Concrete Strength

Keith Kwanghyun Lee

2022-12-14

Introduction

Decide on the research question

Fit a linear regression model to assess the effect of predictors on the Concrete Strength, and make a prediction about Strength given specified values of the predictors.

Determine the response variable and potential predictors

The response variable: Strength of concrete (Continuous)

Potential predictors are :

1. Blast Furnace Slag(BFS) Slag Produced in Blast Furnace
2. Fly Ash(FA) Amount of ash produced
3. Water Amount of water required
4. Superplasticizer rigidity of cement after drying
5. Coarse Aggregate(CA) The coarse nature of the cement particles
6. Fine Aggregate(FAA) Fineness of the cement
7. Age Age or time before it needs repairing

8.Cement Cement # Analysis

Data Preparation and Cleaning

Import data

```
data <- read_csv("C:/Users/khlee/OneDrive/Documents/GWANGJAAA/NYU/Fall22/Regression/concrete_data.csv")
```

```
## Rows: 1030 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coars...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
names(data)[2] <- 'BFS' # Name Change
names(data)[3] <- 'FA' # Name Change
names(data)[6] <- 'CA' # Name Change
names(data)[7] <- 'FAA' # Name Change

ran <- sample(1:nrow(data),0.8*nrow(data))
data_tr <- data[ran,]
data_tt <- data[-ran,]

data %>%
  head(3)
```

```
## # A tibble: 3 x 9
##   Cement   BFS    FA Water Superplasticizer    CA   FAA   Age Strength
##    <dbl> <dbl> <dbl> <dbl>            <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1   540      0     0   162              2.5  1040   676    28     80.0
## 2   540      0     0   162              2.5  1055   676    28     61.9
## 3   332.   142.    0   228              0     932   594   270     40.3
```

We change some predictors name, such as: Blast Furnace Slag to BFS. Fly Ash to FAA Coarse Aggregate to CA Fine Aggregate to FA

Solve the missing data issue

```r
sum(is.na(data))
```

```
## [1] 0
```

There is no missing value in this dataset. Our data is redy to be analyzed

Exploratory Data Analysis

Summary statistics

This dataset has 1030 observations and 9 Variables(8 Predictors with 1 Response)

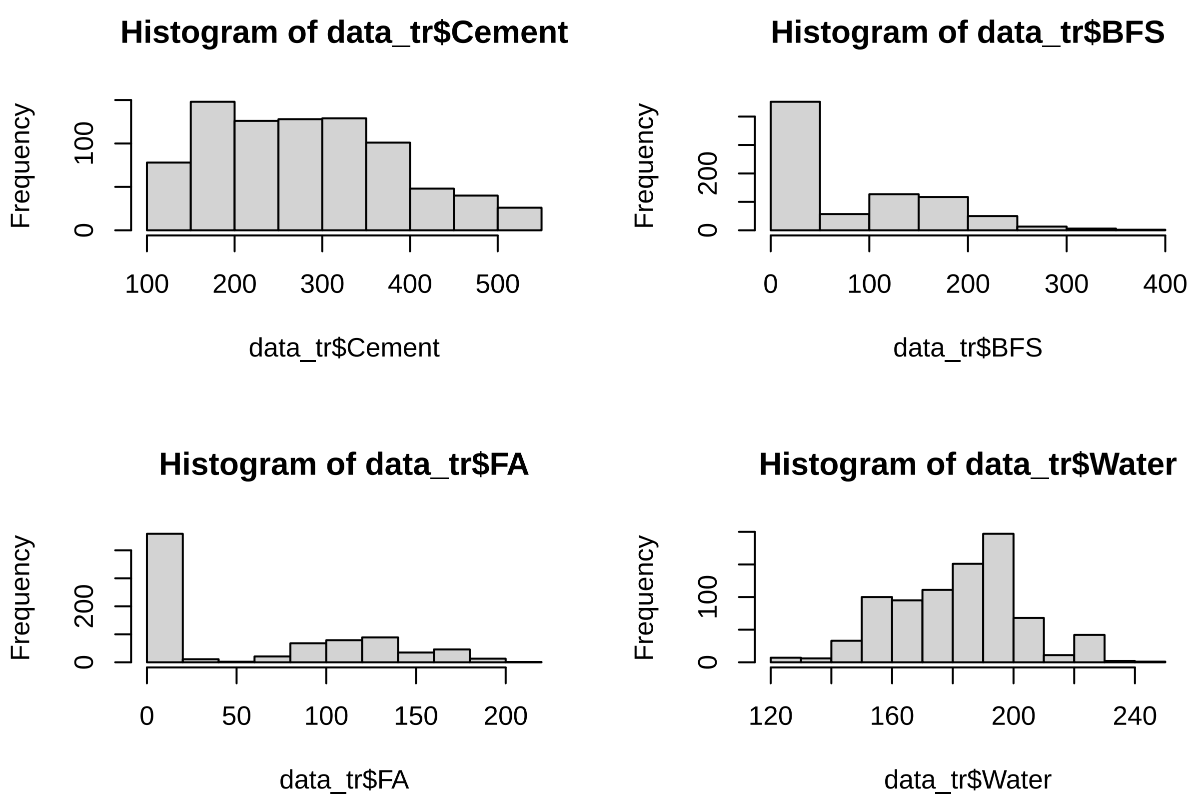
```r
summary(data)
```

```
##      Cement            BFS              FA             Water      
##  Min.   :102.0   Min.   :  0.0   Min.   :  0.00   Min.   :121.8  
##  1st Qu.:192.4   1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.:164.9  
##  Median :272.9   Median : 22.0   Median :  0.00   Median :185.0  
##  Mean   :281.2   Mean   : 73.9   Mean   : 54.19   Mean   :181.6  
##  3rd Qu.:350.0   3rd Qu.:142.9   3rd Qu.:118.30   3rd Qu.:192.0  
##  Max.   :540.0   Max.   :359.4   Max.   :200.10   Max.   :247.0  
##  Superplasticizer       CA              FAA             Age        
##  Min.   : 0.000   Min.   : 801.0   Min.   :594.0   Min.   :  1.00  
##  1st Qu.: 0.000   1st Qu.: 932.0   1st Qu.:731.0   1st Qu.:  7.00  
##  Median : 6.400   Median : 968.0   Median :779.5   Median : 28.00  
```

