# Regression and diagnostics

*Sai Raghuram Kothapalli*

*October 30, 2018*

## Import important libraries

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

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

```r
library(gvlma)
```

## Problem statement - Perform Regression and further analysis on the given dataset

### Tasks to do are -

1. Scatter plot and find the initial parameters
2. Build a few potential regression models
3. Perform regression diagnostics using both typical and enhanced approach
4. Find unusual observations and corrective measure to fix those
5. Find best regression model

## Concrete Slump Test Data

### Import the data

```r
cstd <-  readxl::read_excel("Concrete Slump Test Data.xlsx")
head(cstd)
```

```
## # A tibble: 6 x 11
##      No Cement  Slag `Fly Ash` Water    SP `Coarse Aggrega~
##   <dbl>  <dbl> <dbl>     <dbl> <dbl> <dbl>            <dbl>
## 1     1    273    82       105   210     9              904
## 2     2    163   149       191   180    12              843
## 3     3    162   148       191   179    16              840
## 4     4    162   148       190   179    19              838
## 5     5    154   112       144   220    10              923
## 6     6    147    89       115   202     9              860
## # ... with 4 more variables: `Fine Aggregate` <dbl>, Slump <dbl>, `Slump
## #   Flow` <dbl>, `28-day Compressive Strength` <dbl>
```

**Task 1**

As given in the data description, we will divide the data into predictors and responses

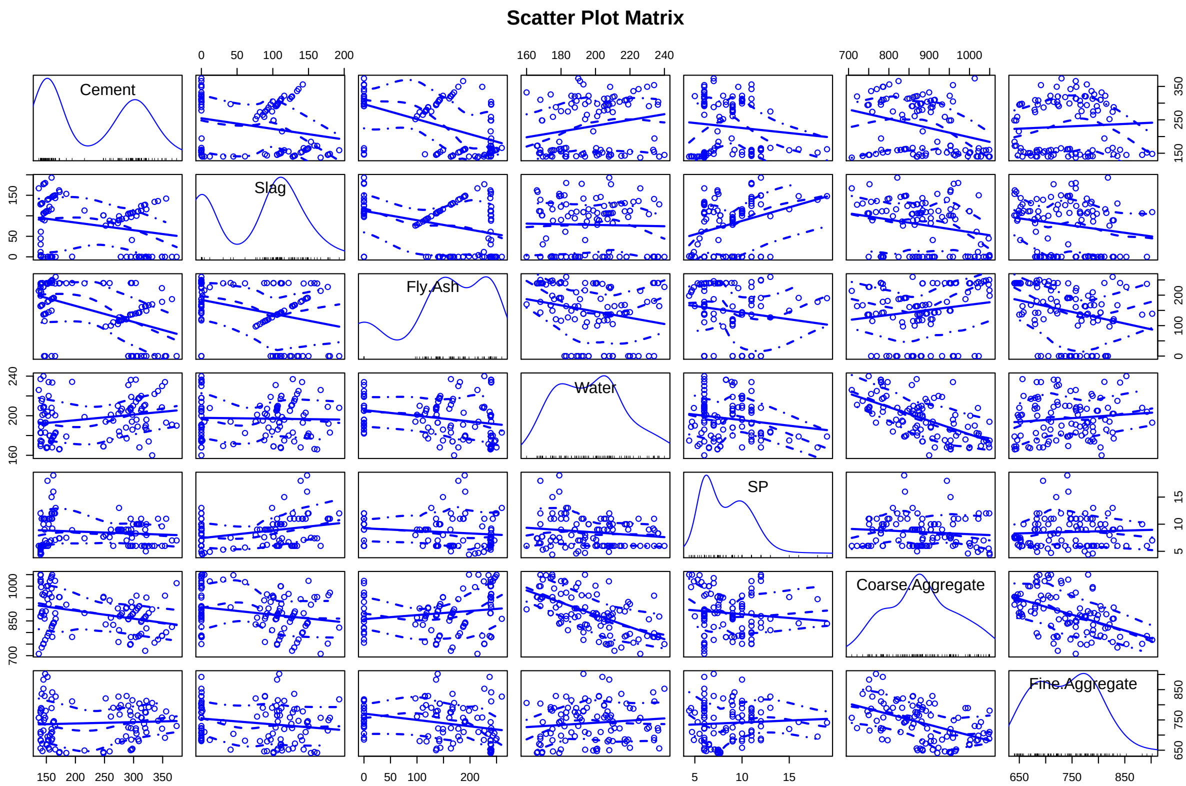
```
predictors <- as.data.frame(cstd[, c("Cement", "Slag", "Fly Ash", "Water", "SP","Coarse Aggregate", "Fi
head(predictors)
```

```
##   Cement Slag Fly Ash Water SP Coarse Aggregate Fine Aggregate
## 1    273   82     105   210  9              904            680
## 2    163  149     191   180 12              843            746
## 3    162  148     191   179 16              840            743
## 4    162  148     190   179 19              838            741
## 5    154  112     144   220 10              923            658
## 6    147   89     115   202  9              860            829
```

```
responses <- as.data.frame(cstd[, c("Slump", "Slump Flow", "28-day Compressive Strength")])
head(responses)
```

```
##   Slump Slump Flow 28-day Compressive Strength
## 1    23       62.0                       34.99
## 2     0       20.0                       41.14
## 3     1       20.0                       41.81
## 4     3       21.5                       42.08
## 5    20       64.0                       26.82
## 6    23       55.0                       25.21
```

```
## the scatter plot for the predictors
scatterplotMatrix(predictors, main = "Scatter Plot Matrix")
```




