# Multivariate Regression Analysis - MA4142

Group - 8

Kethari Narasimha Vardhan - MA20BTECH11006
Varunaditya Singhal - MA20BTECH11021

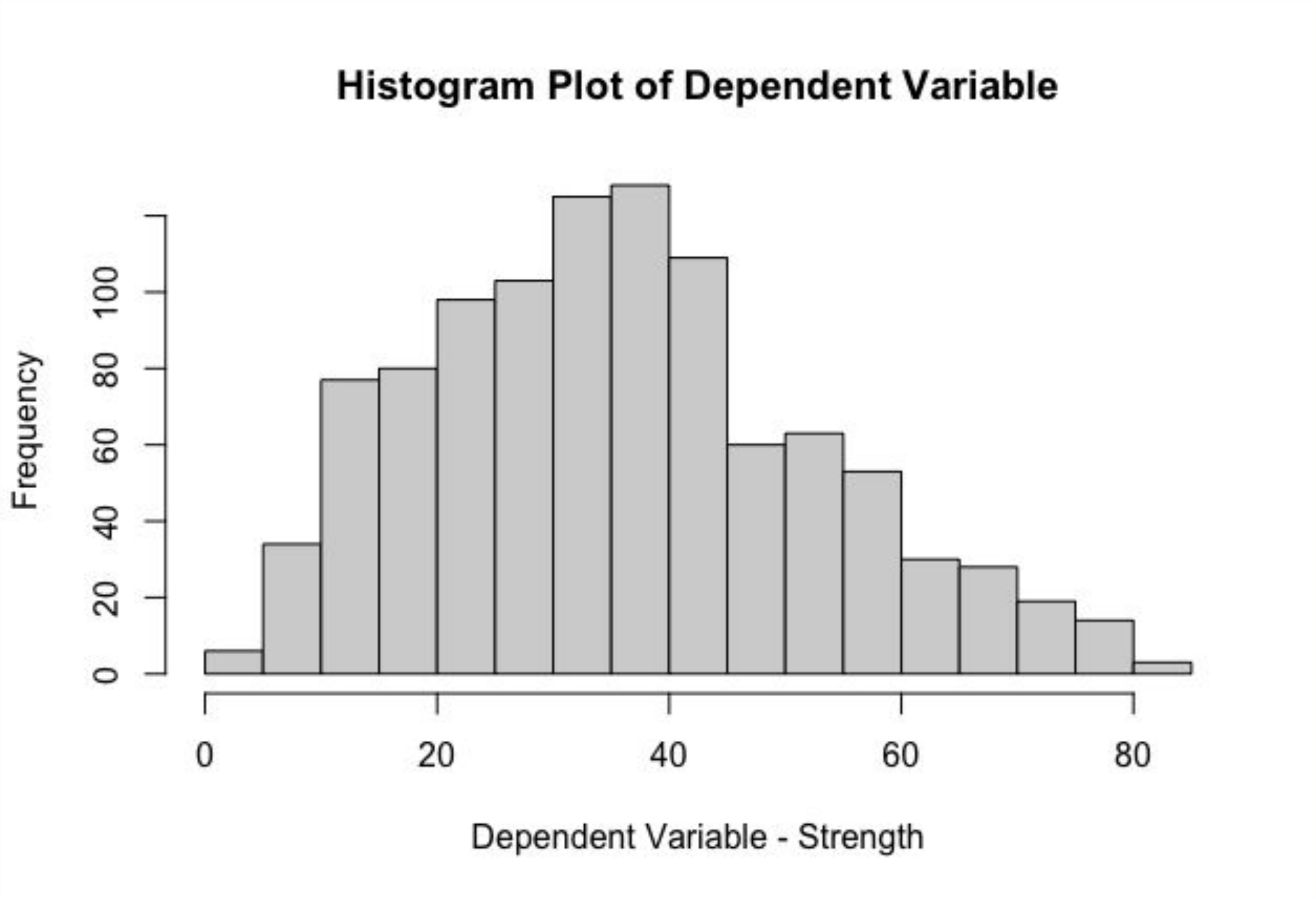
# Dataset :

```
> # Summary statistics
> summary(data)
     Cement           Slag           FlyAsh           Water         Plasticizer       CoarseAgg         FineAgg           Age            Strength
 Min.   :102.0   Min.   :  0.0   Min.   :  0.00   Min.   :121.8   Min.   : 0.000   Min.   : 801.0   Min.   :594.0   Min.   :  1.00   Min.   : 2.332
 1st Qu.:192.4   1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.:164.9   1st Qu.: 0.000   1st Qu.: 932.0   1st Qu.:731.0   1st Qu.:  7.00   1st Qu.:23.707
 Median :272.9   Median : 22.0   Median :  0.00   Median :185.0   Median : 6.350   Median : 968.0   Median :779.5   Median : 28.00   Median :34.443
 Mean   :281.2   Mean   : 73.9   Mean   : 54.19   Mean   :181.6   Mean   : 6.203   Mean   : 972.9   Mean   :773.6   Mean   : 45.66   Mean   :35.818
 3rd Qu.:350.0   3rd Qu.:142.9   3rd Qu.:118.27   3rd Qu.:192.0   3rd Qu.:10.160   3rd Qu.:1029.4   3rd Qu.:824.0   3rd Qu.: 56.00   3rd Qu.:46.136
 Max.   :540.0   Max.   :359.4   Max.   :200.10   Max.   :247.0   Max.   :32.200   Max.   :1145.0   Max.   :992.6   Max.   :365.00   Max.   :82.599
```