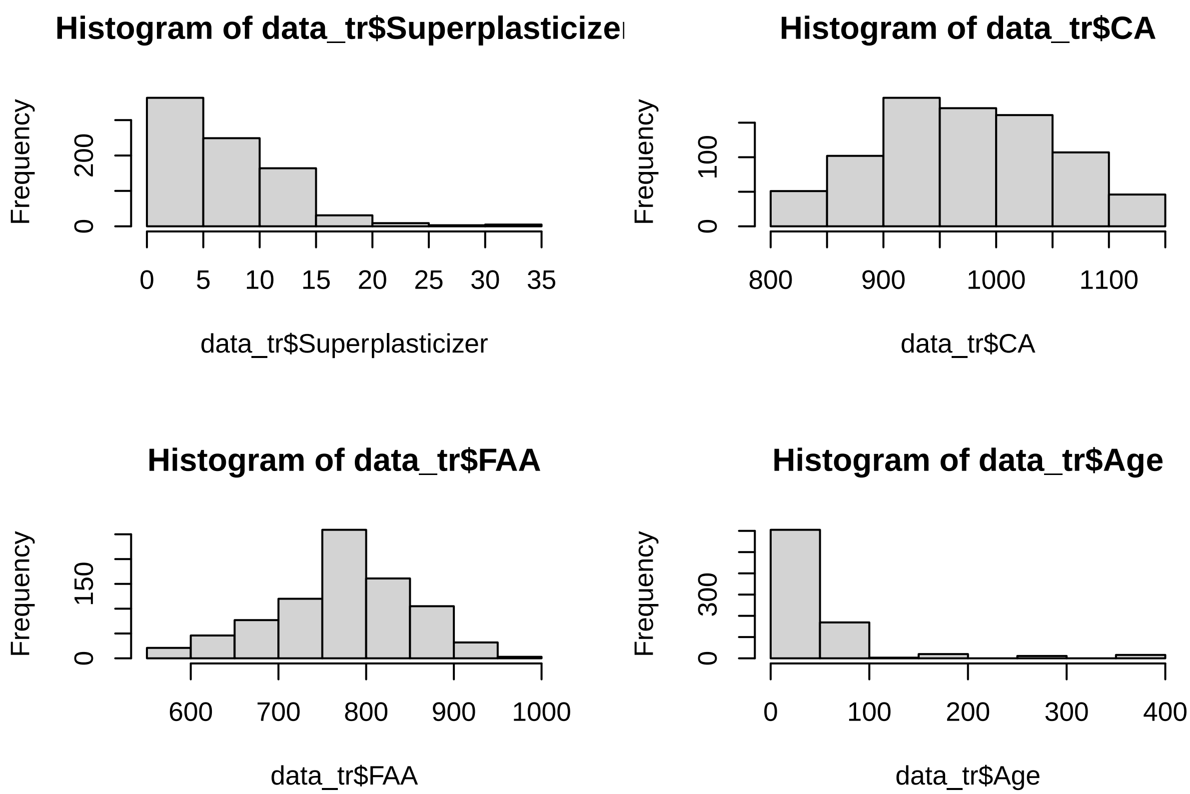
```
##  Mean   : 6.205   Mean   : 972.9   Mean   :773.6   Mean   : 45.66
##  3rd Qu.:10.200   3rd Qu.:1029.4   3rd Qu.:824.0   3rd Qu.: 56.00
##  Max.   :32.200   Max.   :1145.0   Max.   :992.6   Max.   :365.00
##     Strength
##  Min.   : 2.33
##  1st Qu.:23.71
##  Median :34.45
##  Mean   :35.82
##  3rd Qu.:46.13
##  Max.   :82.60
```

Univariate plots

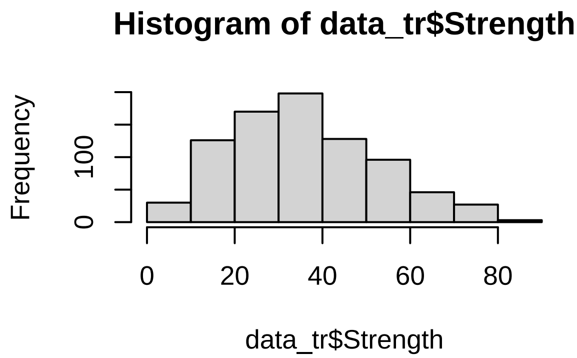
```
par(mfrow = c(2,2))
hist(data_tr$Cement)
hist(data_tr$BFS)
hist(data_tr$FA)
hist(data_tr$Water)
```




```
hist(data_tr$Superplasticizer)
hist(data_tr$CA)
hist(data_tr$FAA)
hist(data_tr$Age)
```



