 and 511 DF,  p-value: 0.0359
```

```
accuracy(step.model1)
```

```
##                         ME     RMSE      MAE  MPE MAPE      MASE
## Training set -1.415721e-16 1.380929 1.142264 -Inf  Inf 0.9870172
```

Since there are 12 variables, 12 models are created and we try to estimate their average prediction error

```
# use 10-fold cross-validation to estimate the average prediction error (RMSE) of each of the 12 m
        odels
# Set seed for reproducibility
set.seed(500)
# Set up repeated k-fold cross-validation
#use the 10-fold cross-validation
train.control<- trainControl(method = "cv", number = 10)
# Train the model
step.model2 <- train(Area ~.,data = Forest_Fires,
                     method = "leapSeq",
                     tuneGrid = data.frame(nvmax = 1:12),
                     trControl = train.control
                     )
step.model2$results
```

nvmax <int>	RMSE <dbl>	Rsquared <dbl>	MAE <dbl>	RMSESD <dbl>	RsquaredSD <dbl>	MAESD <dbl>
1	1	1.416352	0.031051503	1.177217	0.09940090	0.022687956	0.04750525
2	2	1.419357	0.018040834	1.177729	0.09057403	0.020861567	0.04532602
3	3	1.411269	0.008586597	1.170015	0.09686757	0.010013091	0.04677334
4	4	1.414238	0.002036145	1.173862	0.09734008	0.001799743	0.05129877
5	5	1.411829	0.015731845	1.171946	0.08384533	0.023735372	0.05140978
6	6	1.446831	0.009294203	1.184831	0.10995309	0.011923825	0.04613453
7	7	1.410119	0.006094086	1.171204	0.08993166	0.008277717	0.05055387
8	8	1.445523	0.014312428	1.178228	0.11039037	0.019617268	0.05017119

	nvmax <int>	RMSE <dbl>	Rsquared <dbl>	MAE <dbl>	RMSESD <dbl>	RsquaredSD <dbl>	MAESD <dbl>
9	9	1.448517	0.013307094	1.179045	0.11219188	0.020388530	0.05219422
10	10	1.408787	0.015946791	1.169197	0.08535859	0.021786403	0.05367289

1-10 of 12 rows Previous **1** 2 Next

```
 #Fine tune the selection of predictor variables

step.model2$bestTune
```

	nvmax <int>
10	10

1 row

```
 #therefore, the best prediction model with 3 predictor variables

#get the final model
summary(step.model2$finalModel)
```

```
## Subset selection object
## 12 Variables  (and intercept)
##       Forced in Forced out
## X         FALSE      FALSE
## Y         FALSE      FALSE
## Month     FALSE      FALSE
## Day       FALSE      FALSE
## FFMC      FALSE      FALSE
## DMC       FALSE      FALSE
## DC        FALSE      FALSE
## ISI       FALSE      FALSE
## Temp      FALSE      FALSE
## RH        FALSE      FALSE
## Wind      FALSE      FALSE
## Rain      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: 'sequential replacement'
##           X   Y   Month Day FFMC DMC DC  ISI Temp RH  Wind Rain
## 1  ( 1 )  " " " " " "   " " " "  "*" " " " " " "  " " " "  " "
## 2  ( 1 )  " " " " " "   " " " "  " " "*" " " " "  " " "*"  " "
## 3  ( 1 )  "*" " " " "   " " " "  " " "*" " " " "  " " "*"  " "
## 4  ( 1 )  "*" " " " "   " " " "  " " "*" " " " "  "*" "*"  " "
## 5  ( 1 )  "*" " " "*"   " " " "  "*" " " " " " "  "*" "*"  " "
## 6  ( 1 )  "*" " " "*"   " " " "  "*" " " "*" " "  "*" "*"  " "
## 7  ( 1 )  "*" " " "*"   "*" " "  "*" " " "*" " "  "*" "*"  " "
## 8  ( 1 )  "*" " " "*"   "*" "*"  "*" " " "*" " "  "*" "*"  " "
## 9  ( 1 )  "*" " " "*"   "*" "*"  "*" " " "*" " "  "*" "*"  "*"
## 10  ( 1 ) "*" "*" "*"   "*" "*"  "*" "*" "*" "*"  "*" " "  " "
```