There are two ways to look into this graph

1. The diagonal graphs represent the distribution of the respective predictor columns. For example, Cement seems to have bi-modal graph whereas Coarse aggregate is normal is nature.
2. Relationship with respect to each other. For example, with increase in Fly Ash, the Cement value decreases and so on.

### Task 2

Let us divide this into 4 types of regression models

1. Simple Linear
2. Polynomial
3. Multi Linear
4. Multi Linear with interactions

I will provide examples of each of the following, the rest of the cases are assumed.

1. simple linear regression Let us use Slump as our response variable for our analysis.

```
fit1 <- lm(responses$Slump ~ predictors$Water)
summary(fit1)
```

```
##
## Call:
## lm(formula = responses$Slump ~ predictors$Water)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.843  -3.535   2.359   6.240  10.678
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -21.78737    7.55330  -2.884  0.00479 **
## predictors$Water   0.20204    0.03811   5.301 6.78e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.778 on 101 degrees of freedom
## Multiple R-squared:  0.2177, Adjusted R-squared:  0.2099
## F-statistic:  28.1 on 1 and 101 DF,  p-value: 6.784e-07
```

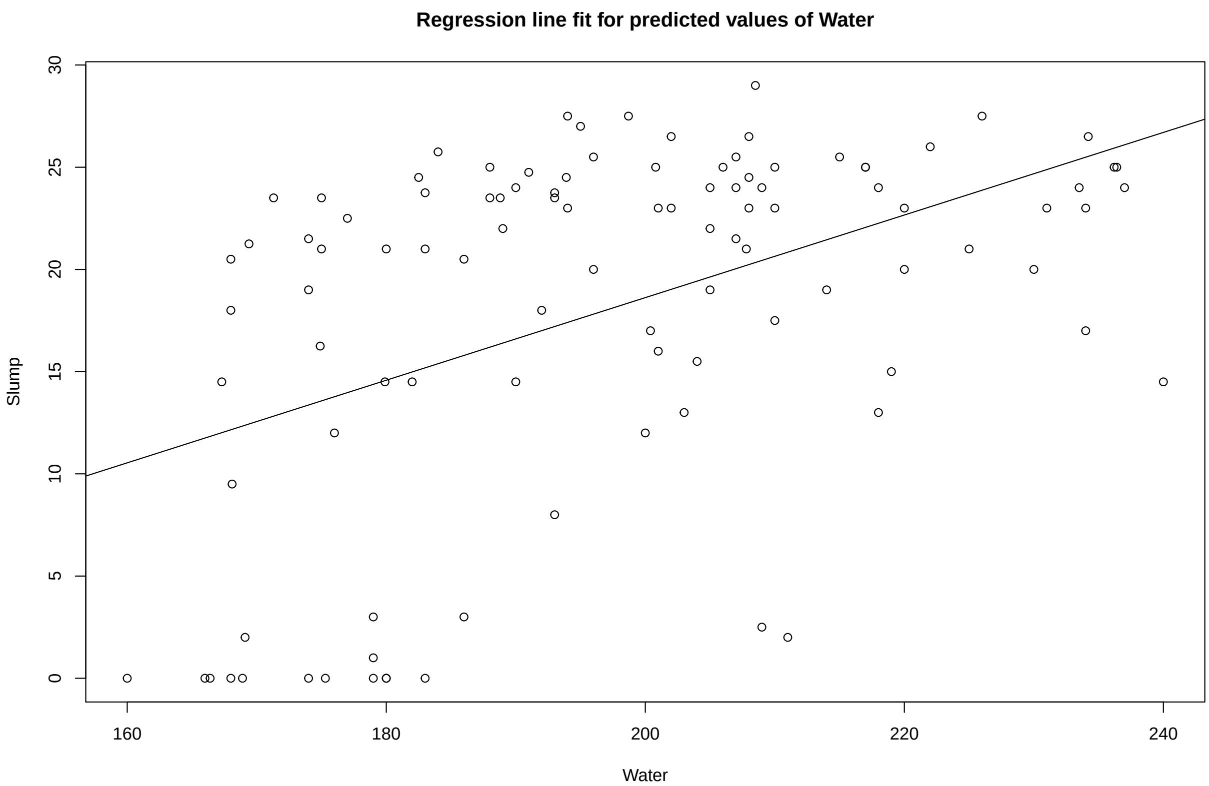
Let us make a dataframe of the fitted values with the original values.

```
fit_water <- fitted(fit1)
or_water <- cstd$Water
residuals <- residuals(fit1)
compare <- data.frame(fit_water, or_water, residuals)
head(compare)
```

```
##   fit_water or_water  residuals
## 1  20.64114      210   2.358865
## 2  14.57992      180 -14.579920
## 3  14.37788      179 -13.377880
## 4  14.37788      179 -11.377880
## 5  22.66154      220  -2.661540
## 6  19.02481      202   3.975189
```