Attributes - 9 , Instances - 1030

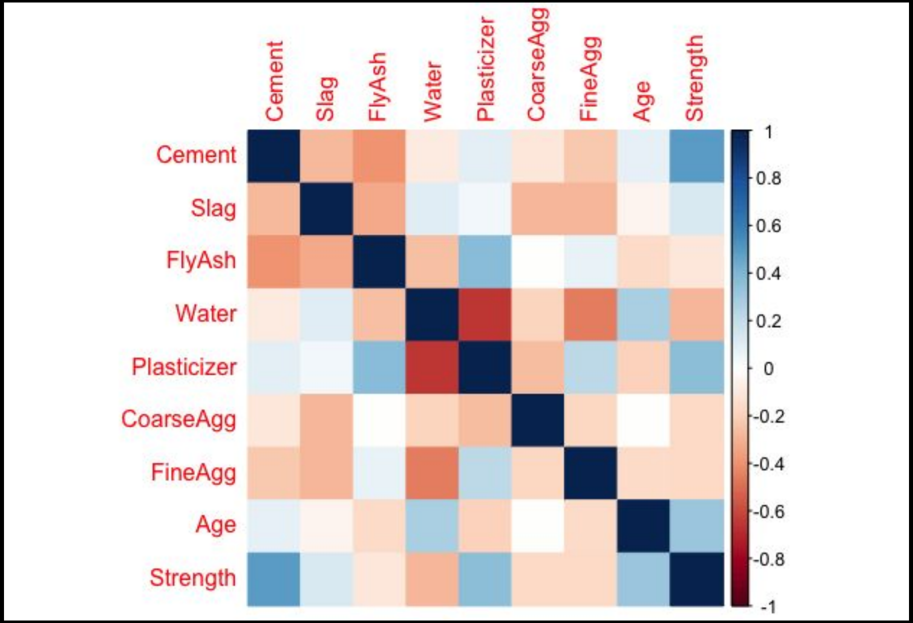
70% of the data is used for training and 30% for testing.

Link for the data



Histogram Plot of Dependent Variable
Frequency
0
20
40
60
80
100
0
20
40
60
80
Dependent Variable - Strength