```
modelaaa <- lm(formula = Area ~ X + DC + Wind, data = Forest_Fires)
summary(modelaaa)
```

```
##
## Call:
## lm(formula = Area ~ X + DC + Wind, data = Forest_Fires)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5183 -1.0823 -0.6516  0.9281  5.7673
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3801571  0.2622505   1.450   0.1478
## X           0.0411523  0.0265718   1.549   0.1221
## DC          0.0005033  0.0002531   1.988   0.0473 *
## Wind        0.0654530  0.0349190   1.874   0.0614 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.391 on 513 degrees of freedom
## Multiple R-squared:  0.01576,    Adjusted R-squared:  0.01001
## F-statistic: 2.738 on 3 and 513 DF,  p-value: 0.04286
```

```
accuracy(modelaaa)
```

```
##                         ME     RMSE      MAE  MPE MAPE     MASE
## Training set -1.135497e-17 1.386029 1.143796 -Inf  Inf 0.988341
```

```
#the second order linear model
library(forecast)
model2 <- lm(Area ~ X+Y+Month + Day + (FFMC + DMC + DC + ISI )^2+ Temp + RH + Wind+Rain,data=Fores
        t_Fires)
summary(model2)
```

```
##
## Call:
## lm(formula = Area ~ X + Y + Month + Day + (FFMC + DMC + DC +
##     ISI)^2 + Temp + RH + Wind + Rain, data = Forest_Fires)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8554 -1.0953 -0.5259  0.8757  5.5569
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.073e-02  2.099e+00  -0.015    0.988
## X            4.300e-02  3.208e-02   1.340    0.181
## Y            6.430e-03  6.060e-02   0.106    0.916
## Month        2.602e-02  1.841e-02   1.414    0.158
## Day          2.042e-02  3.327e-02   0.614    0.540
## FFMC         5.621e-03  2.464e-02   0.228    0.820
## DMC         -3.432e-02  4.329e-02  -0.793    0.428
## DC           2.919e-03  6.460e-03   0.452    0.652
## ISI          9.461e-02  5.701e-01   0.166    0.868
## Temp         4.503e-04  1.927e-02   0.023    0.981
## RH          -5.048e-03  5.397e-03  -0.935    0.350
## Wind         8.301e-02  3.791e-02   2.190    0.029 *
## Rain         4.652e-02  2.165e-01   0.215    0.830
## FFMC:DMC     4.064e-04  4.860e-04   0.836    0.403
## FFMC:DC     -2.811e-05  7.534e-05  -0.373    0.709
## FFMC:ISI    -1.539e-03  6.045e-03  -0.255    0.799
## DMC:DC      -3.722e-06  6.773e-06  -0.549    0.583
## DMC:ISI      2.334e-04  4.263e-04   0.547    0.584
## DC:ISI      -1.048e-05  8.478e-05  -0.124    0.902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.4 on 498 degrees of freedom
## Multiple R-squared:  0.03221,    Adjusted R-squared:  -0.002774
## F-statistic: 0.9207 on 18 and 498 DF,  p-value: 0.5533
```

```
accuracy(model2)
```

```
##                        ME     RMSE      MAE MPE MAPE     MASE
## Training set 9.701261e-17 1.374401 1.132069 NaN  Inf 0.978208
```

```
# Stepwise regression model
step.model1 <- stepAIC(model2, direction = "backward",
                       trace = FALSE)
summary(step.model1) #return to the best final model
```

```
##
## Call:
## lm(formula = Area ~ X + Month + DMC + RH + Wind, data = Forest_Fires)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6551 -1.0850 -0.6000  0.9061  5.6014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5112159  0.3096796   1.651   0.0994 .
## X            0.0433550  0.0265691   1.632   0.1033
## Month        0.0203525  0.0144363   1.410   0.1592
## DMC          0.0020954  0.0009817   2.134   0.0333 *
## RH          -0.0057045  0.0037953  -1.503   0.1334
## Wind         0.0697302  0.0348560   2.001   0.0460 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.389 on 511 degrees of freedom
## Multiple R-squared:  0.02299,    Adjusted R-squared:  0.01343
## F-statistic: 2.405 on 5 and 511 DF,  p-value: 0.0359
```

```
accuracy(step.model1)
```

```
##                         ME     RMSE      MAE  MPE MAPE      MASE
## Training set -1.415721e-16 1.380929 1.142264 -Inf  Inf 0.9870172
```

After fine tuning processes alongwith the 10 fold cross validation it has been confirmed that the model is Area = 0.511216 + X(0.043355) + Month(0.020352) + DMC(0.002095) + RH(-0.005705) + Wind(0.069730) with RMSE as 1.38.