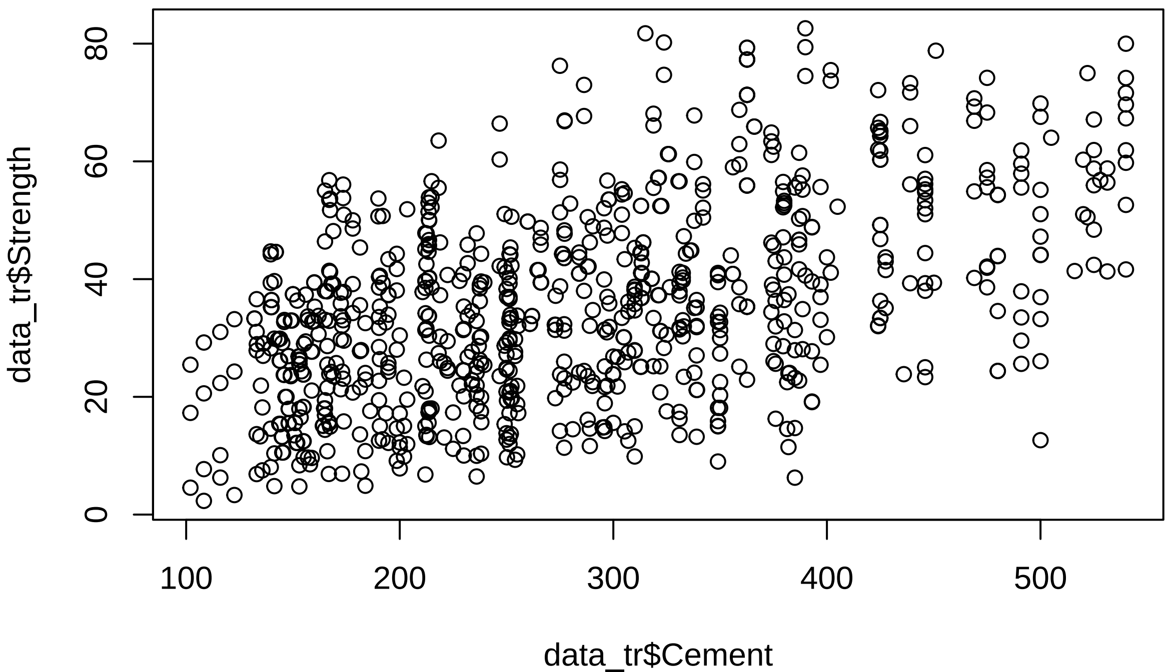

```
hist(data_tr$Strength)
```



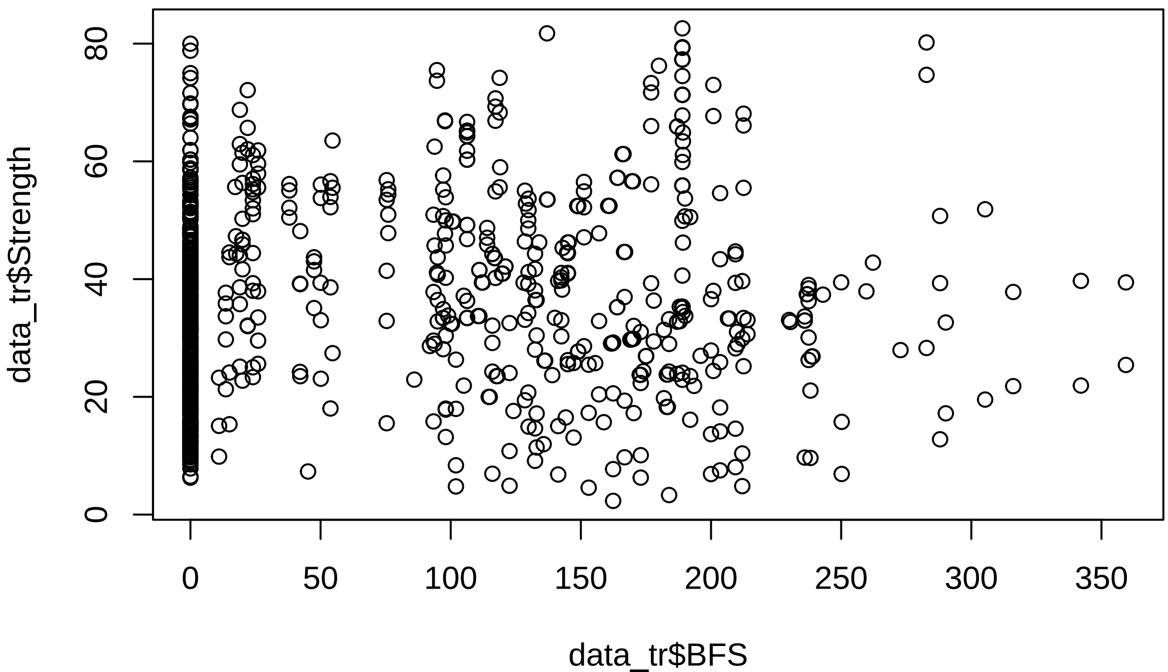
Description: In general, **Water**, **Ca**, **Faa**, and **strength** follow normal distribution; The distribution of **Cement**,**FA**, **BFS**, **Water** and **superplasticizer** is right skewed;

plots for multiple variables

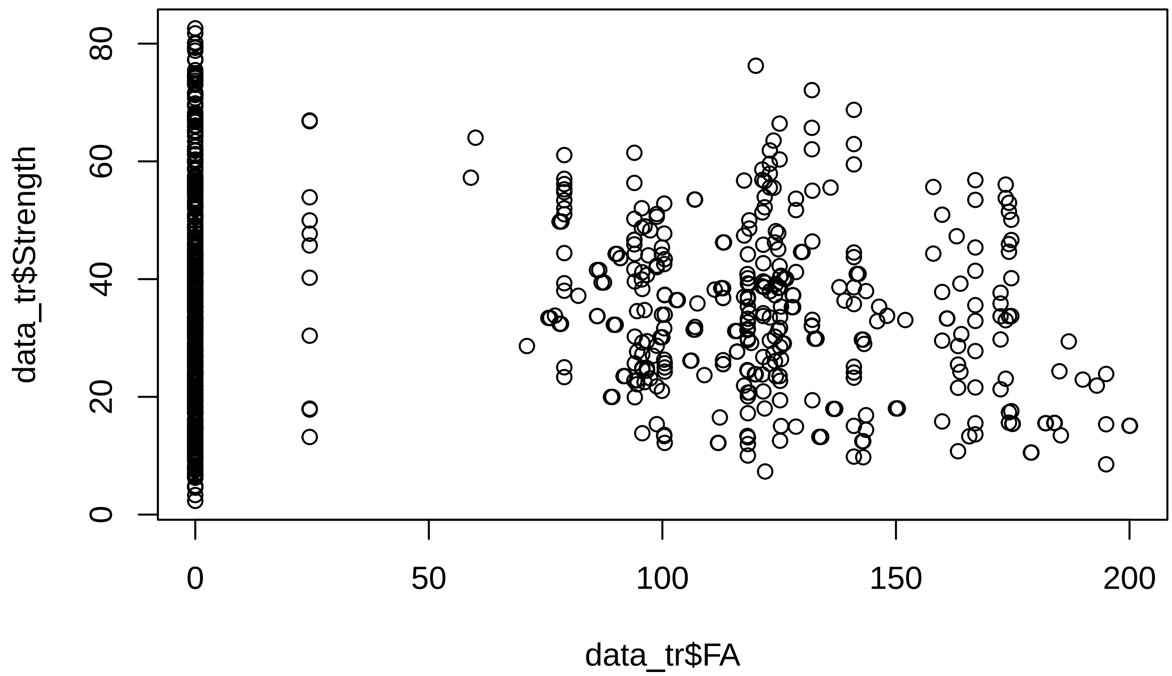
```
par(mfrow = c(1,1))
plot(data_tr$Cement, data_tr$Strength)
```



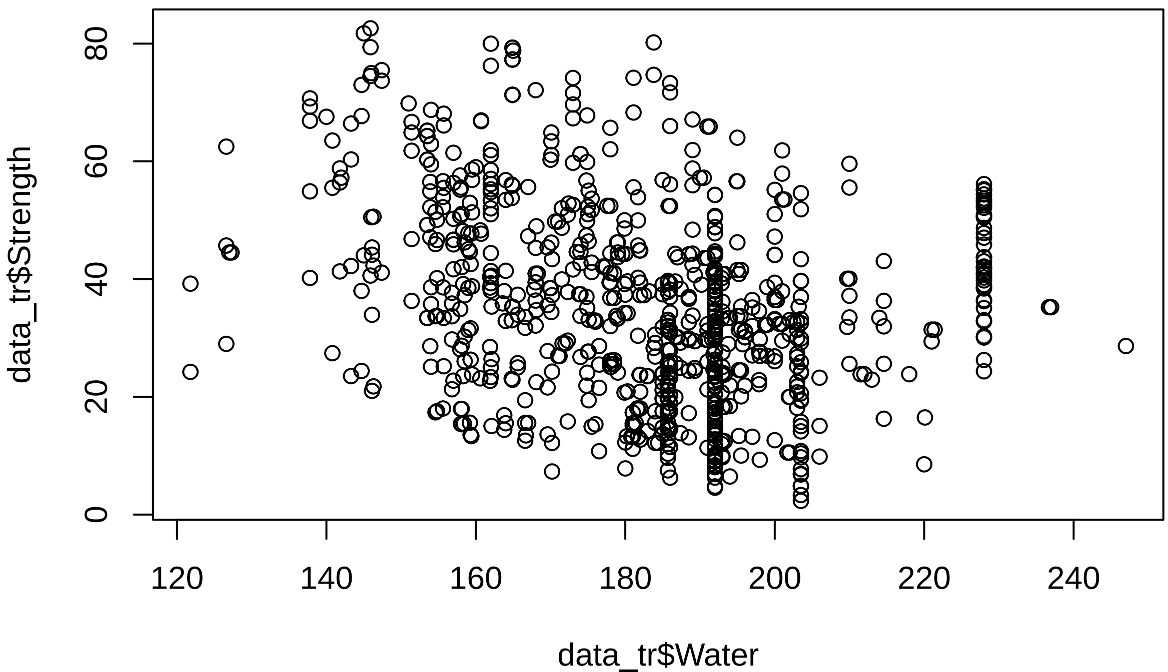

```
plot(data_tr$BFS,data_tr$Strength)
```



