Data 603 Project

Ahuchogu, Divine | Ouano, Archangelo - 30235358

2023-11-28

Chapter 1: Introduction

This project will be fitting a Multiple linear regression model that would predict the Concrete compressive strength of a certain concrete mixture. To achieve this we used a concrete dataset that contains records where each record represents a unique concrete mixture and its associated Concrete compressive strength. It contains 8 potential predictor variables that are all quantitative.

To restrict the model and determine the best predictor variables, Stepwise, Backward Elimination, Forward Selection, and All-Possible-Regression procedures were used. After coming up with the best First-order regression model, interaction terms and high-order model are considered.

Once we've determined the best possible multiple linear regression model, a diagnostic analysis was conducted to verify if this model meets the different regression assumptions.

Before we continue on the regression aspect of this project, it's important to clearly understand what Concrete Compressive strength is all about and why is it important. An article written by Lysett (2019) [2] about concrete strength defined Concrete Compressive Strength as the "most common and well-accepted measurement of concrete strength, which measures the ability of concrete to withstand loads that will decrease the size of the concrete." This is significantly important since it will determine the quality and longevity of concrete projects as well as the associated costs in meeting concrete strength requirements.

Motivation

The motivation behind this topic is we were curious about what makes the most important component of an infrastructure strong. This is the reason we decided to work on this topic.

Objectives

The objectives of this project are the following:
* To fit a multiple linear regression model that only contains the statistically significant predictor variables that will predict Concrete Compressive Strength; * To ensure that our final regression model conforms with the different regression assumptions;

Chapter 2: Methodology

Data

The dataset used in this project was taken from UC Irvine Machine Learning Repository. It is an open source dataset with the following license: https://creativecommons.org/licenses/by/4.0/legalcode. This dataset contains 1030 records where each record represents a unique concrete-mixture and the associated Concrete compressive strength. It has a total of 9 variables that are all quantitative, namely:

- Cement is a quantitative variable and is measure by kg/m3 mixture

- Blast Furnace Slag is a quantitative variable and is measured by kg/m3 mixture
- Fly Ash is a quantitative variable and is measured by kg/m3 mixture
- Water is a quantitative variable and is measured by kg/m3 mixture
- Superplasticizer is a quantitative variable and is measured by kg/m3 mixture
- Coarse Aggregate is a quantitative variable and is measured by kg/m3 mixture
- Fine Aggregate is a quantitative variable and is measured by kg/m3 mixture
- Age is a quantitative variable and is measured by Days (i.e., 365 days (about 12 months) of a year)
- Concrete compressive strength is a quantitative variable and is measured by MPa.

Concrete compressive strength is our response variable and the rest are our predictor variables.

Workflow

- Fit all possible predictor variables to the regression model
- Use Stepwise, Backward Elimination, Forward Selection, and All-Possible-Regression Selection Procedures to determine the best predictor variables
- Consider interaction terms and High-order model.
- Perform regression diagnostic analysis
- If the regression model doesn't meet the assumptions perform Box-Cox Transformation

Workload Distribution

- Fitting the Regression model by determining the best predictor variables using the different Selection procedure methods and considering interaction terms and high-order model - Archangelo Ouano
- Performing regression model diagnostic test by verifying if the model conforms with the different regression assumptions - Divine Ahuchogu

Chapter 3: Main Results of the Analysis

Fitting the Full model

```
concretedata =
↪  read.csv("https://raw.githubusercontent.com/Archangelo08/Data-603-Project/main/cleanedconc_data.csv
↪  header=TRUE)
head(concretedata, 6)
```

```
##   Cement Blast_Furnace_Slag Fly_Ash Water Superplasticizer Coarse_Aggregate
## 1  168.0               42.1   163.8 121.8              5.7           1058.7
## 2  168.0               42.1   163.8 121.8              5.7           1058.7
## 3  168.0               42.1   163.8 121.8              5.7           1058.7
## 4  168.0               42.1   163.8 121.8              5.7           1058.7
## 5  168.0               42.1   163.8 121.8              5.7           1058.7
## 6  213.7               98.1    24.5 181.7              6.9           1065.8
##   Fine_Aggregate Age Concrete_compressive_strength
```

```
## 1          780.1   3                          7.75
## 2          780.1  14                         17.82
## 3          780.1  28                         24.24
## 4          780.1  56                         32.85
## 5          780.1 100                         39.23
## 6          785.4   3                         18.00
```

```r
tail(concretedata, 5)
```

```
##     Cement Blast_Furnace_Slag Fly_Ash Water Superplasticizer Coarse_Aggregate
## 221  139.7              163.9   127.7 236.7              5.8            868.6
## 222  264.5              111.0    86.5 195.5              5.9            832.6
## 223  276.4              116.0    90.3 179.6              8.9            870.1
## 224  148.5              139.4   108.6 192.7              6.1            892.4
## 225  260.9              100.5    78.3 200.6              8.6            864.5
##     Fine_Aggregate Age Concrete_compressive_strength
## 221          655.6  28                         35.23
## 222          790.4  28                         41.54
## 223          768.3  28                         44.28
## 224          780.0  28                         23.70
## 225          761.5  28                         32.40
```

Creating the Full model:

```r
fullmodel = lm(Concrete_compressive_strength~Cement +
                 Blast_Furnace_Slag +
                 Fly_Ash+Water +
                 Superplasticizer +
                 Coarse_Aggregate +
                 Fine_Aggregate+Age,
               data=concretedata)
summary(fullmodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Fly_Ash + Water + Superplasticizer + Coarse_Aggregate + Fine_Aggregate +
##     Age, data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4393  -5.4489  -0.6626   5.7432  24.3461
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -52.43156   44.19664  -1.186   0.2368
## Cement               0.11838    0.01571   7.536 1.31e-12 ***
## Blast_Furnace_Slag   0.12635    0.02570   4.916 1.74e-06 ***
## Fly_Ash              0.04048    0.02622   1.544   0.1240
## Water               -0.08597    0.05638  -1.525   0.1288
## Superplasticizer     0.08844    0.19974   0.443   0.6584
## Coarse_Aggregate     0.01629    0.01662   0.981   0.3279
## Fine_Aggregate       0.04299    0.01924   2.234   0.0265 *
## Age                  0.38440    0.02423  15.864  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.354 on 216 degrees of freedom
## Multiple R-squared:  0.6672, Adjusted R-squared:  0.6549
## F-statistic: 54.13 on 8 and 216 DF,  p-value: < 2.2e-16
```

Full model Regression Equation:

$\widehat{\text{Concrete Compressive Strength}} = \beta_0 + \beta_1 Cement + \beta_2 Blast\_Furnace\_Slag + \beta_3 Fly\_Ash + \beta 4 Water + \beta_5 Superplasticizer + \beta_6 Coarse\_Aggregate + \beta_7 Fine\_Aggregate + \beta_8 Age$

Choosing the Best model using the following regression procedures:
* Stepwise Regression Procedure
* Backward Elimination Procedure
* Forward Selection Procedure
* All-Possible-Regression Selection Procedure

Using **Stepwise Regression Procedure**:

```
stepfullmodel=ols_step_both_p(fullmodel, pent=0.05, prem=0.1, details=FALSE)
summary(stepfullmodel$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.187  -5.647  -1.089   6.261  25.833
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        22.69556    5.32123   4.265 2.97e-05 ***
## Age                 0.38222    0.02407  15.878  < 2e-16 ***
## Cement              0.09667    0.00696  13.888  < 2e-16 ***
## Blast_Furnace_Slag  0.09163    0.01394   6.571 3.56e-10 ***
## Water              -0.16091    0.03026  -5.318 2.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.39 on 220 degrees of freedom
## Multiple R-squared:  0.6581, Adjusted R-squared:  0.6519
## F-statistic: 105.9 on 4 and 220 DF,  p-value: < 2.2e-16
```

Using **Backward Elimination Procedure**:

```
backfullmodel = ols_step_backward_p(fullmodel, prem=0.1, details=FALSE)
summary(backfullmodel$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -20.8045  -5.9149  -0.7904   5.6779  24.8380
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.108613  13.477344  -0.008   0.9936
## Cement              0.101367   0.007379  13.736  < 2e-16 ***
## Blast_Furnace_Slag  0.102100   0.014991   6.811 9.23e-11 ***
## Water              -0.139147   0.032338  -4.303 2.54e-05 ***
## Fine_Aggregate      0.022293   0.012117   1.840   0.0671 .
## Age                 0.383723   0.023957  16.017  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.345 on 219 degrees of freedom
## Multiple R-squared:  0.6633, Adjusted R-squared:  0.6556
## F-statistic:  86.3 on 5 and 219 DF,  p-value: < 2.2e-16
```

By using the Backward Elimination Procedure, it provides the following first-order regression model: $\beta_0 + \beta_1 Cement + \beta_2 Blast\_Furnace\_Slag + \beta_3 Water + \beta_4 Fine\_Aggregate + \beta_5 Age$. However, the p-value of Fine_Aggregate is > 0.05, so this will be dropped from the model.