# Correlation Matrix



# Packages/Libraries Used

car, readxl, corrplot, lmtest

# Regression Model

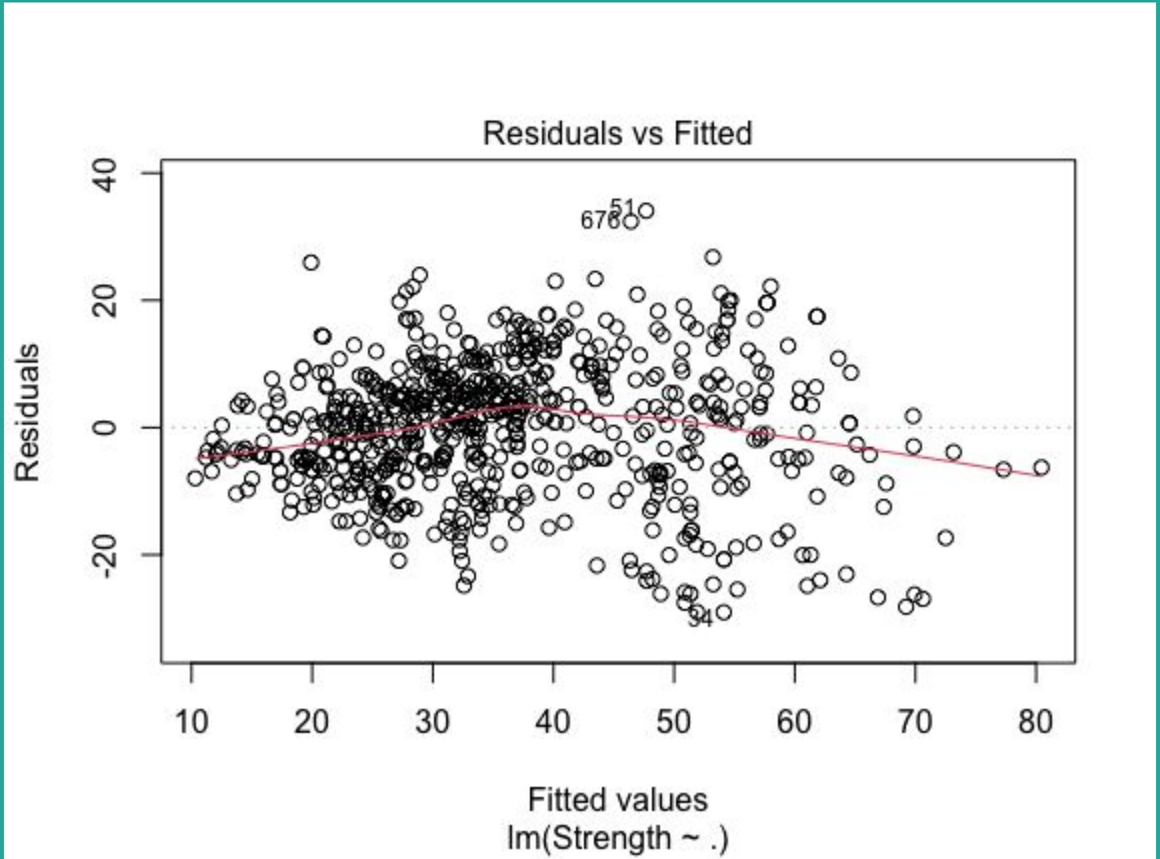
(First)

```
> # Fitting linear regression model on the train data-set
> model <- lm(Strength ~ ., data = train_data)
> # Summary of the model
> summary(model)

Call:
lm(formula = Strength ~ ., data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-29.032  -6.538   0.668   6.666  34.070

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -53.42099   31.59564  -1.691  0.09132 .
Cement        0.12753    0.01036  12.311  < 2e-16 ***
Slag          0.11283    0.01224   9.218  < 2e-16 ***
FlyAsh        0.09778    0.01525   6.413 2.61e-10 ***
Water        -0.10532    0.04726  -2.228  0.02616 *
Plasticizer   0.35925    0.11024   3.259  0.00117 **
CoarseAgg     0.02965    0.01114   2.662  0.00794 **
FineAgg       0.02928    0.01281   2.286  0.02254 *
Age           0.11824    0.00672  17.595  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.41 on 712 degrees of freedom
Multiple R-squared:  0.618,	Adjusted R-squared:  0.6137
F-statistic:   144 on 8 and 712 DF,  p-value: < 2.2e-16
```

# Assumptions

- Linearity
- Autocorrelation
- Homoscedasticity
- Normality of errors
- Multicollinearity

# Linearity Test :



# Durbin - Watson Test (Auto correlation) :

```
> # 2. Independence of errors (Auto correlation)
> # Performing Durbin-Watson test for Auto correlation
> # Durbin-Watson statistic close to 2 implies no auto correlation
> dwtest(model)

        Durbin-Watson test

data:  model
DW = 2.0045, p-value = 0.5229
alternative hypothesis: true autocorrelation is greater than 0
```