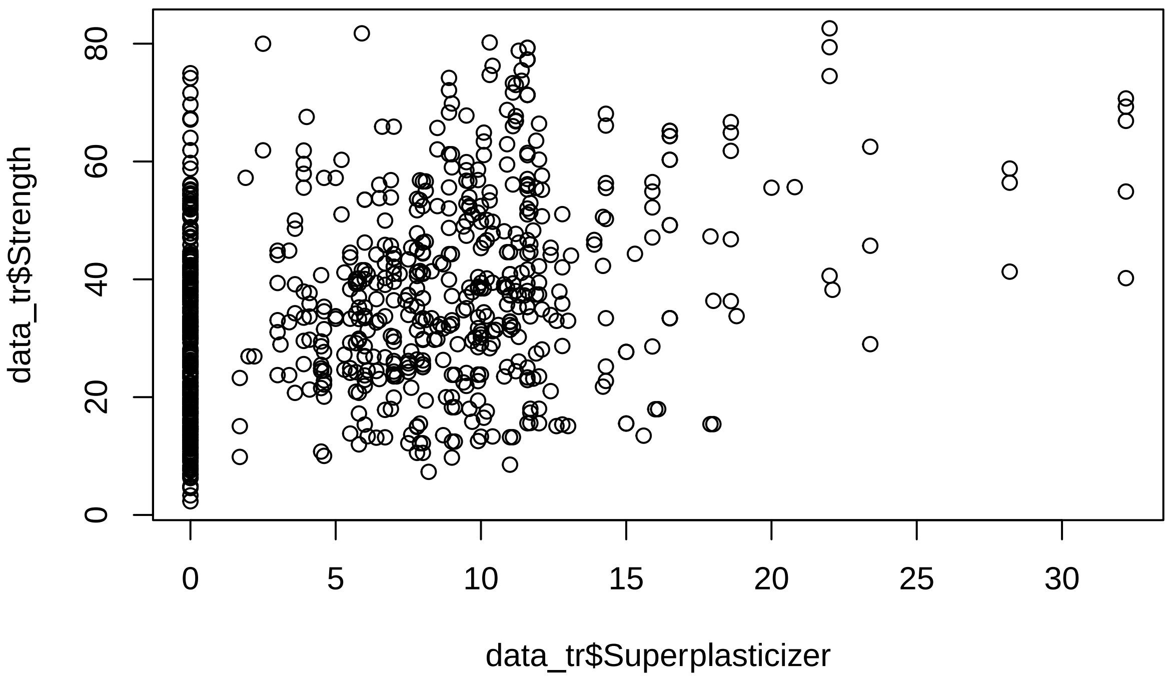

```
plot(data_tr$FA,data_tr$Strength)
```



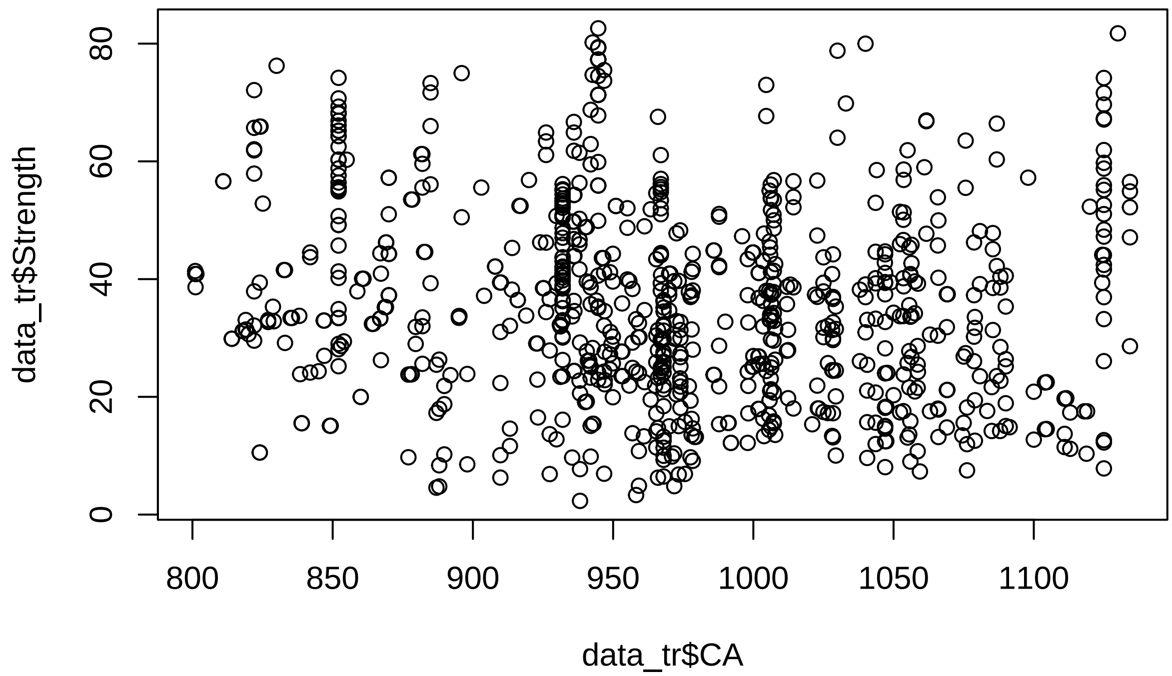

```
plot(data_tr$Water,data_tr$Strength)
```