Using **Forward Selection Procedure**

```
forwardfullmodel = ols_step_forward_p(fullmodel, penter=0.05, details=FALSE)
summary(forwardfullmodel$model)
```

```
##
## Call:
## lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
##     data = l)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.187  -5.647  -1.089   6.261  25.833
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        22.69556    5.32123   4.265 2.97e-05 ***
## Age                 0.38222    0.02407  15.878  < 2e-16 ***
## Cement              0.09667    0.00696  13.888  < 2e-16 ***
## Blast_Furnace_Slag  0.09163    0.01394   6.571 3.56e-10 ***
## Water              -0.16091    0.03026  -5.318 2.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.39 on 220 degrees of freedom
## Multiple R-squared:  0.6581, Adjusted R-squared:  0.6519
## F-statistic: 105.9 on 4 and 220 DF,  p-value: < 2.2e-16
```

Using **All-Possible-Regression selection procedure**:

```
best_subset = regsubsets(Concrete_compressive_strength~Cement +
                           Blast_Furnace_Slag +
                           Fly_Ash+Water +
                           Superplasticizer +
                           Coarse_Aggregate +
```

```r
                             Fine_Aggregate +
                             Age,
                           data=concretedata,
                           nv=8)
summary(best_subset)
```

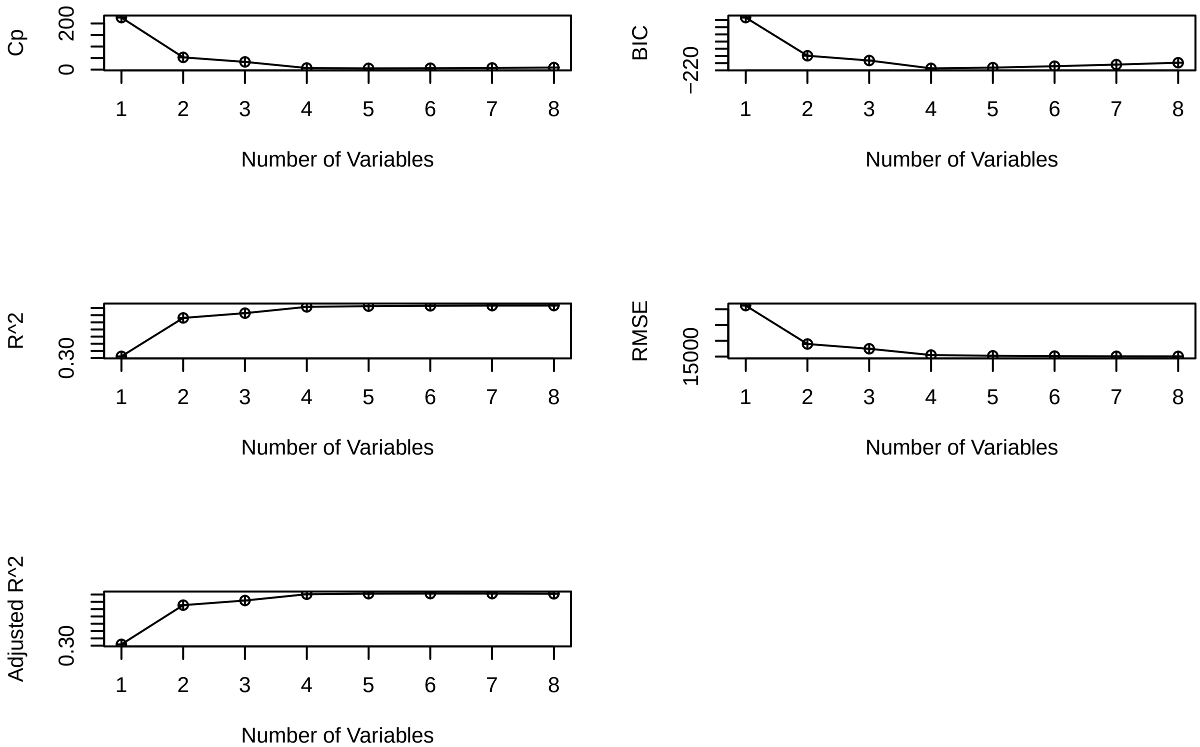
```
## Subset selection object
## Call: regsubsets.formula(Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Fly_Ash + Water + Superplasticizer + Coarse_Aggregate + Fine_Aggregate +
##     Age, data = concretedata, nv = 8)
## 8 Variables  (and intercept)
##                    Forced in Forced out
## Cement                 FALSE      FALSE
## Blast_Furnace_Slag     FALSE      FALSE
## Fly_Ash                FALSE      FALSE
## Water                  FALSE      FALSE
## Superplasticizer       FALSE      FALSE
## Coarse_Aggregate       FALSE      FALSE
## Fine_Aggregate         FALSE      FALSE
## Age                    FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Cement Blast_Furnace_Slag Fly_Ash Water Superplasticizer
## 1  ( 1 ) " "    " "                " "     " "   " "
## 2  ( 1 ) "*"    " "                " "     " "   " "
## 3  ( 1 ) "*"    "*"                " "     " "   " "
## 4  ( 1 ) "*"    "*"                " "     "*"   " "
## 5  ( 1 ) "*"    "*"                " "     "*"   " "
## 6  ( 1 ) "*"    "*"                "*"     "*"   " "
## 7  ( 1 ) "*"    "*"                "*"     "*"   " "
## 8  ( 1 ) "*"    "*"                "*"     "*"   "*"
##          Coarse_Aggregate Fine_Aggregate Age
## 1  ( 1 ) " "              " "            "*"
## 2  ( 1 ) " "              " "            "*"
## 3  ( 1 ) " "              " "            "*"
## 4  ( 1 ) " "              " "            "*"
## 5  ( 1 ) " "              "*"            "*"
## 6  ( 1 ) " "              "*"            "*"
## 7  ( 1 ) "*"              "*"            "*"
## 8  ( 1 ) "*"              "*"            "*"
```

```r
reg_summary = summary(best_subset)
rsquare = c(reg_summary$rsq)
cp = c(reg_summary$cp)
AdjustedR = c(reg_summary$adjr2)
RMSE = c(reg_summary$rss)
BIC = c(reg_summary$bic)
cbind(rsquare,cp,BIC,RMSE,AdjustedR)
```

```
##        rsquare         cp        BIC     RMSE AdjustedR
## [1,] 0.3120532 225.505112  -73.32764 31161.05 0.3089682
## [2,] 0.5808616  53.037662 -179.40136 18985.18 0.5770856
## [3,] 0.6141788  33.413459 -192.62137 17476.05 0.6089414
## [4,] 0.6581219   6.892605 -214.41224 15485.62 0.6519059
```

```
## [5,] 0.6633258   5.515081 -212.44730 15249.90 0.6556392
## [6,] 0.6655487   6.072290 -208.52173 15149.21 0.6563437
## [7,] 0.6668988   7.196054 -204.01571 15088.06 0.6561536
## [8,] 0.6672009   9.000000 -198.80374 15074.38 0.6548750
```

```r
par(mfrow=c(3,2))
plot(reg_summary$cp,type="o",pch=10,xlab="Number of Variables",ylab="Cp")
plot(reg_summary$bic,type="o",pch=10,xlab="Number of Variables",ylab="BIC")
plot(reg_summary$rsq,type="o",pch=10,xlab="Number of Variables",ylab="R^2")
plot(reg_summary$rss,type="o",pch=10,xlab="Number of Variables",ylab="RMSE")
plot(reg_summary$adjr2,type="o",pch=10,xlab="Number of Variables",ylab="Adjusted R^2")
```



Based on the output above, we will be selecting 4 subset of predictor variables namely:
* Cement
* Blast_Furnace_Slag
* Water
* Age

Summary of best predictor variables selected by the different Regression selection procedures above:

Stepwise:
* Age
* Cement
* Blast_Furnace_Slag
* Water

Backward Elimination:
* Cement
* Blast_Furnace_Slag
* Water
* Age

Forward Selection:
* Age
* Cement
* Blast_Furnace_Slage
* Water

All-Possible-Regression:
* Cement
* Blast_Furnace_Slag
* Water
* Age

With these results, our **best first-order regression model** is the following:

```
bestfirstorder = lm(Concrete_compressive_strength~Cement +
                      Blast_Furnace_Slag +
                      Water +
                      Age,
                    data=concretedata)
summary(bestfirstorder)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + Age, data = concretedata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.187  -5.647  -1.089   6.261  25.833
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        22.69556    5.32123   4.265 2.97e-05 ***
## Cement              0.09667    0.00696  13.888  < 2e-16 ***
## Blast_Furnace_Slag  0.09163    0.01394   6.571 3.56e-10 ***
## Water              -0.16091    0.03026  -5.318 2.58e-07 ***
## Age                 0.38222    0.02407  15.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.39 on 220 degrees of freedom
## Multiple R-squared:  0.6581, Adjusted R-squared:  0.6519
## F-statistic: 105.9 on 4 and 220 DF,  p-value: < 2.2e-16
```

First-Order Regression Equation:

$\text{Concrete }\widehat{\text{Compressive}}\text{ Strength} = \beta_0 + \beta_1 Cement + \beta_2 Blat\_Furnace\_Slag + \beta_3 Water + \beta_4 Age$

Checking for Interactions and High-order Model

Intearction terms

```
bestFOinterac = lm(Concrete_compressive_strength~(Cement+Blast_Furnace_Slag+Water+Age)^2,
↪  data=concretedata)
summary(bestFOinterac)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ (Cement + Blast_Furnace_Slag +
##     Water + Age)^2, data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4600  -5.2463  -0.6596   5.1732  22.8420
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.272e+00  2.128e+01   0.436  0.66350
## Cement                     1.321e-01  5.401e-02   2.446  0.01525 *
## Blast_Furnace_Slag         1.267e-01  1.192e-01   1.062  0.28934
## Water                     -2.757e-02  1.234e-01  -0.223  0.82345
## Age                        5.716e-02  2.325e-01   0.246  0.80602
## Cement:Blast_Furnace_Slag  3.187e-04  1.347e-04   2.366  0.01888 *
## Cement:Water              -4.222e-04  3.033e-04  -1.392  0.16537
## Cement:Age                 8.529e-04  2.976e-04   2.865  0.00458 **
## Blast_Furnace_Slag:Water  -5.958e-04  5.967e-04  -0.998  0.31924
## Blast_Furnace_Slag:Age     6.383e-05  7.517e-04   0.085  0.93241
## Water:Age                  7.139e-04  1.553e-03   0.460  0.64617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.095 on 214 degrees of freedom
## Multiple R-squared:  0.6904, Adjusted R-squared:  0.676
## F-statistic: 47.73 on 10 and 214 DF,  p-value: < 2.2e-16
```