# Breusch - Pagan Test (Homoscedasticity) :

```
> # 3. Homoscedasticity
> # Perform the Breusch-Pagan test for heteroscedasticity
> # The p-value far less than 0.05 indicates evidence of heteroscedasticity
> bptest(model)

	studentized Breusch-Pagan test

data:  model
BP = 94.821, df = 8, p-value < 2.2e-16
```

# Applying WLS

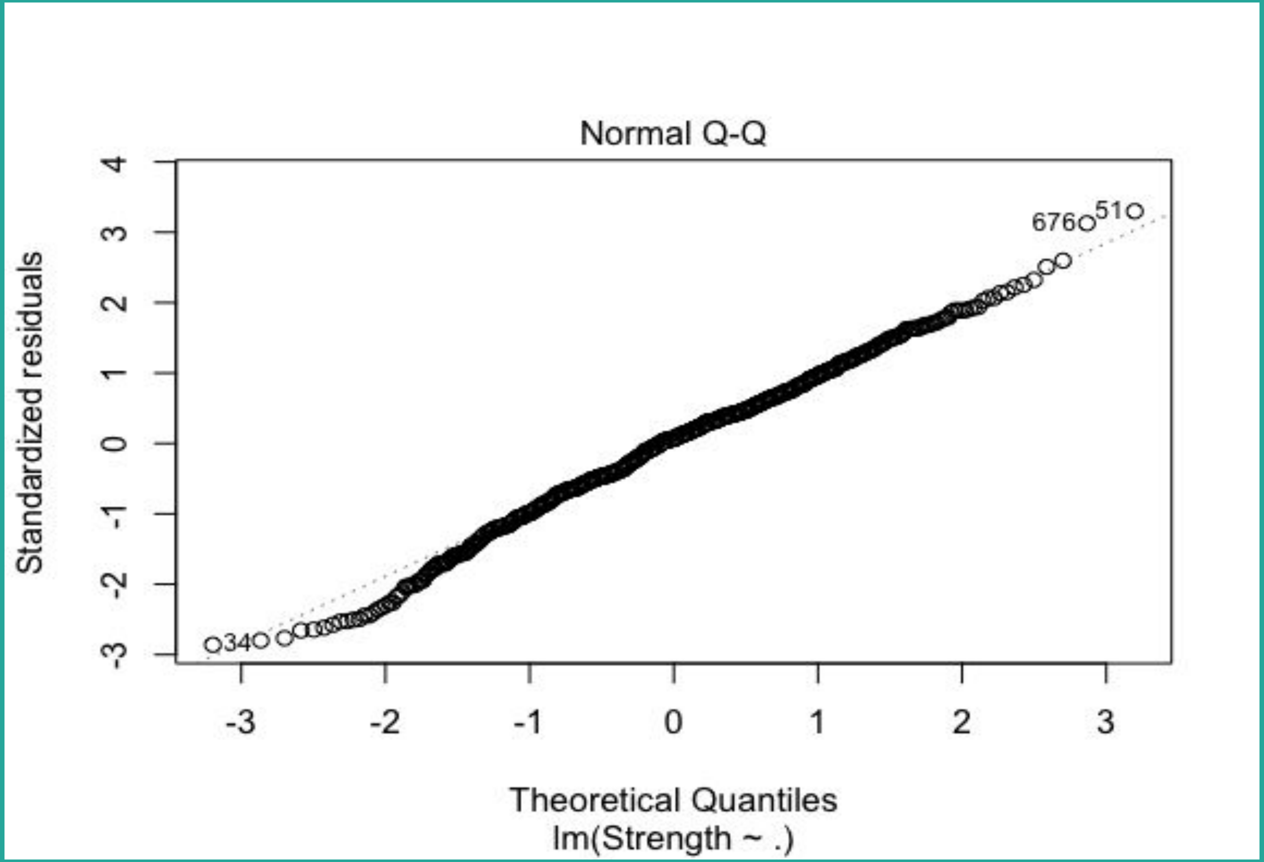
- WLS can be more efficient and accurate than OLS when the data is heteroscedastic, but it requires knowing or estimating the weights for each observation.
- WLS assumes that the data is **heteroscedastic**, meaning that the variability changes as a function of the input variables.
- WLS assigns different weights to each observation based on how reliable or variable they are, while OLS treats all observations equally.

```
> # Check for heteroscedasticity for the new model
> bptest(wls_model)

	studentized Breusch-Pagan test

data:  wls_model
BP = 3.3915, df = 8, p-value = 0.9074

> # Check for autocorrelation in the residuals for the new model
> durbinWatsonTest(wls_model)
 lag Autocorrelation D-W Statistic p-value
   1     0.003232539      1.992087   0.974
 Alternative hypothesis: rho != 0
```