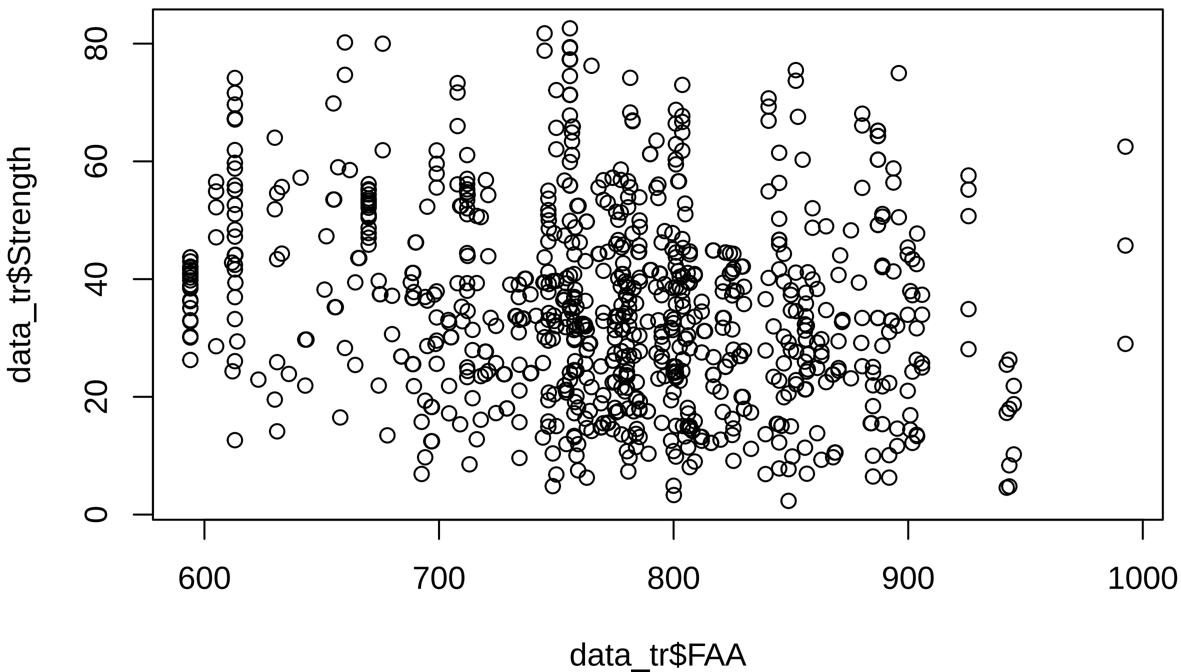

```
plot(data_tr$Superplasticizer,data_tr$Strength)
```



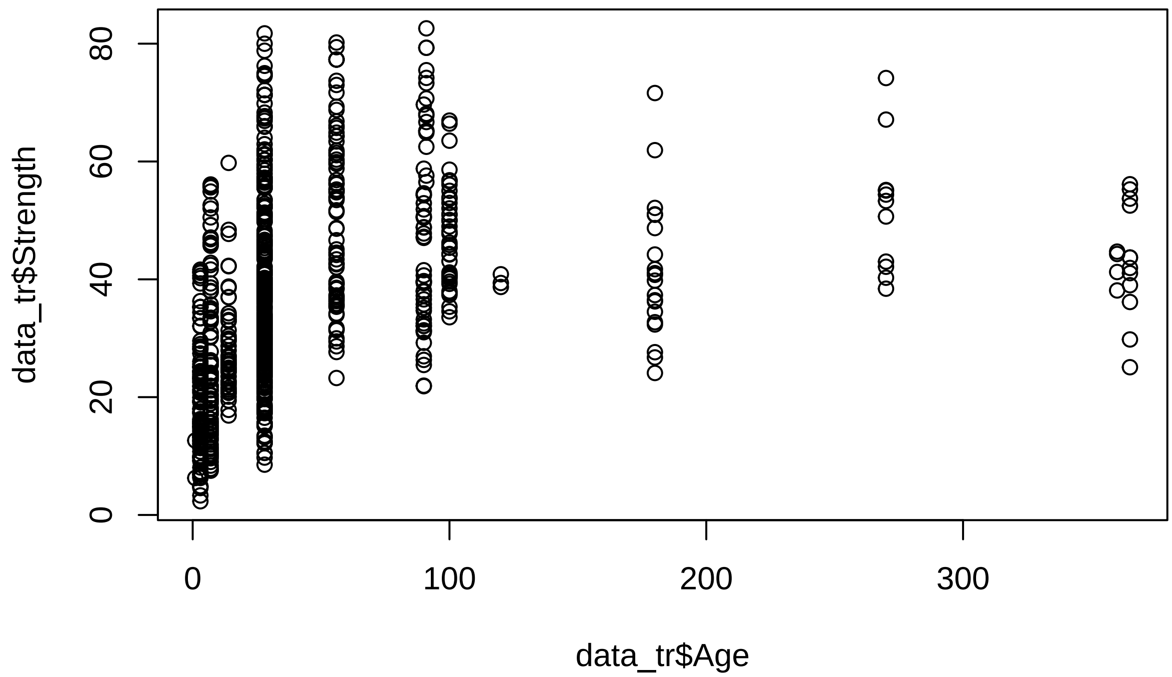

```
plot(data_tr$CA,data_tr$Strength)
```




```
plot(data_tr$FAA,data_tr$Strength)
```




```
plot(data_tr$Age,data_tr$Strength)
```