```
set.seed(500)
# Set up repeated k-fold cross-validation
#use the 10-fold cross-validation
train.control<- trainControl(method = "cv", number = 10)
# Train the model
step.model2 <- train(Area ~ X+Y+Month + Day + (FFMC + DMC + DC + ISI)^2 + Temp + RH + Wind+Rain,da
        ta = Forest_Fires,
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax = 1:9),
                    trControl = train.control
                    )
step.model2$results
```

nvmax <int>	RMSE <dbl>	Rsquared <dbl>	MAE <dbl>	RMSESD <dbl>	RsquaredSD <dbl>	MAESD <dbl>
1	1.416485	0.030808511	1.177423	0.09947893	0.02268430	0.04752185
2	2	1.417732	0.012367382	1.175511	0.09286047	0.01449218	0.04920982
3	3	1.413391	0.014797138	1.171767	0.09227325	0.01850312	0.05271350
4	4	1.416668	0.003874009	1.172985	0.09241848	0.00685748	0.04758234
5	5	1.420836	0.007427436	1.176667	0.08286948	0.01517676	0.04388768
6	6	1.454785	0.014900800	1.185071	0.10780818	0.02173738	0.03673318
7	7	1.454901	0.015292097	1.184359	0.10680542	0.01980581	0.03720874
8	8	1.460236	0.015961465	1.189092	0.11526062	0.02214489	0.04318716
9	9	1.459634	0.016800670	1.184949	0.12177216	0.01947608	0.04775744

9 rows

```
 #Fine tune the selection of predictor variables
step.model2$bestTune
```

	nvmax <int>
3	3

1 row

```
 #therefore, the best prediction model with 3 predictor variables

#get the final model
summary(step.model2$finalModel)
```

```
## Subset selection object
## 18 Variables  (and intercept)
##           Forced in Forced out
## X             FALSE      FALSE
## Y             FALSE      FALSE
## Month         FALSE      FALSE
## Day           FALSE      FALSE
## FFMC          FALSE      FALSE
## DMC           FALSE      FALSE
## DC            FALSE      FALSE
## ISI           FALSE      FALSE
## Temp          FALSE      FALSE
## RH            FALSE      FALSE
## Wind          FALSE      FALSE
## Rain          FALSE      FALSE
## FFMC:DMC      FALSE      FALSE
## FFMC:DC       FALSE      FALSE
## FFMC:ISI      FALSE      FALSE
## DMC:DC        FALSE      FALSE
## DMC:ISI       FALSE      FALSE
## DC:ISI        FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: 'sequential replacement'
##          X   Y   Month Day FFMC DMC DC  ISI Temp RH  Wind Rain FFMC:DMC FFMC:DC
## 1  ( 1 ) " " " " " " " "   " " " " " " " " " " " " " " " "  " "       " "
## 2  ( 1 ) " " " " " " " "   " " " " " " " " " " " " "*" " "  " "       "*"
## 3  ( 1 ) "*" " " " " " "   " " " " " " " " " " " " "*" " "  " "       "*"
##          FFMC:ISI DMC:DC DMC:ISI DC:ISI
## 1  ( 1 ) " "      "*"     " "      " "
## 2  ( 1 ) " "      " "     " "      " "
## 3  ( 1 ) " "      " "     " "      " "
```

Interpret the prediction results

The step AIC is used to choose a model by AIC in a stepwise algorithm and fine tunes the model. With each step, a step wise selected model is returned. In the 10 fold cross validation process, our best and final model has three predictor variables namely X: x-axis spatial coordinate within the Montesinho park map: 1 to 9, DC: DC index from the FWI system: 7.9 to 860.6 and Wind. The RMSE is eventually reduced.