Let's see if the line fits or not.

```
plot( predictors$Water, responses$Slump, xlab = "Water", ylab = "Slump", main = "Regression line fit fo
abline(fit1)
```




2. Polynomial Regression

Let us use Water again for the sake of simplicity. Let us use Water + Water^2 as our predictor.

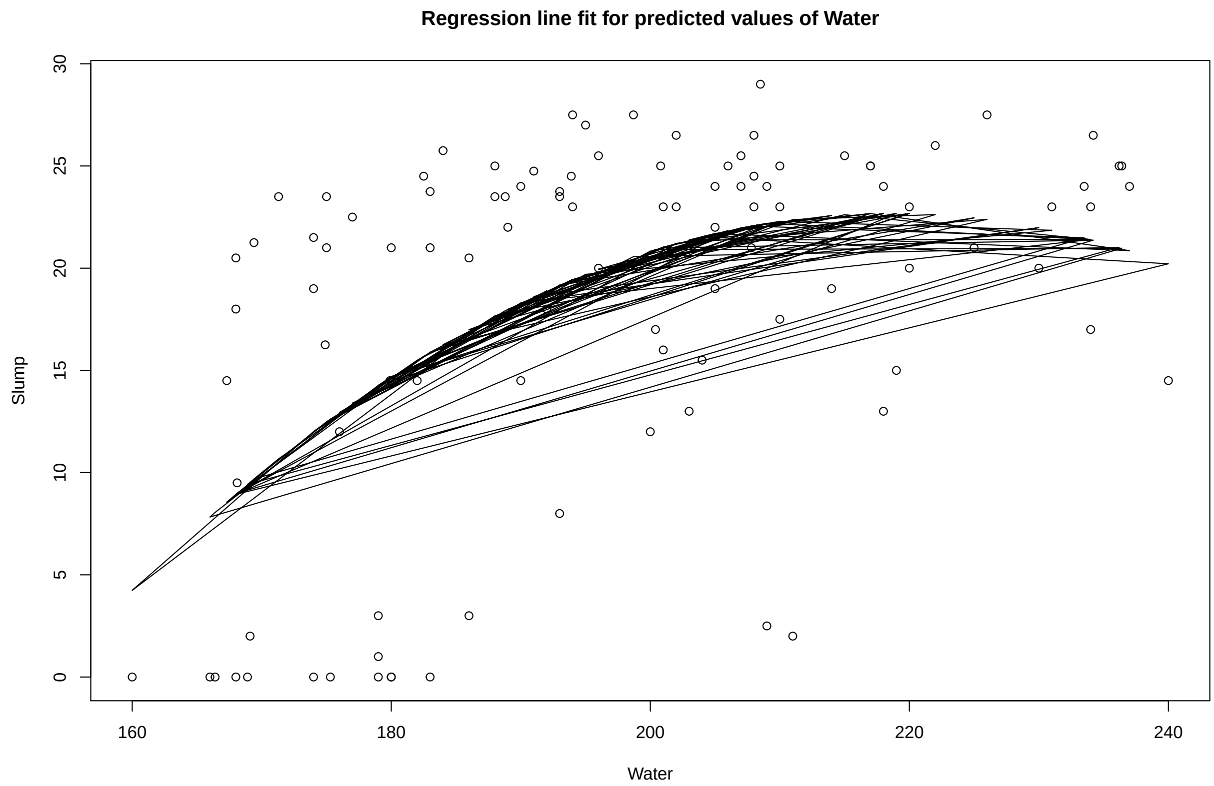
```r
fit2 <- lm(responses$Slump ~ predictors$Water + I(predictors$Water^2))
summary(fit2)
```

```
##
## Call:
## lm(formula = responses$Slump ~ predictors$Water + I(predictors$Water^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.377  -4.323   1.974   5.123  12.828
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.342e+02  6.823e+01  -3.432 0.000872 ***
## predictors$Water       2.350e+00  6.872e-01   3.420 0.000907 ***
## I(predictors$Water^2) -5.377e-03  1.718e-03  -3.131 0.002287 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.46 on 100 degrees of freedom
## Multiple R-squared:  0.2875, Adjusted R-squared:  0.2733
## F-statistic: 20.18 on 2 and 100 DF,  p-value: 4.353e-08
```

We observe that the R2 vaue, which explains the variance of the data has also increased, which suggests that this is a bit better better model.