Using partial T test to drop interaction terms that are not significant in predicting the response variable, with $\alpha = 0.05$, the only interaction terms that are significant are the following:
* Cement:Blast_Furnace_Slag
* Cement:Age

```r
bestFOredinterac =
↪   lm(Concrete_compressive_strength~Cement+Blast_Furnace_Slag+Water+Age+Cement:Blast_Furnace_Slag+Cemen
↪   data=concretedata)
summary(bestFOredinterac)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + Age + Cement:Blast_Furnace_Slag + Cement:Age, data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.7612  -5.2174  -0.6791   5.5197  23.1825
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               31.7716958  5.5157620   5.760 2.84e-08 ***
## Cement                     0.0589147  0.0108441   5.433 1.48e-07 ***
## Blast_Furnace_Slag         0.0276904  0.0288332   0.960 0.337935
## Water                     -0.1580912  0.0291780  -5.418 1.59e-07 ***
## Age                        0.1710056  0.0614408   2.783 0.005854 **
```

```
## Cement:Blast_Furnace_Slag  0.0003061  0.0001280   2.391 0.017659 *
## Cement:Age                 0.0008821  0.0002424   3.639 0.000342 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.071 on 218 degrees of freedom
## Multiple R-squared:  0.6865, Adjusted R-squared:  0.6778
## F-statistic: 79.55 on 6 and 218 DF,  p-value: < 2.2e-16
```

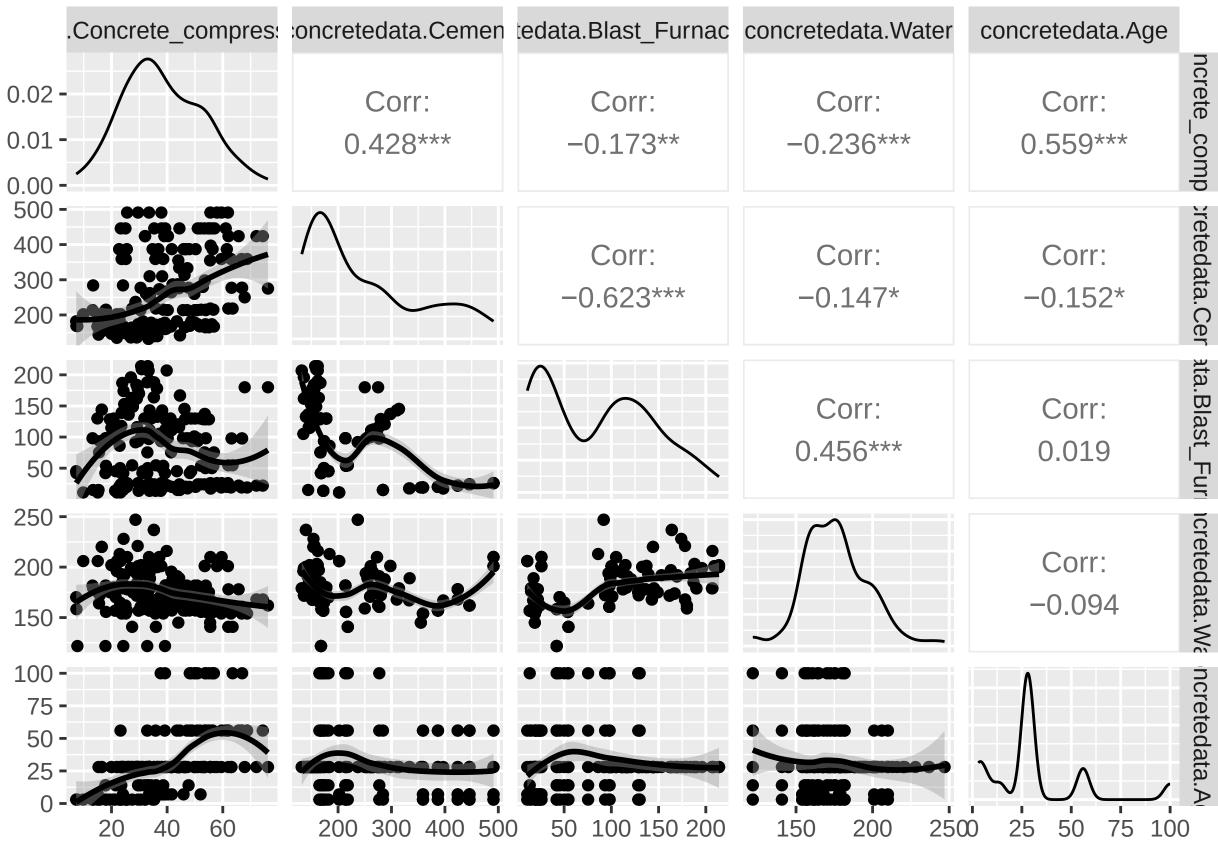
Comparing the best first-order model with the same regression model, but with interaction terms (significant only):
Best First-order regression model:
* RMSE = 8.39
* Adjusted R-squared = 0.6519

Best First-order regression model, including significant interaction terms:
* RMSE = 8.071
* Adjusted R-squared = 0.6778

We can infer that the best first-order model that includes significant interaction terms is better.

High-order model

```
bestFOconcretedata = data.frame(
  concretedata$Concrete_compressive_strength,
  concretedata$Cement,
  concretedata$Blast_Furnace_Slag,
  concretedata$Water,
  concretedata$Age)

ggpairs(bestFOconcretedata,
        lower=list(
          continuous="smooth_loess",
          combo="facethist",
          discrete="facetbar",
          na="na"),
        progress=FALSE)
```



Based on the ggpairs matrix visual, the predictor variable that potentially supports high-order model are: * Blast_Furnace_Slag * Age (maybe)

Let's create the regression model that includes high-order model:

```
secondordermodel = lm(Concrete_compressive_strength~Cement +
↪  poly(Blast_Furnace_Slag,2,raw=T) + Water + poly(Age,2,raw=T) +
↪  Cement:Blast_Furnace_Slag + Cement:Age, data=concretedata)
summary(secondordermodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + poly(Blast_Furnace_Slag,
##     2, raw = T) + Water + poly(Age, 2, raw = T) + Cement:Blast_Furnace_Slag +
##     Cement:Age, data = concretedata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.349  -5.352  -0.457   5.161  22.420
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           2.418e+01  5.566e+00   4.345 2.14e-05
## Cement                                8.061e-02  1.156e-02   6.975 3.68e-11
## poly(Blast_Furnace_Slag, 2, raw = T)1  3.522e-02  6.280e-02   0.561   0.5755
## poly(Blast_Furnace_Slag, 2, raw = T)2 -1.041e-04  2.055e-04  -0.506   0.6130
## Water                                -1.815e-01  2.713e-02  -6.692 1.87e-10
## poly(Age, 2, raw = T)1                8.679e-01  1.204e-01   7.209 9.32e-12
```

```
## poly(Age, 2, raw = T)2                -5.032e-03  7.679e-04  -6.553 4.06e-10
## Cement:Blast_Furnace_Slag               2.772e-04  1.419e-04   1.953   0.0521
## Cement:Age                             -2.175e-05  2.617e-04  -0.083   0.9339
##
## (Intercept)                            ***
## Cement                                 ***
## poly(Blast_Furnace_Slag, 2, raw = T)1
## poly(Blast_Furnace_Slag, 2, raw = T)2
## Water                                  ***
## poly(Age, 2, raw = T)1                 ***
## poly(Age, 2, raw = T)2                 ***
## Cement:Blast_Furnace_Slag              .
## Cement:Age
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.406 on 216 degrees of freedom
## Multiple R-squared:  0.7385, Adjusted R-squared:  0.7288
## F-statistic: 76.24 on 8 and 216 DF,  p-value: < 2.2e-16
```

Based on the output above, transforming Blast_Furnace_Slag predictor variable into a high-order model turned out to be not significant in predicting the response variable. On the other hand, Age is. Let's check if transforming Age into its third order is still significant:

```r
thirdordermodel = lm(Concrete_compressive_strength~Cement + Blast_Furnace_Slag + Water +
↪  poly(Age,3,raw=T) + Cement:Blast_Furnace_Slag + Cement:Age, data=concretedata)
summary(thirdordermodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + poly(Age, 3, raw = T) + Cement:Blast_Furnace_Slag +
##     Cement:Age, data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6408  -5.1915  -0.5124   4.9235  21.9047
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.469e+01  5.124e+00   4.819 2.71e-06 ***
## Cement                     7.755e-02  1.029e-02   7.537 1.30e-12 ***
## Blast_Furnace_Slag         5.786e-03  2.644e-02   0.219   0.8270
## Water                     -1.847e-01  2.686e-02  -6.876 6.53e-11 ***
## poly(Age, 3, raw = T)1     1.141e+00  1.844e-01   6.190 2.99e-09 ***
## poly(Age, 3, raw = T)2    -1.382e-02  4.555e-03  -3.034   0.0027 **
## poly(Age, 3, raw = T)3     6.137e-05  3.125e-05   1.964   0.0508 .
## Cement:Blast_Furnace_Slag  2.852e-04  1.177e-04   2.424   0.0162 *
## Cement:Age                 7.997e-05  2.644e-04   0.302   0.7626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.345 on 216 degrees of freedom
## Multiple R-squared:  0.7428, Adjusted R-squared:  0.7332
## F-statistic: 77.96 on 8 and 216 DF,  p-value: < 2.2e-16
```

It turns out transforming Age into its third model is not significant with p value > 0.05. So, we'll stop at its second order model.