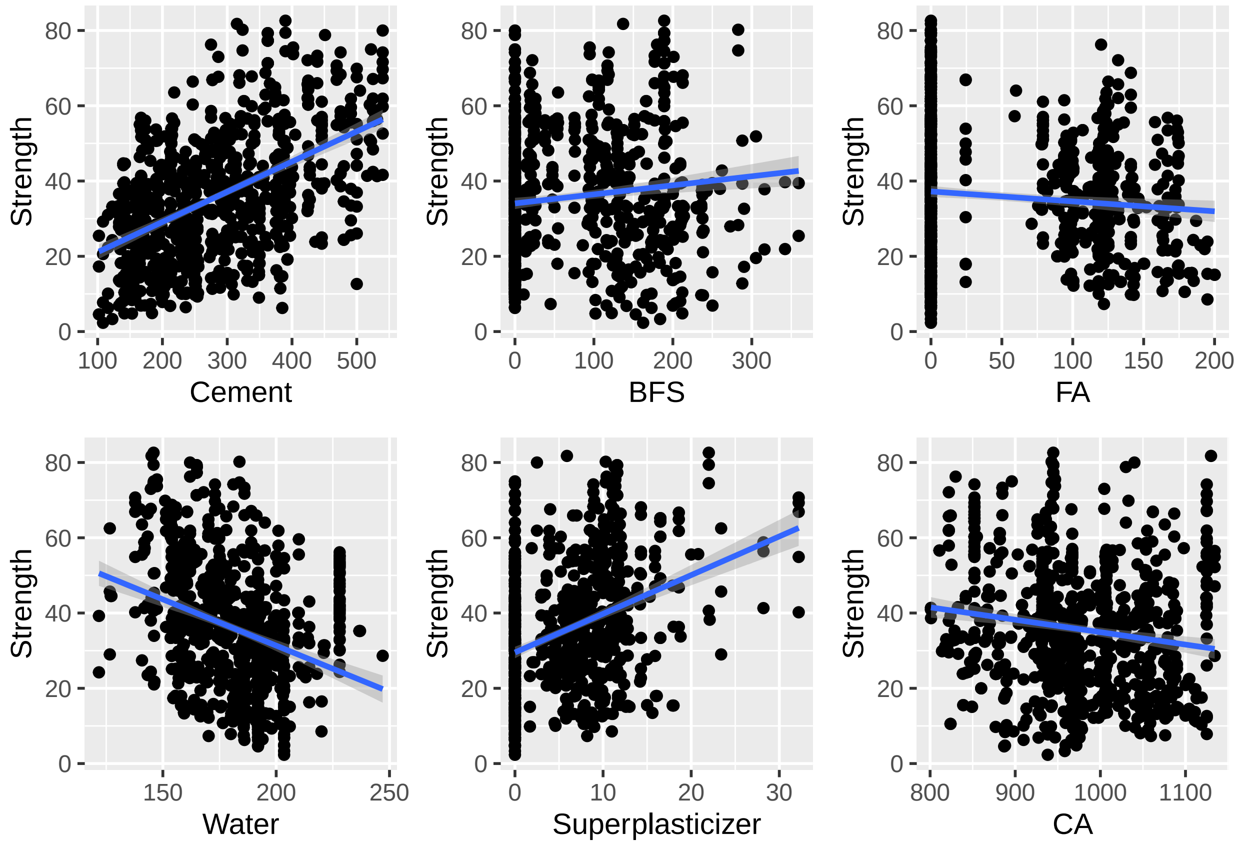



```
library ("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
require(ggplot2)
p1 <- ggplot(data_tr, aes(Cement, Strength)) + geom_point() + stat_smooth(method="lm")
p2 <- ggplot(data_tr, aes(BFS, Strength)) + geom_point() + stat_smooth(method="lm")
p3 <- ggplot(data_tr, aes(FA, Strength)) + geom_point() +stat_smooth(method="lm")
p4 <- ggplot(data_tr, aes(Water, Strength)) + geom_point() + stat_smooth( method="lm")
p5 <- ggplot(data_tr, aes(Superplasticizer, Strength)) + geom_point() + stat_smooth(method="lm")
p6 <- ggplot(data_tr, aes(CA, Strength)) + geom_point()+ stat_smooth( method="lm")
grid.arrange(p1, p2, p3, p4, p5, p6, nrow = 2)
```

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

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



We can learn that linear trends among **cement** and **ca** to **strength**. Thus, trying to fit a linear model is reasonable.

Inference: hypothesis testing

test one predictor, FAA

```
full <- lm(Strength ~., data_tr) # Full Model
summary(full)
```

```
##
## Call:
## lm(formula = Strength ~ ., data = data_tr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.181  -6.240   0.764   6.566  34.641
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.405094  29.692974   0.216   0.8293
## Cement              0.113998   0.009321  12.230  < 2e-16 ***
## BFS                 0.094211   0.011181   8.426  < 2e-16 ***
## FA                  0.082624   0.013820   5.979 3.37e-09 ***
```

```
## Water             -0.197441   0.045204  -4.368 1.42e-05 ***
## Superplasticizer  0.213105   0.105103   2.028   0.0429 *
## CA                0.007451   0.010486   0.711   0.4776
## FAA               0.010310   0.011948   0.863   0.3884
## Age               0.112841   0.005987  18.848  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.31 on 815 degrees of freedom
## Multiple R-squared:  0.6249, Adjusted R-squared:  0.6212
## F-statistic: 169.7 on 8 and 815 DF,  p-value: < 2.2e-16
```

```r
wofaa <- lm(Strength ~ .-FAA,
                  data_tr)

anova(wofaa, full)
```

```
## Analysis of Variance Table
##
## Model 1: Strength ~ (Cement + BFS + FA + Water + Superplasticizer + CA +
##     FAA + Age) - FAA
## Model 2: Strength ~ Cement + BFS + FA + Water + Superplasticizer + CA +
##     FAA + Age
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    816 86740
## 2    815 86661  1    79.179 0.7446 0.3884
```

P value is large, so we fail to reject the null hypothesis that Fine Aggregate = 0

test one predictor, CA

```r
woca <- lm(Strength ~ .-CA,
                  data_tr)
anova(woca, full)
```

```
## Analysis of Variance Table
##
## Model 1: Strength ~ (Cement + BFS + FA + Water + Superplasticizer + CA +
##     FAA + Age) - CA
## Model 2: Strength ~ Cement + BFS + FA + Water + Superplasticizer + CA +
##     FAA + Age
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    816 86715
## 2    815 86661  1    53.682 0.5048 0.4776
```

P value is smaller than 0.05, so we reject the null hypothesis that Coarse Aggregate = 0

test a group of variables:

```r
wocanfaa <- lm(Strength ~ Cement + BFS + FA + Water +
                    Superplasticizer + Age,data_tr)
anova(wocanfaa, full)
```

```
## Analysis of Variance Table
##
## Model 1: Strength ~ Cement + BFS + FA + Water + Superplasticizer + Age
## Model 2: Strength ~ Cement + BFS + FA + Water + Superplasticizer + CA +
##     FAA + Age
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    817 86740
## 2    815 86661  2    79.204 0.3724 0.6892
```

P value is large, so we so we fail to reject the null hypothesis that Coarse Aggregate and Fine Aggregate = 0, which matched with our BIC model.