Similarly, let us plot it.

```
plot( predictors$Water, responses$Slump, xlab = "Water", ylab = "Slump", main = "Regression line fit fo
lines(predictors$Water, fitted(fit2))
```




```
# lines((fit2))
```

These are the regression lines for the polynomial function which we gave as predictors in the formula.

3, Multi Linear Regression

Let us take in 4 variables

```
fit3 <- lm(responses$Slump ~ predictors$Water + predictors$`Fly Ash` + predictors$`Cement` + predictors$
summary(fit3)
```

```
##
## Call:
## lm(formula = responses$Slump ~ predictors$Water + predictors$`Fly Ash` +
##     predictors$Cement + predictors$Slag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.583  -6.283   2.055   5.218  12.652
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -8.68790    9.00067  -0.965 0.336795
## predictors$Water        0.19015    0.03770   5.043  2.1e-06 ***
## predictors$`Fly Ash` -0.02044    0.01166  -1.753 0.082701 .
## predictors$Cement      -0.01536    0.01218  -1.260 0.210494
## predictors$Slag        -0.05360    0.01466  -3.657 0.000413 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.397 on 98 degrees of freedom
## Multiple R-squared:  0.3135, Adjusted R-squared:  0.2855
## F-statistic: 11.19 on 4 and 98 DF,  p-value: 1.615e-07
```

4. Multi Linear with interactions

Let us consider a mixture of fine aggregate and water because of their relation from the scatter plot

```
fit4 <- lm(responses$Slump ~ predictors$Water + predictors$`Fine Aggregate` +  predictors$Water : predi
summary(fit4)
```

```
##
## Call:
## lm(formula = responses$Slump ~ predictors$Water + predictors$`Fine Aggregate` +
##     predictors$Water:predictors$`Fine Aggregate`)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.651  -4.442   2.217   5.386  12.030
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                -1.814e+02  8.590e+01  -2.112
## predictors$Water                            9.330e-01  4.336e-01   2.152
## predictors$`Fine Aggregate`                 2.202e-01  1.172e-01   1.878
## predictors$Water:predictors$`Fine Aggregate` -1.010e-03  5.906e-04  -1.710
##                                            Pr(>|t|)
## (Intercept)                                  0.0372 *
## predictors$Water                             0.0338 *
## predictors$`Fine Aggregate`                  0.0633 .
## predictors$Water:predictors$`Fine Aggregate`   0.0905 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.631 on 99 degrees of freedom
## Multiple R-squared:  0.2619, Adjusted R-squared:  0.2395
## F-statistic: 11.71 on 3 and 99 DF,  p-value: 1.245e-06
```

**Task 3**

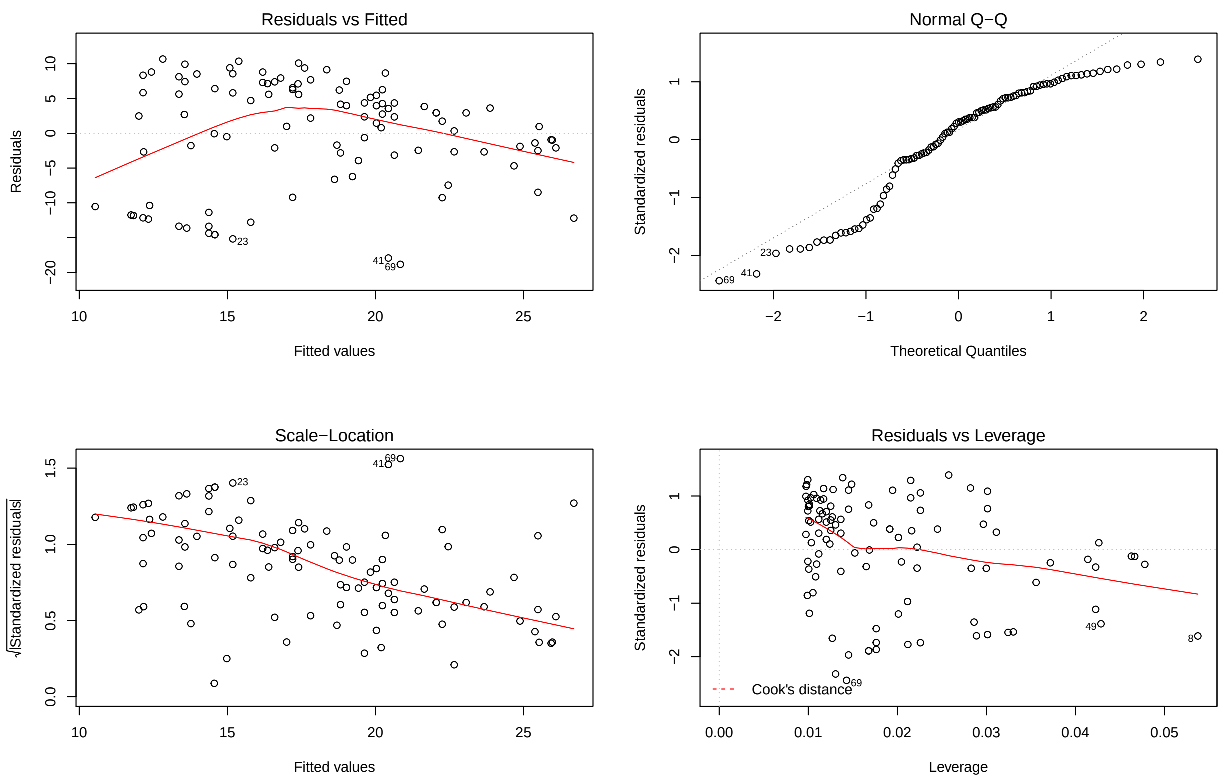
Regression diagnostics are done basically to see if the model is not violating the linearity and normality assumptions.