This is the best regression model, including interaction terms and high-order model:

```
bestmodel = lm(Concrete_compressive_strength~Cement + Blast_Furnace_Slag + Water +
↪  poly(Age,2,raw=T) + Cement:Blast_Furnace_Slag + Cement:Age, data=concretedata)
summary(bestmodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + poly(Age, 2, raw = T) + Cement:Blast_Furnace_Slag +
##     Cement:Age, data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6120  -5.2145  -0.5163   4.9635  21.6908
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.524e+01  5.150e+00   4.902 1.86e-06 ***
## Cement                     7.803e-02  1.035e-02   7.536 1.29e-12 ***
## Blast_Furnace_Slag         6.420e-03  2.661e-02   0.241  0.80956
## Water                     -1.801e-01  2.694e-02  -6.686 1.91e-10 ***
## poly(Age, 2, raw = T)1     8.651e-01  1.201e-01   7.206 9.41e-12 ***
## poly(Age, 2, raw = T)2    -5.001e-03  7.641e-04  -6.545 4.22e-10 ***
## Cement:Blast_Furnace_Slag  3.175e-04  1.173e-04   2.707  0.00733 **
## Cement:Age                -1.884e-05  2.612e-04  -0.072  0.94257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.393 on 217 degrees of freedom
## Multiple R-squared:  0.7382, Adjusted R-squared:  0.7297
## F-statistic:  87.4 on 7 and 217 DF,  p-value: < 2.2e-16
```

After we have applied a high-order transformation on the Age predictor variable the interaction term Cement:Age turned out to be insignificant in predicting the response variable, so we will be dropping this.

```
bestmodel = lm(Concrete_compressive_strength~Cement + Blast_Furnace_Slag + Water +
↪  poly(Age,2,raw=T) + Cement:Blast_Furnace_Slag, data=concretedata)
summary(bestmodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + poly(Age, 2, raw = T) + Cement:Blast_Furnace_Slag,
##     data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6056  -5.1944  -0.5135   4.9702  21.6962
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               25.3808699  4.7678813   5.323 2.53e-07 ***
```

```
## Cement                      0.0775459  0.0079122   9.801  < 2e-16 ***
## Blast_Furnace_Slag          0.0065590  0.0264791   0.248  0.80460    
## Water                      -0.1800125  0.0268421  -6.706 1.69e-10 ***
## poly(Age, 2, raw = T)1      0.8578395  0.0660286  12.992  < 2e-16 ***
## poly(Age, 2, raw = T)2     -0.0049722  0.0006479  -7.674 5.52e-13 ***
## Cement:Blast_Furnace_Slag  0.0003171  0.0001169   2.712  0.00721 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.376 on 218 degrees of freedom
## Multiple R-squared:  0.7382, Adjusted R-squared:  0.731 
## F-statistic: 102.4 on 6 and 218 DF,  p-value: < 2.2e-16
```

Comparing the regression model that includes only interaction terms and the regression model that includes both interaction terms and high-order model:
Best regression model only interaction terms:
* RMSE = 8.071
* Adjusted R-squared = 0.6778

Best regression model with interaction terms and high-order model:
* RMSE = 7.376
* Adjusted R-squared = 0.731

We can infer that the regression model that includes both interaction terms and high-order model (all significant) is better.

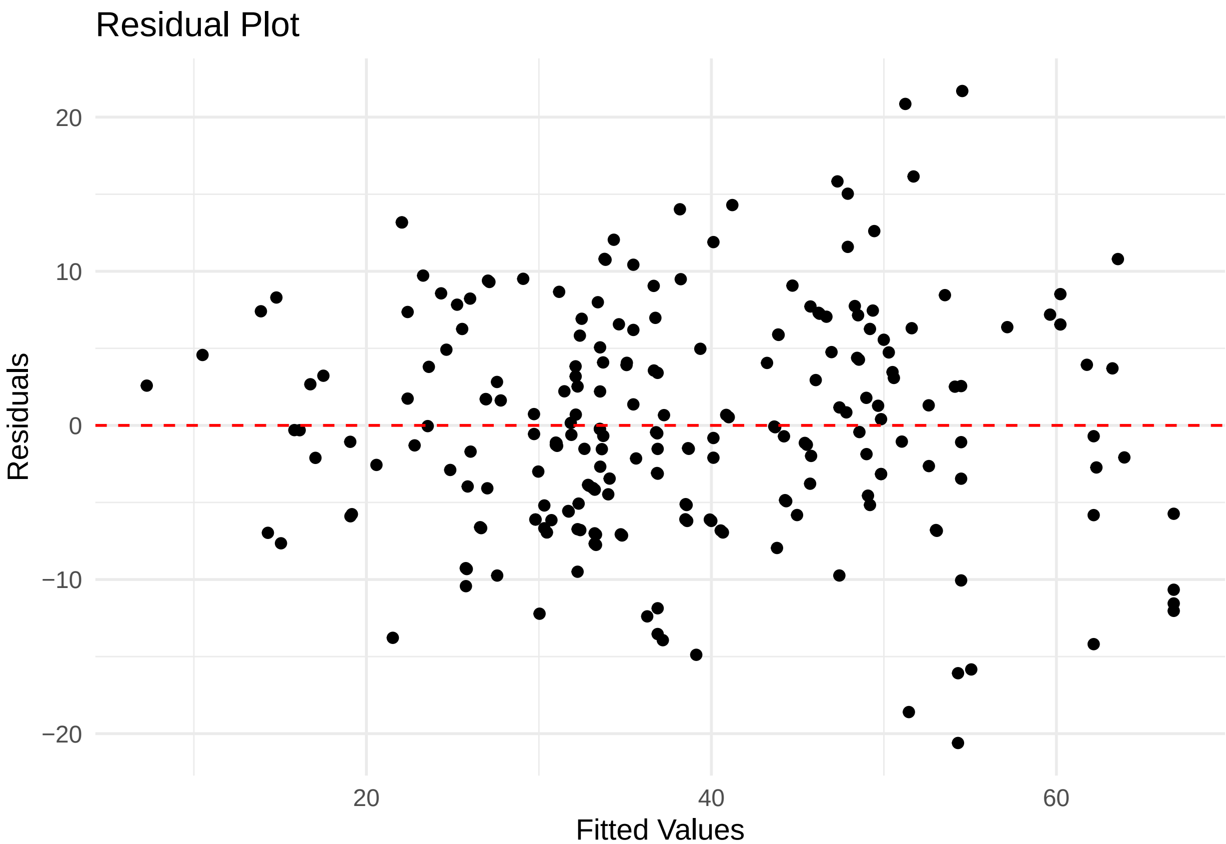
This is the best model after using the different regression selection procedures and including both interaction terms and high-order model

$\widehat{Concrete\_compressive\_strength}$ = 25.3809 + (0.0775459+0.0003171Blast_Furnace_Slag)Cement + (0.0065590+0.0003171)Blast_Furnace_Slag + 0.8578Age - 0.0050Age^2

Checking the Regression Assumptions

Linearity Assumption

```r
residuals <- residuals(bestmodel)
fitted.values <- fitted(bestmodel)

# Create a data frame for ggplot
residual_plot_data <- data.frame(Fitted = fitted.values, Residuals = residuals)

ggplot(residual_plot_data, aes(x = Fitted, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```



Equal Variance Assumption

Defining the Hypothesis test:

$$H_0 : \text{Heteroscedasticity does not exist}$$
$$H_A : \text{Heteroscedasticity does exist}$$

Using the Breusch-Pagan test to check for Heteroscedasticity:

```
bptest(bestmodel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bestmodel
## BP = 32.279, df = 6, p-value = 1.443e-05
```

Based on the result from the Breusch-Pagan test p-value, we can say that there is heteroscedasticity in the model and conclude that the equal variance assumption is not met by the model.

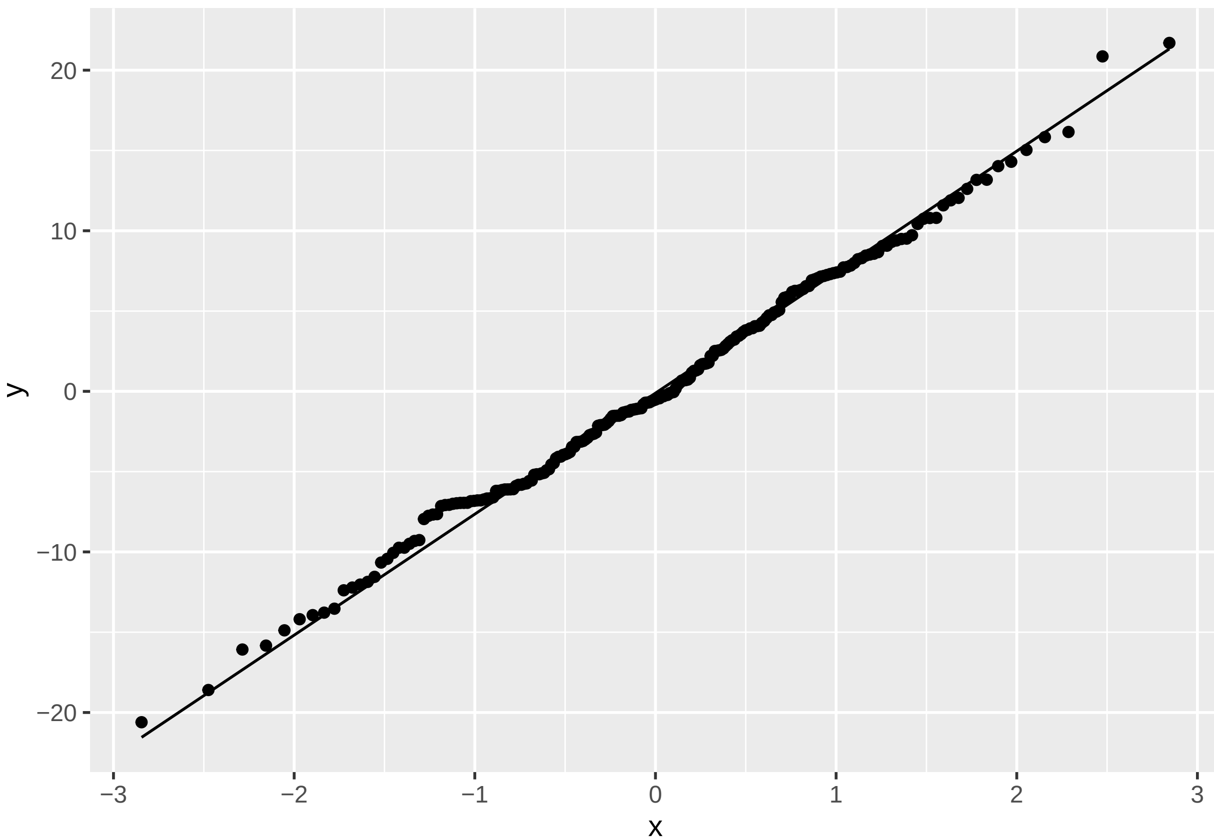
Normality Assumption

$$H_0 : \text{The data are significantly normally distributed}$$
$$H_A : \text{The data are not significantly normally distributed}$$