Model selection

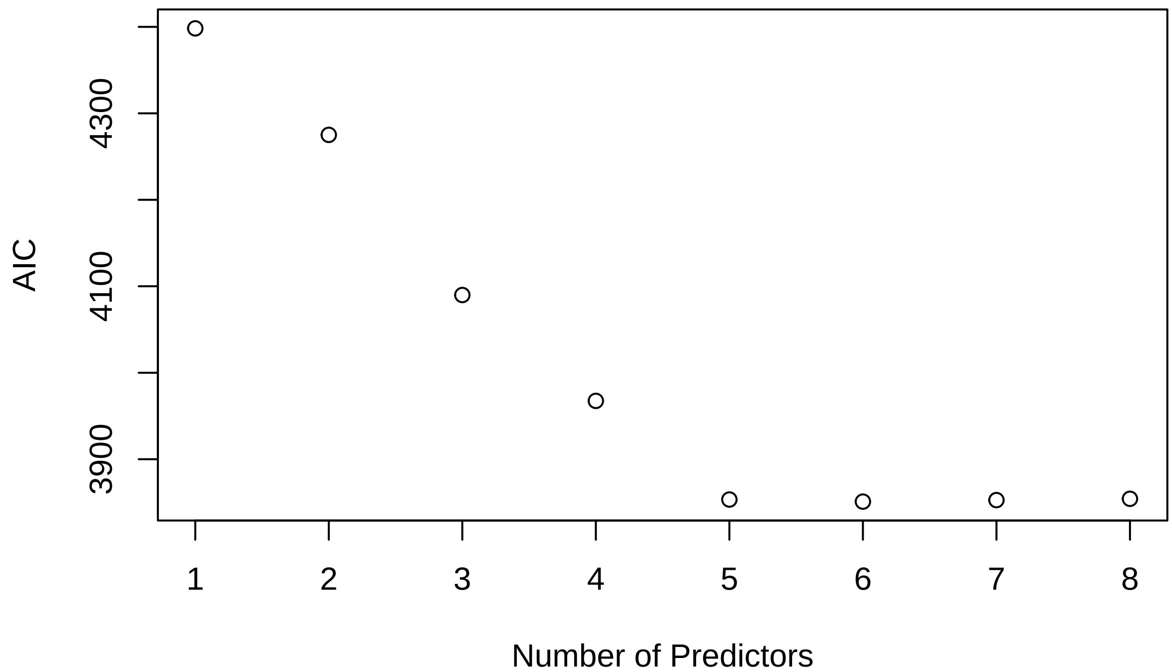
```r
require(leaps)
```

```
## Loading required package: leaps
```

```r
b <- regsubsets(Strength~.,data=data_tr)

rs  = summary(b)
rs$which
```

```
##   (Intercept) Cement   BFS    FA Water Superplasticizer    CA   FAA   Age
## 1        TRUE   TRUE FALSE FALSE FALSE            FALSE FALSE FALSE FALSE
## 2        TRUE   TRUE FALSE FALSE FALSE             TRUE FALSE FALSE FALSE
## 3        TRUE   TRUE FALSE FALSE FALSE             TRUE FALSE FALSE  TRUE
## 4        TRUE   TRUE  TRUE FALSE  TRUE            FALSE FALSE FALSE  TRUE
## 5        TRUE   TRUE  TRUE  TRUE  TRUE            FALSE FALSE FALSE  TRUE
## 6        TRUE   TRUE  TRUE  TRUE  TRUE             TRUE FALSE FALSE  TRUE
## 7        TRUE   TRUE  TRUE  TRUE  TRUE             TRUE FALSE  TRUE  TRUE
## 8        TRUE   TRUE  TRUE  TRUE  TRUE             TRUE  TRUE  TRUE  TRUE
```

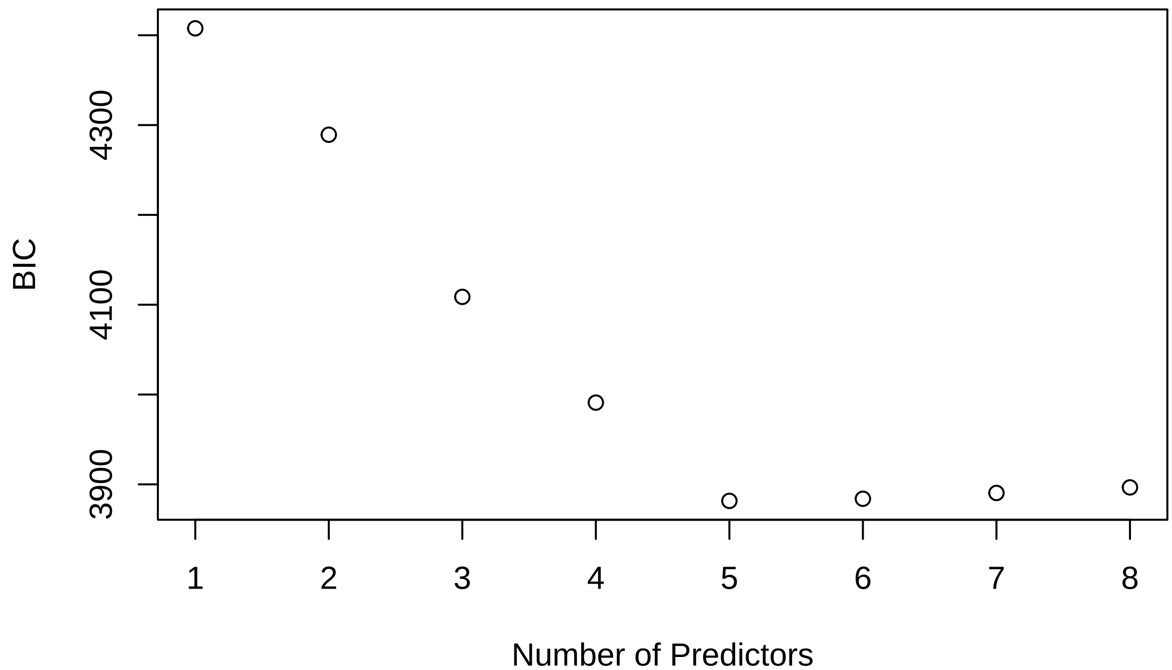
```r
n = 824

AIC <- n*log((rs$rss)/n) + (2:9)*2
plot(AIC ~ I(1:8), ylab="AIC", xlab="Number of Predictors")
```




```
which.min(AIC)
```

```
## [1] 6
```

```
BIC <- n*log(rs$rss/n) + (2:9)*log(n)
plot(BIC ~ I(1:8), ylab="BIC", xlab="Number of Predictors")
```




```
which.min(BIC)
```

```
## [1] 5
```

```
wocanfaa <- lm(Strength ~ Cement + BFS + FA + Water +
                    Age,data_tr)
```

Although our AIC and BIC do not match, we chose to continue with BIC.

Inference: Confidence Intervals

95% confidence interval for BIC model

```
confint(wocanfaa)
```

```
##                   2.5 %      97.5 %
## (Intercept) 27.27936981 43.64944244
## Cement       0.10226941  0.11901549
## BFS          0.08093052  0.10075465
## FA           0.06769326  0.09669597
## Water       -0.29690270 -0.22239783
## Age          0.10111553  0.12450822
```

0 is not in the any confidence interval for all predictors, this indicates that the null hypothesis that beta = 0 for any of them would be rejected at alpha = 5% level.