# Updated Summary

```
> # Summary of the new model
> summary(wls_model)

Call:
lm(formula = Strength ~ Age + FineAgg + CoarseAgg + Plasticizer +
    Water + FlyAsh + Slag + Cement, data = train_data, weights = weights)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-5.8325 -2.6193  0.7712  2.4697  5.8090

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -65.525863  15.234305  -4.301 1.94e-05 ***
Age           0.125613   0.004187  30.002  < 2e-16 ***
FineAgg       0.035030   0.005947   5.890 5.94e-09 ***
CoarseAgg     0.032503   0.005496   5.914 5.18e-09 ***
Plasticizer   0.334771   0.057509   5.821 8.84e-09 ***
Water        -0.092191   0.022674  -4.066 5.32e-05 ***
FlyAsh        0.105245   0.006815  15.444  < 2e-16 ***
Slag          0.119386   0.006198  19.262  < 2e-16 ***
Cement        0.133536   0.005184  25.757  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.863 on 712 degrees of freedom
Multiple R-squared:  0.8868,    Adjusted R-squared:  0.8855
F-statistic: 697.3 on 8 and 712 DF,  p-value: < 2.2e-16
```

# Q - Q Plot (Normality of errors) :



# VIF values (Multicollinearity) :

High VIF values indicate high multicollinearity.

Correction Methods:

1. Remove the variables with the highest VIF value (Slag)
2. Ridge Regression

```
> # 5. Multicollinearity
> # VIF values more than 5 or 10 indicate problem with multicollinearity
> vif(wls_model)
        Age     FineAgg   CoarseAgg Plasticizer       Water      FlyAsh
   1.052610    7.746232    7.284560    4.158788    6.307489    6.933859
       Slag      Cement
  11.919606    8.989446
> # Removing the variable with highest VIF Value
> new_train_data = train_data[, -2]
> new_test_data = test_data[, -2]
> # Training the new model
> final_model <- lm(Strength ~., data=new_train_data, weights=weights)
> # Check for multicollinearity after removing the highest VIF variable
> vif(final_model)
     Cement      FlyAsh       Water Plasticizer   CoarseAgg     FineAgg
   1.408337    1.418931    3.551594    4.140427    2.570922    2.001242
        Age
   1.039295
```

# Final train Summary:

```
> # Summary of the final model
> summary(final_model)

Call:
lm(formula = Strength ~ ., data = new_train_data, weights = weights)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-8.5798 -2.8854  0.3369  2.9202 11.4646

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 185.851696   9.686803  19.186  < 2e-16 ***
Cement        0.041828   0.002529  16.539  < 2e-16 ***
FlyAsh       -0.011821   0.003799  -3.111  0.00194 **
Water        -0.380891   0.020970 -18.164  < 2e-16 ***
Plasticizer   0.408376   0.070722   5.774 1.15e-08 ***
CoarseAgg    -0.052656   0.004024 -13.085  < 2e-16 ***
FineAgg      -0.063623   0.003726 -17.078  < 2e-16 ***
Age           0.134683   0.005127  26.268  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.528 on 713 degrees of freedom
Multiple R-squared:  0.8278,    Adjusted R-squared:  0.8261
F-statistic: 489.8 on 7 and 713 DF,  p-value: < 2.2e-16
```

# Predictions :

```
> # Printing the calculating metrics
> print(paste0("Mean Squared Error: ", mse))
[1] "Mean Squared Error: 121.142746846228"
> print(paste0("Mean Absolute Error: ", mae))
[1] "Mean Absolute Error: 8.5141676212751"
> print(paste0("Root Mean Square Error: ", rmse))
[1] "Root Mean Square Error: 11.0064865804773"
> print(paste0("R-Squared: ", rsq))
[1] "R-Squared: 0.827830023152468"
> print(paste0("Adjusted R-Squared: ", adj_rsq))
[1] "Adjusted R-Squared: 0.823826070202525"
```

# Thank You