```
# Shapiro-Wilk normality test
shapiro.test(residuals(bestmodel))
```

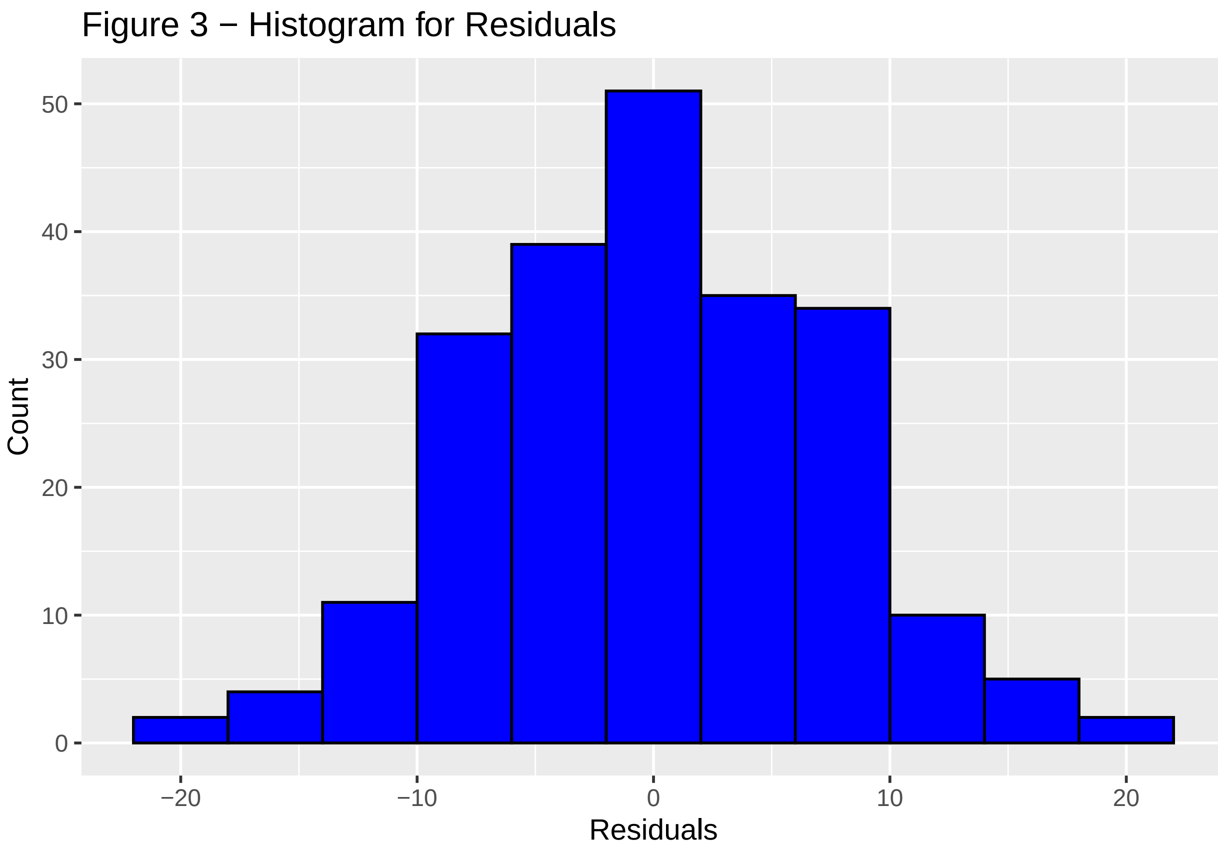
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(bestmodel)
## W = 0.99583, p-value = 0.8049
```

```r
ggplot(data = concretedata, aes(sample=bestmodel$residuals)) + stat_qq() + stat_qq_line()
```



This Shapiro-Wilk normality test with a p-value of 0.8049 and the normality graph, gives us an evidence to say that the normality assumption is met by the model.

```r
ggplot(data = concretedata, aes(x = residuals(bestmodel))) +   geom_histogram(binwidth =
↪  4, fill = "blue", color = "black") +   labs(title = "Figure 3 - Histogram for
↪  Residuals", x = "Residuals", y = "Count")
```



This histogram also further shows the conclusion of our normality assumption.

Mutlicollinearity

```
# Since our best model is a higher order model, we will perform the VIF on the first
↪  order model and the higher order model which is our best model.

firstordermodel = lm(Concrete_compressive_strength~Cement+Blast_Furnace_Slag+Water+Age,
↪  data=concretedata)

vif(firstordermodel)
```

```
##             Cement Blast_Furnace_Slag              Water                Age
##           1.746437           2.106157           1.322332           1.040551
```

```
imcdiag(firstordermodel, method="VIF")
```

```
##
## Call:
## imcdiag(mod = firstordermodel, method = "VIF")
##
##
##  VIF Multicollinearity Diagnostics
##
##                       VIF detection
## Cement             1.7464         0
## Blast_Furnace_Slag 2.1062         0
## Water              1.3223         0
## Age                1.0406         0
```

```
##
## NOTE:  VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## ===================================
```

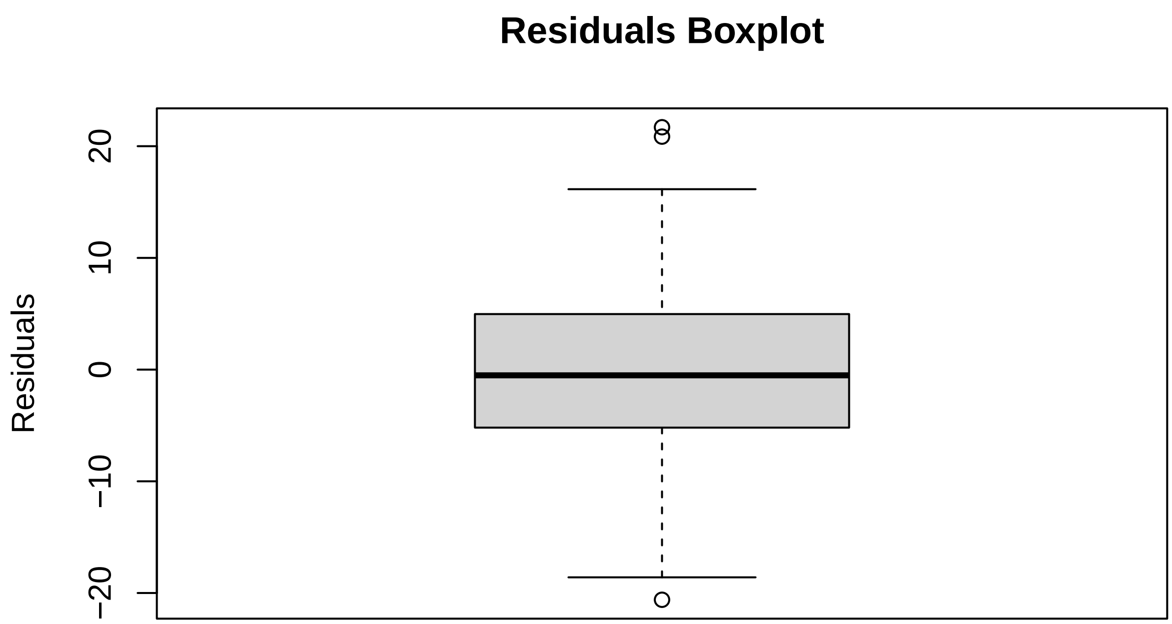
```r
#P erforming it on our best model gives us multicollinearity as seen below

imcdiag(bestmodel, method="VIF")
```

```
##
## Call:
## imcdiag(mod = bestmodel, method = "VIF")
##
##
##  VIF Multicollinearity Diagnostics
##
##                                VIF detection
## Cement                      2.9198         0
## Blast_Furnace_Slag          9.8256         0
## Water                       1.3462         0
## poly(Age, 2, raw = T)1     10.1289         1
## poly(Age, 2, raw = T)2     10.3858         1
## Cement:Blast_Furnace_Slag   5.9113         0
##
## Multicollinearity may be due to poly(Age, 2, raw = T)1 poly(Age, 2, raw = T)2 regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## ===================================
```