### Typical Approach

The most common method is to apply plot() to the object returned by the lm(). Doing so, produces 4 graphs used for evaluating the model fit.

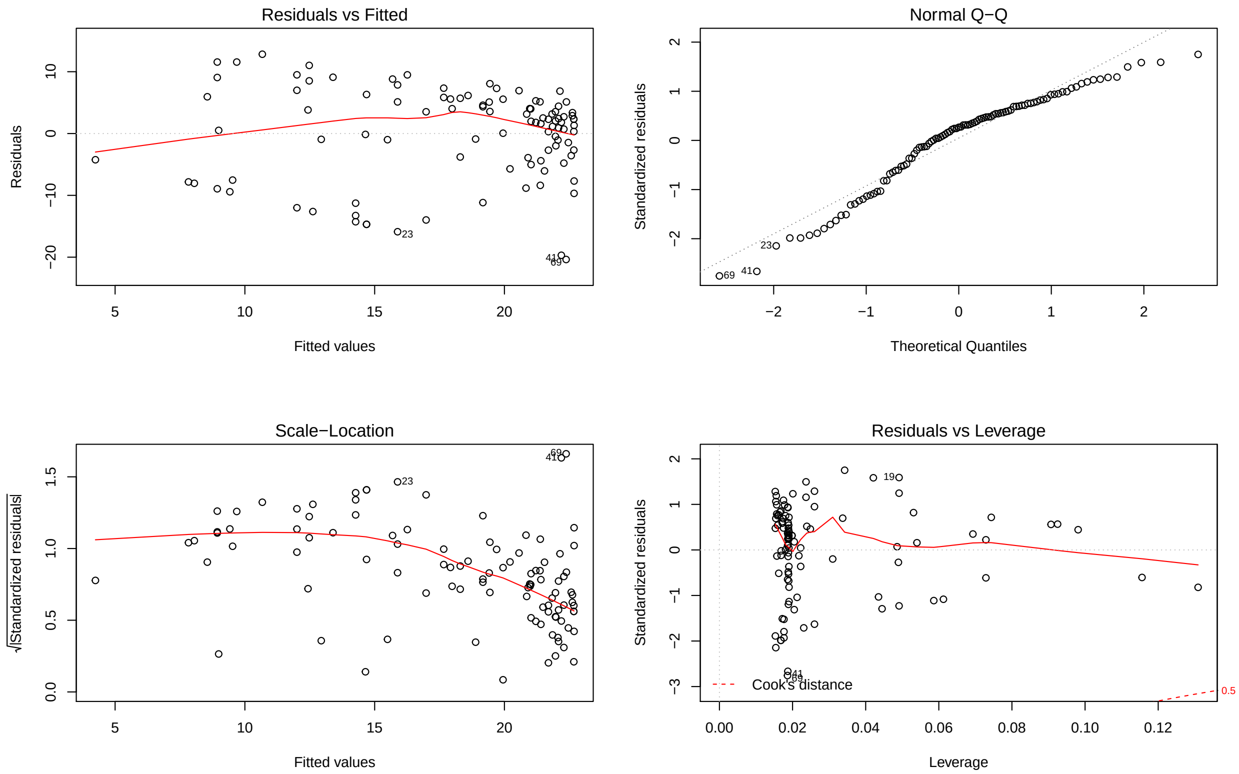
```r
# Using fit1
par(mfrow=c(2,2))

plot(fit1)
```




```r
# Using fit1
par(mfrow=c(2,2))

plot(fit2)
```