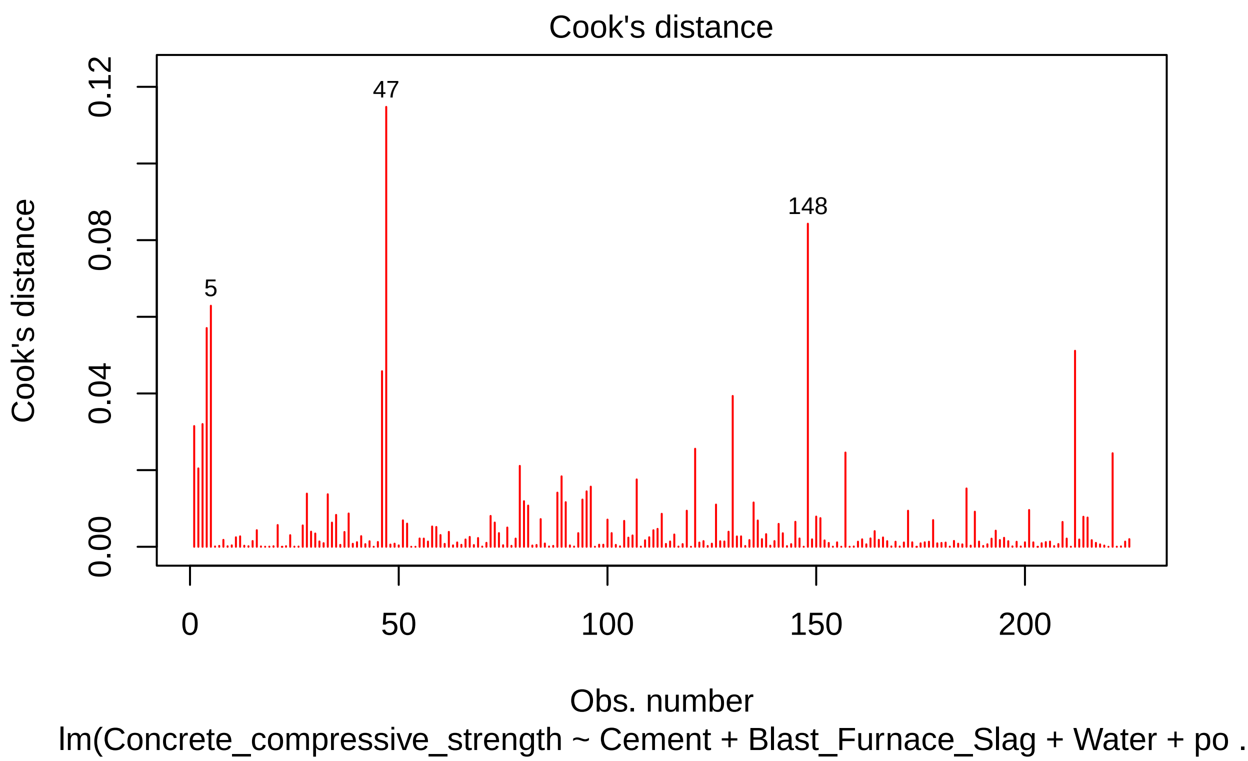
From the two multicollinearity test using the VIF method, we can see that on the first order model, that no multicollinearity was detected but we cannot say the same for our best model given that multicollinearity was detected. This is expected since we transformed the Age predictor variable into its second order model.

Outliers

```r
residuals <- residuals(bestmodel)

# Create a boxplot for residuals
boxplot(residuals, main = "Residuals Boxplot", ylab = "Residuals")
```



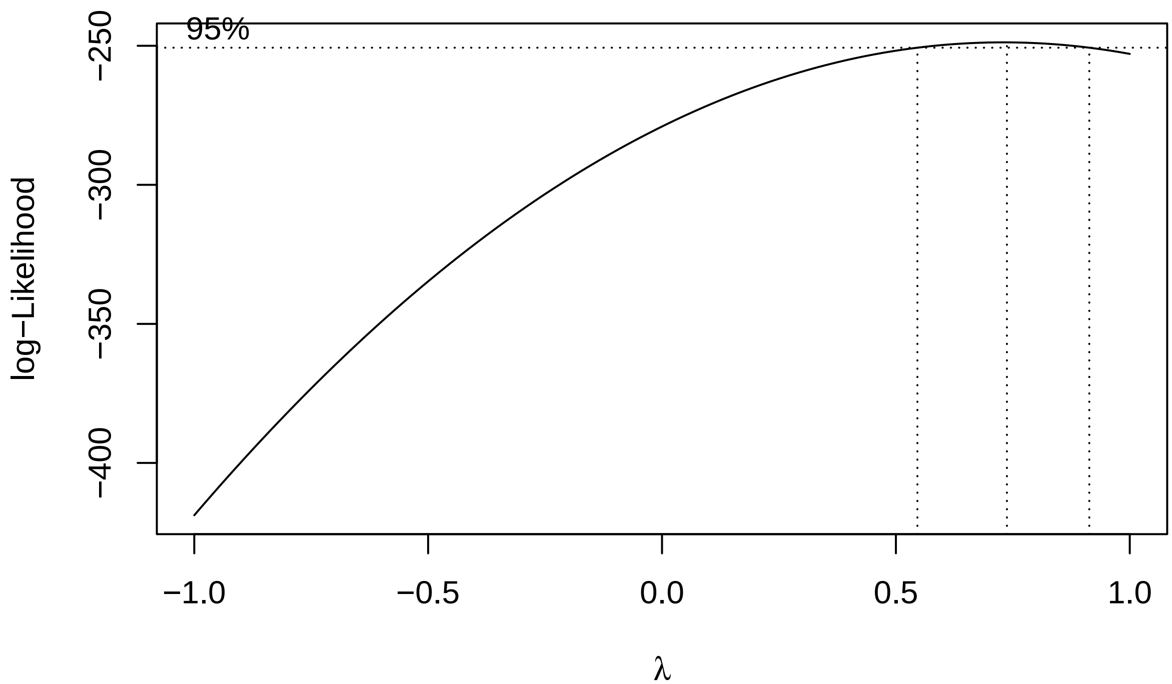

```r
plot(bestmodel, col = "red", which = 4)
```



We can also see that outliers were detected but they are not influential and as such is ignored.

Applying Box-Cox transformation to fix the Heteroscedasticity issue

```r
bc=boxcox(bestmodel,lambda=seq(-1,1))
```



Calculating for λ

```
bestlambda=bc$x[which(bc$y==max(bc$y))]
bestlambda
```

```
## [1] 0.7373737
```

Transforming the response variable:

```
bcmodel = lm((((Concrete_compressive_strength^0.7373737)-1)/0.7373737)~Cement +
               Blast_Furnace_Slag +
               Water +
               poly(Age,2,raw=T) +
               Cement:Blast_Furnace_Slag,
             data=concretedata)
summary(bcmodel)
```

```
##
## Call:
## lm(formula = (((Concrete_compressive_strength^0.7373737) - 1)/0.7373737) ~
##     Cement + Blast_Furnace_Slag + Water + poly(Age, 2, raw = T) +
##         Cement:Blast_Furnace_Slag, data = concretedata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.771 -1.904 -0.063  1.981  7.119
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.282e+01  1.844e+00   6.952 4.12e-11 ***
## Cement                      3.064e-02  3.060e-03  10.013  < 2e-16 ***
## Blast_Furnace_Slag          5.157e-03  1.024e-02   0.503   0.6151
## Water                      -6.905e-02  1.038e-02  -6.651 2.31e-10 ***
## poly(Age, 2, raw = T)1      3.372e-01  2.554e-02  13.204  < 2e-16 ***
```

```
## poly(Age, 2, raw = T)2    -1.974e-03  2.506e-04  -7.877 1.57e-13 ***
## Cement:Blast_Furnace_Slag  1.174e-04  4.522e-05   2.596   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.853 on 218 degrees of freedom
## Multiple R-squared:  0.7395, Adjusted R-squared:  0.7323
## F-statistic: 103.1 on 6 and 218 DF,  p-value: < 2.2e-16
```

Comparing the best model and the model after Box-Cox transformation

Best model:

```r
summary(bestmodel)
```

```
##
## Call:
## lm(formula = Concrete_compressive_strength ~ Cement + Blast_Furnace_Slag +
##     Water + poly(Age, 2, raw = T) + Cement:Blast_Furnace_Slag,
##     data = concretedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6056  -5.1944  -0.5135   4.9702  21.6962
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               25.3808699  4.7678813   5.323 2.53e-07 ***
## Cement                     0.0775459  0.0079122   9.801  < 2e-16 ***
## Blast_Furnace_Slag         0.0065590  0.0264791   0.248  0.80460
## Water                     -0.1800125  0.0268421  -6.706 1.69e-10 ***
## poly(Age, 2, raw = T)1     0.8578395  0.0660286  12.992  < 2e-16 ***
## poly(Age, 2, raw = T)2    -0.0049722  0.0006479  -7.674 5.52e-13 ***
## Cement:Blast_Furnace_Slag  0.0003171  0.0001169   2.712  0.00721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.376 on 218 degrees of freedom
## Multiple R-squared:  0.7382, Adjusted R-squared:  0.731
## F-statistic: 102.4 on 6 and 218 DF,  p-value: < 2.2e-16
```

Model after Box-Cox Transformation:

```r
summary(bcmodel)
```

```
##
## Call:
## lm(formula = (((Concrete_compressive_strength^0.7373737) - 1)/0.7373737) ~
##     Cement + Blast_Furnace_Slag + Water + poly(Age, 2, raw = T) +
##         Cement:Blast_Furnace_Slag, data = concretedata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.771 -1.904 -0.063  1.981  7.119
##
## Coefficients:
```

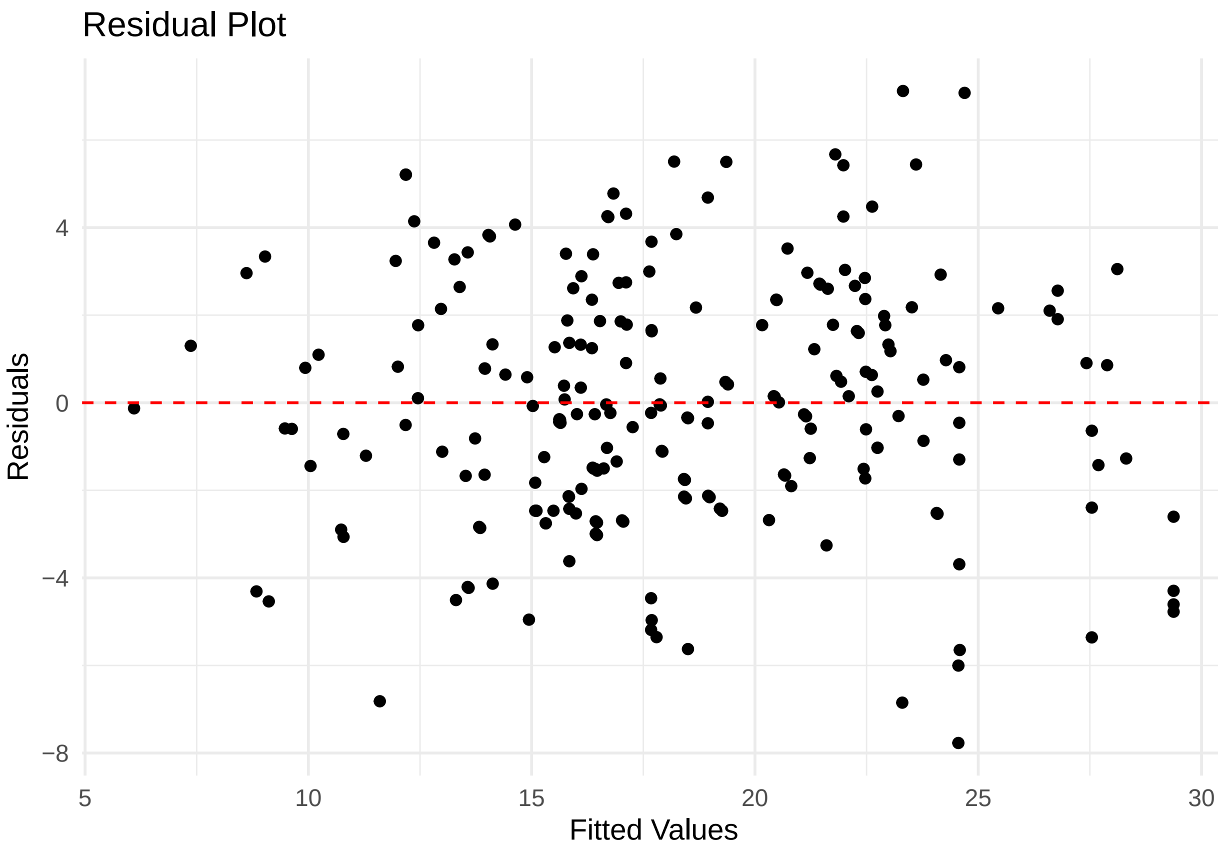
```
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.282e+01  1.844e+00   6.952 4.12e-11 ***
## Cement                       3.064e-02  3.060e-03  10.013  < 2e-16 ***
## Blast_Furnace_Slag           5.157e-03  1.024e-02   0.503   0.6151
## Water                       -6.905e-02  1.038e-02  -6.651 2.31e-10 ***
## poly(Age, 2, raw = T)1       3.372e-01  2.554e-02  13.204  < 2e-16 ***
## poly(Age, 2, raw = T)2      -1.974e-03  2.506e-04  -7.877 1.57e-13 ***
## Cement:Blast_Furnace_Slag  1.174e-04  4.522e-05   2.596   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.853 on 218 degrees of freedom
## Multiple R-squared:  0.7395, Adjusted R-squared:  0.7323
## F-statistic: 103.1 on 6 and 218 DF,  p-value: < 2.2e-16
```

If we look at the RMSE, the model after Box-Cox transformation has been significantly reduced to 2.853. Similarly, the RMSE model also increased from 0.731 to 0.7323. With this information, we can conclude that this is our best model despite it not conforming with the Equal Variance Assumption. Here's the regression equation:

$\widehat{\text{Concrete Compressive Strength}}$ = 12.82 + (0.03064+0.0001174Blast_Furnace_Slag)Cement + (0.005157+0.0001174Cement)Blast_Furnace_Slag - 0.06905Water + 0.3372Age - 0.001974Age^2

Performing Regression Diagnostic with the new model `bcmodel`

```r
residuals <- residuals(bcmodel)
fitted.values <- fitted(bcmodel)

# Create a data frame for ggplot
residual_plot_data <- data.frame(Fitted = fitted.values, Residuals = residuals)

ggplot(residual_plot_data, aes(x = Fitted, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```



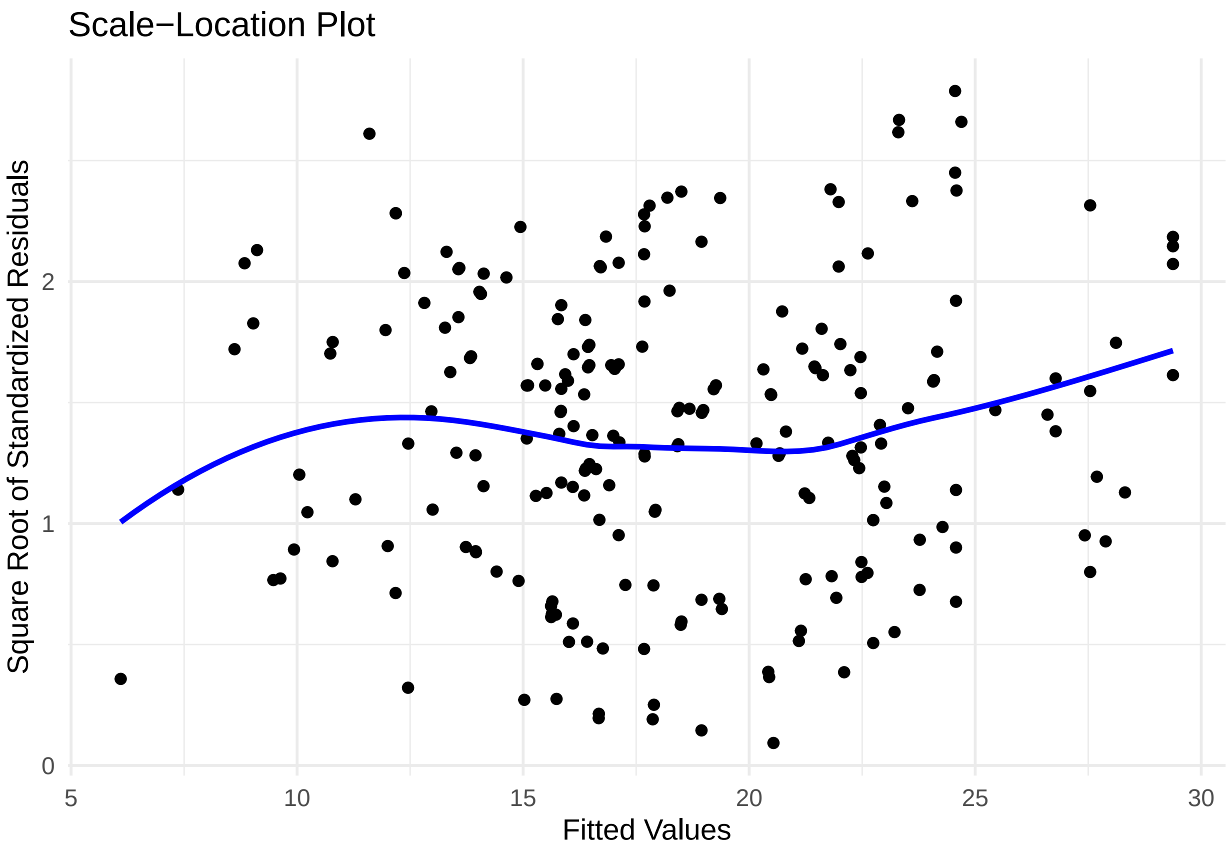
Test for Heteroscedasticity:

```
bptest(bcmodel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bcmodel
## BP = 27.695, df = 6, p-value = 0.0001072
```

```
plot_data <- data.frame(Fitted = fitted.values, Residuals = residuals)

# Create the Scale-Location plot using ggplot2
ggplot(plot_data, aes(x = Fitted, y = sqrt(abs(Residuals)))) + geom_point() +
↪  geom_smooth(method = "loess", se = FALSE, col = "blue") +  labs(title =
↪  "Scale-Location Plot", x = "Fitted Values", y = "Square Root of Standardized
↪  Residuals") + theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