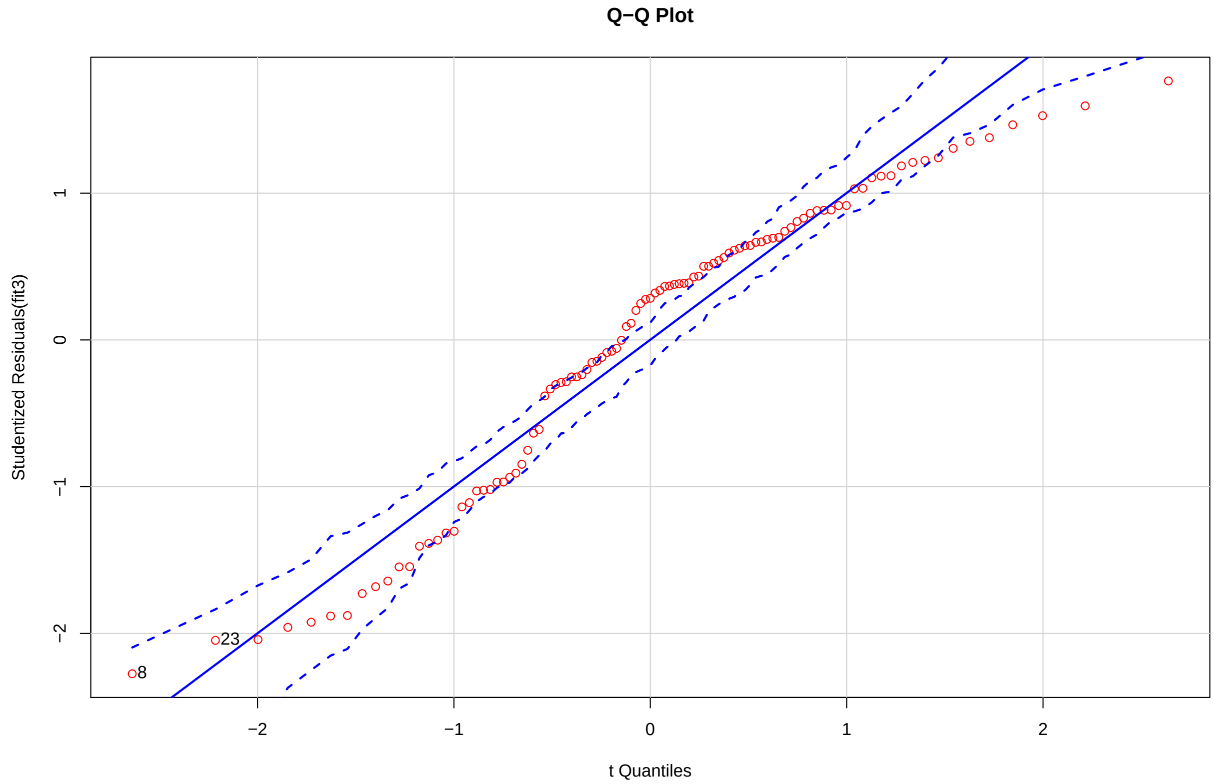



So we see that the polynomial regression model seems to go hand in hand with the normality and linearity assumptions, better than the simple linear model.

### Enhanced Approach

1. **qqplot()** is a starting approach for the **normality test** and is more accurate than the plot() we used earlier.

```
## using fit3
## for some reason, my id.method is not working
qqPlot(fit3, id.method = "identify", labels = row.names(cstd), main = "Q-Q Plot", col = "red")
```




```
## [1]  8 23
```