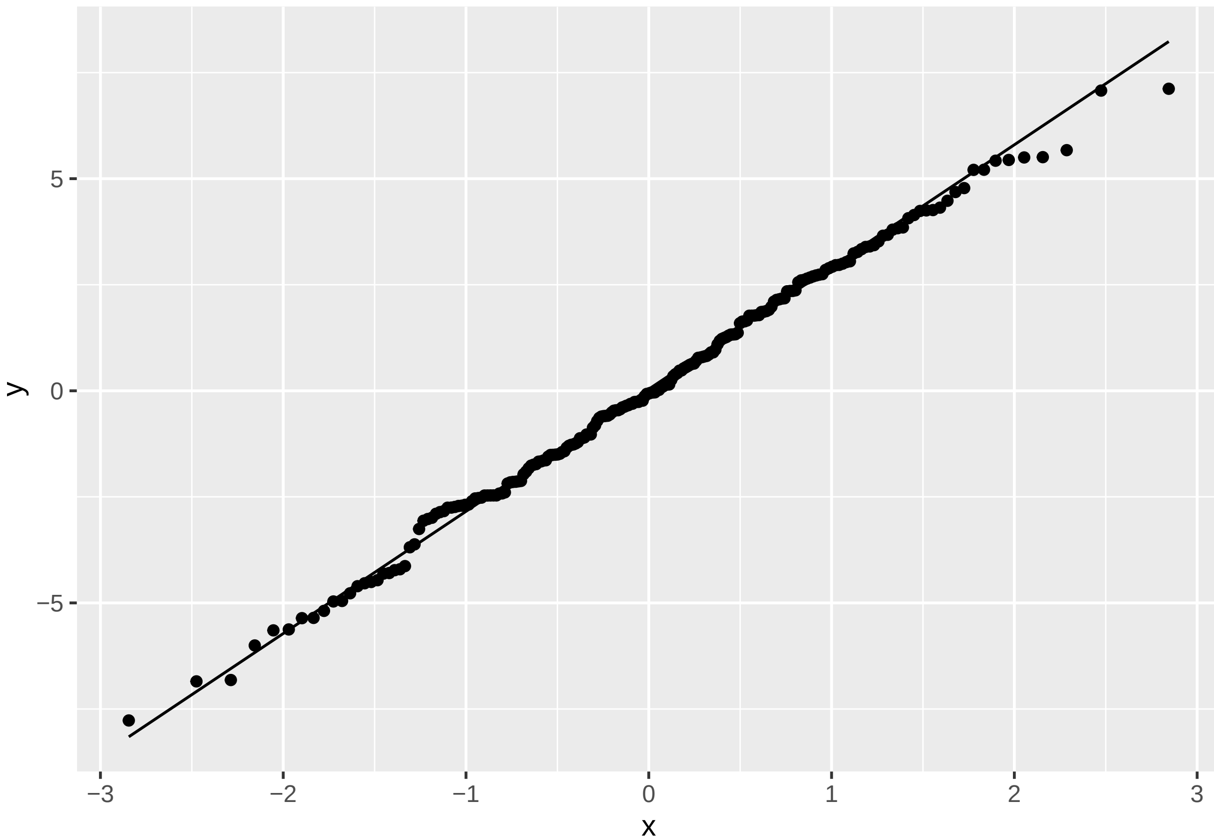


Test for Normality

```r
shapiro.test(residuals(bcmodel))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(bcmodel)
## W = 0.99596, p-value = 0.8247
```

```r
ggplot(data = concretedata, aes(sample=bcmodel$residuals)) + stat_qq() + stat_qq_line()
```



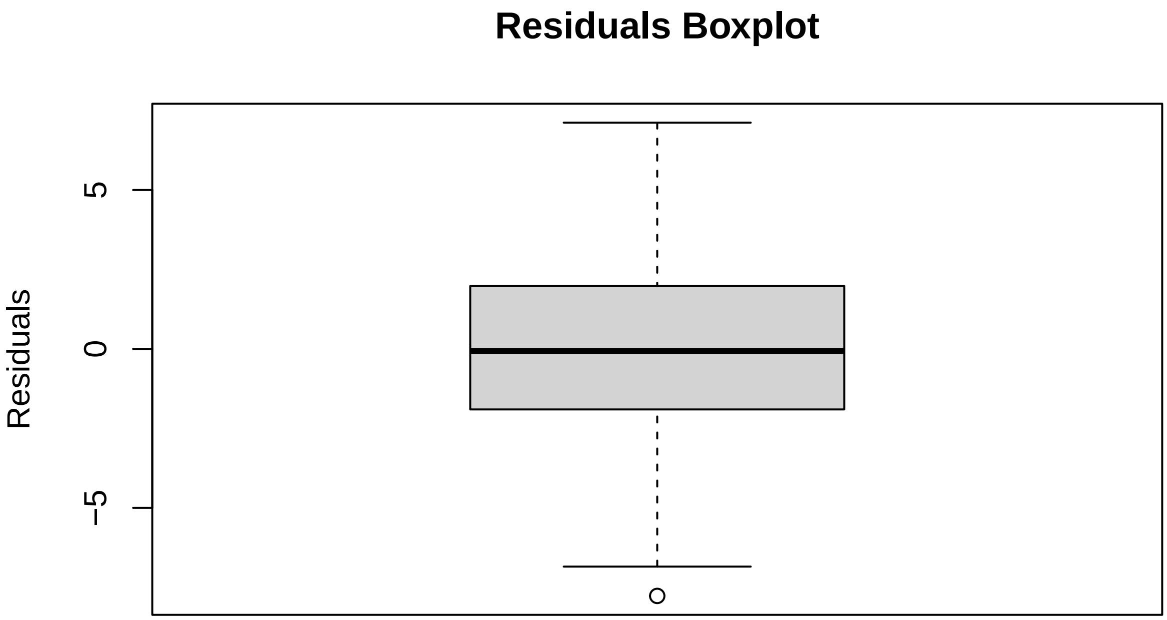
Mutlicollinearity

```r
#Performing it on our bcmodel gives us multicollinearity as well

imcdiag(bcmodel, method="VIF")
```

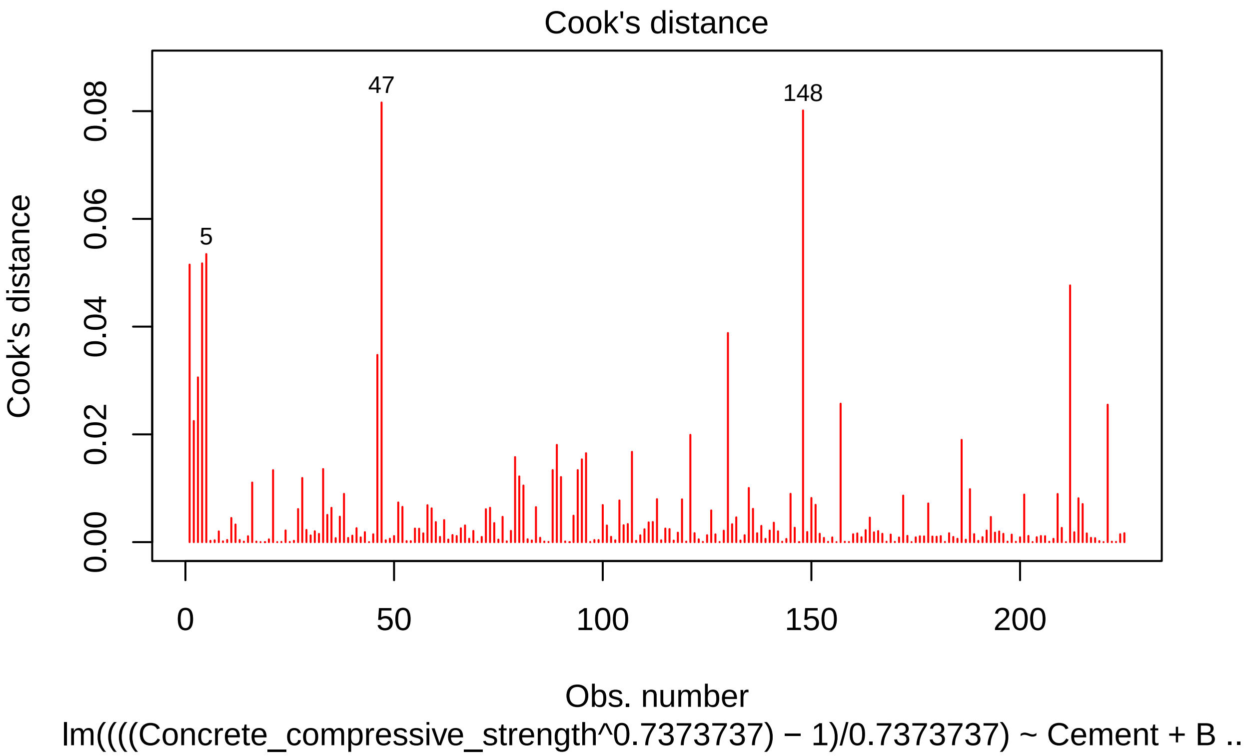
```
##
## Call:
## imcdiag(mod = bcmodel, method = "VIF")
##
##
##  VIF Multicollinearity Diagnostics
##
##                               VIF detection
## Cement                     2.9198         0
## Blast_Furnace_Slag         9.8256         0
## Water                      1.3462         0
## poly(Age, 2, raw = T)1    10.1289         1
## poly(Age, 2, raw = T)2    10.3858         1
## Cement:Blast_Furnace_Slag  5.9113         0
##
## Multicollinearity may be due to poly(Age, 2, raw = T)1 poly(Age, 2, raw = T)2 regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## ===================================
```

Test for Outliers

```r
residuals <- residuals(bcmodel)

boxplot(residuals, main = "Residuals Boxplot", ylab = "Residuals")
```




```r
plot(bcmodel, col = "red", which = 4)
```



Example: Using our best model to predict Concrete Compressive Strength

```r
# Make sure the variable names match those in your original model
new_data <- data.frame(
  Cement = c(168),
  Blast_Furnace_Slag = c(42.1),
  Water = c(121.8),
  Age = c(3)
)

confidence_level <- 0.95

# Use the model to make predictions
predictions <- predict(bcmodel, newdata = new_data, interval = "predict", level =
↪  confidence_level)

print(predictions)
```

```
##        fit      lwr      upr
## 1 11.59976 5.820891 17.37863
```

References

[1] I-Cheng Yeh, "Concrete Compressive Strength." UCI Machine Learning Repository, 1998. doi: 10.24432/C5PK67.
[2] Lysett, T. (2019, March 31). Everything You Need to Know About Concrete Strength | Cor-Tuf. https://cortuf.com/everything-you-need-to-know-about-concrete-strength/