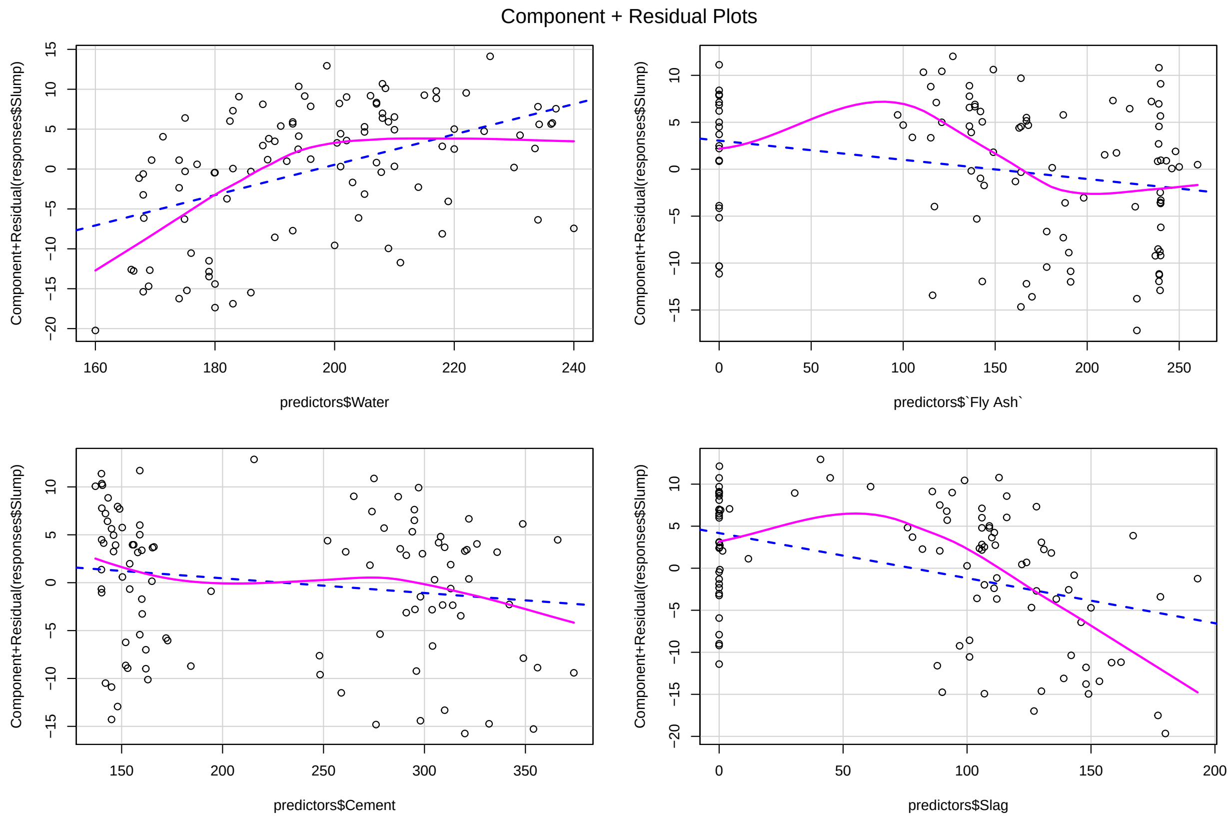
```
## id.method makes the graph interactive to hover over
```

2. Linearity Test

We can look at the non-linearity between the dependent and independent variables by looking at the **components plus residual plots**, generated by **crPlots()** in the car() package.

```
## Using fit3 again

crPlots(fit3)
```




we can see that the variables are meeting the expectations except Fly Ash, which is behaving a bit weird. But overall, yes they do.

3. Let us take the ultimate Global Validation test

This is an ultimate test generated by the function gvlma().

```
library(gvlma)
gvltest <- gvlma(fit3)
summary(gvltest)
```

```
##
## Call:
## lm(formula = responses$Slump ~ predictors$Water + predictors$`Fly Ash` +
##     predictors$Cement + predictors$Slag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.583  -6.283   2.055   5.218  12.652
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -8.68790    9.00067  -0.965 0.336795
## predictors$Water        0.19015    0.03770   5.043  2.1e-06 ***
## predictors$`Fly Ash` -0.02044    0.01166  -1.753 0.082701 .
## predictors$Cement      -0.01536    0.01218  -1.260 0.210494
```

```
## predictors$Slag      -0.05360    0.01466  -3.657 0.000413 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.397 on 98 degrees of freedom
## Multiple R-squared:  0.3135, Adjusted R-squared:  0.2855
## F-statistic: 11.19 on 4 and 98 DF,  p-value: 1.615e-07
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit3)
##
##                     Value   p-value                   Decision
## Global Stat        41.122 2.536e-08 Assumptions NOT satisfied!
## Skewness            3.813 5.085e-02    Assumptions acceptable.
## Kurtosis            2.949 8.594e-02    Assumptions acceptable.
## Link Function      33.021 9.118e-09 Assumptions NOT satisfied!
## Heteroscedasticity  1.339 2.471e-01    Assumptions acceptable.
```

The p-values are less than 0.05 and that is the reason they are not acceptable. We need to look back into our assumptions strategy.

#### Task 4

Screening for unusual observations meaning the outliers or the high-leverage observations

1. Outliers

```
outlierTest(fit3)
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 8 -2.274672           0.025128           NA
```

This indicates that Number 8 is the outlier. This function always results in single value of outliers and the measures to cure this is to delete them from the dataset and check for the test again.

2. High-leverage observations

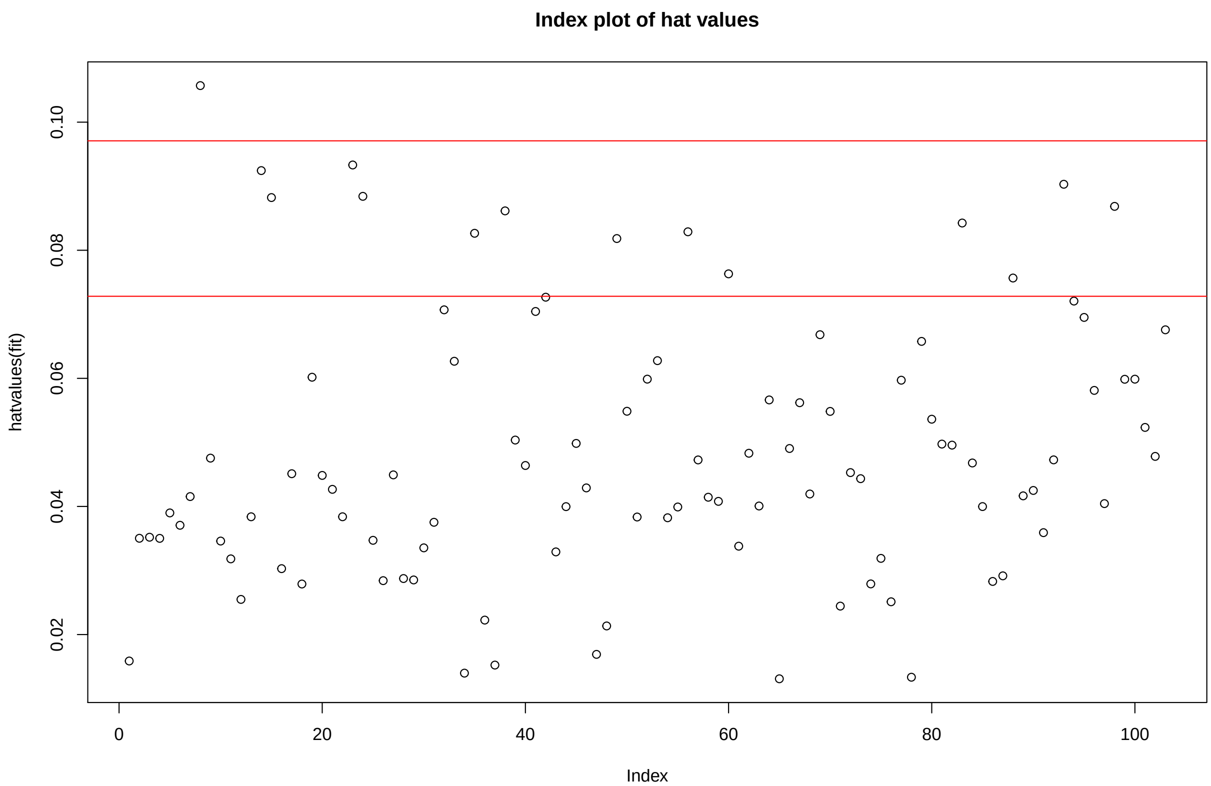
Observations that have high leverage are the outliers with regards to other predictors, meaning they have an unusual combination of predictor values. The response value is not involved in determining the leverage.

They are identified through the **hat-statistic**.

```r
hat.plot <- function(fit) {
  p <- length(coefficients(fit))
  n <- length(fitted(fit))
  plot(hatvalues(fit), main = "Index plot of hat values")
  abline(h = c(1.5,2)* p/n, col= "red")
  identify(1:n, hatvalues(fit), names(hatvalues(fit)))

}

hat.plot(fit3)
```




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

Therefore the values above 1.5 or 2 time the average hat value are to be examined to have high leverage.

Corrective measures are a. Deletion Deletion of the outliers is the traditional way to have corrective measures on the analysis we did till now. This improves dataset's fit to normality assumption. Influential observations are deleted as well because they have inordinate impact on results. This always is not a good practice.

b. Transforming variables

The lambda value in the figure can be evaluated using **powerTransform()** and **boxTidwell()** .

```r
# summary(powerTransform(cstd$Slump))

# boxTidwell(responses$Slump ~ predictors$`Fly Ash` + predictors$Cement)
```

Also, by doing these transformations, it is not ecessary that they are needed in the first place.

**Table 8.5 Common transformations**

| | −2 | −1 | −0.5 | 0 | 0.5 | 1 | 2 |
|---|---|---|---|---|---|---|---|
| Transformation | $1/Y^2$ | $1/Y$ | $1/\sqrt{Y}$ | log(Y) | $\sqrt{Y}$ | None | $Y^2$ |

Figure 1: Common transformations possible

**Task 5**

Selection of best models can be done using Comparison of models

a. By using Analysis of variance (ANOVA)

```r
## Using fit3 and fit4
anova(fit3, fit4)
```

```
## Analysis of Variance Table
##
## Model 1: responses$Slump ~ predictors$Water + predictors$`Fly Ash` + predictors$Cement +
##     predictors$Slag
## Model 2: responses$Slump ~ predictors$Water + predictors$`Fine Aggregate` +
##     predictors$Water:predictors$`Fine Aggregate`
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     98 5361.9
## 2     99 5765.1 -1   -403.22 7.3697 0.007839 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```