95% confidence interval for full model

```
confint(full)
```

```
##                          2.5 %      97.5 %
## (Intercept)      -51.878620215 64.68880858
## Cement             0.095700967  0.13229424
## BFS                0.072263921  0.11615732
## FA                 0.055496599  0.10975095
## Water             -0.286171842 -0.10871017
## Superplasticizer   0.006799795  0.41941015
## CA                -0.013132144  0.02803323
## FAA               -0.013142028  0.03376208
## Age                0.101089693  0.12459292
```

0 is in the confidence interval of CA and FAA, this indicates that the null hypothesis that beta = 0 for them would be rejected at alpha = 5% level. ## Both of the confidence interval at 95% supports our model selection

90% confidence interval for BIC model

```
confint(wocanfaa, level = 0.9)
```

```
##                     5 %        95 %
## (Intercept) 28.59769420 42.33111806
## Cement       0.10361802  0.11766688
## BFS          0.08252701  0.09915816
## FA           0.07002892  0.09436030
## Water       -0.29090263 -0.22839790
## Age          0.10299941  0.12262434
```

We calculated 90 percent confidence interval for beta, 0 is not included in any intervals, we can reject null hypothesis for any predictors at alpha = 0.9 ## They are significant at 0.9 level

90% confidence interval for full model

```
confint(full, level = .9)
```

```
##                            5 %        95 %
## (Intercept)      -42.491080347 55.30126871
## Cement             0.098647937  0.12934727
```

```
## BFS                  0.075798795  0.11262245
## FA                   0.059865872  0.10538168
## Water               -0.271880298 -0.12300172
## Superplasticizer     0.040028598  0.38618135
## CA                  -0.009816967  0.02471806
## FAA                 -0.009364693  0.02998475
## Age                  0.102982482  0.12270013
```

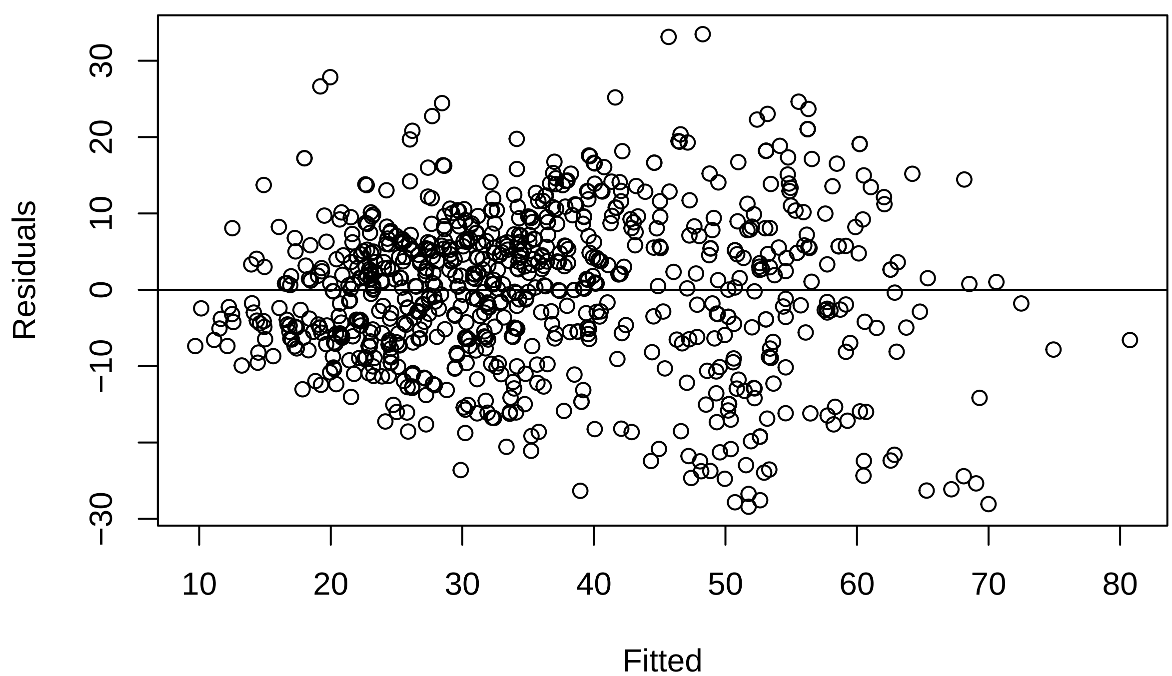
0 is in the confidence interval of CA and FAA

Since our research is based on 95% level, contradiction made between 95% and 90% confidence interval does not support our proceduire but it does not discourage our process either.

Diagnostics

Constant Variance

```
plot(fitted(wocanfaa), residuals(wocanfaa), xlab = "Fitted", ylab = "Residuals")
abline(h=0)
```




```
resi1 <- residuals(wocanfaa)
yhat1 <- fitted(wocanfaa)
summary(yhat1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   9.699  25.479  33.607  35.816  46.119  80.751
```

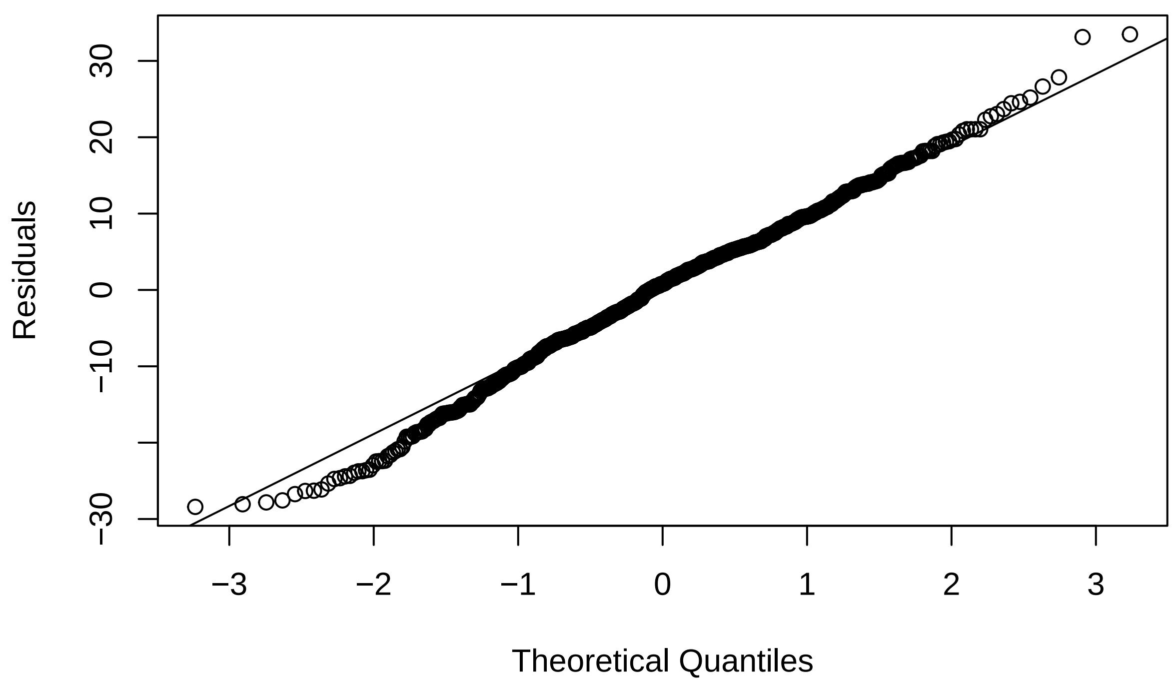
```
var.test(residuals(wocanfaa)[yhat1>36.15], residuals(wocanfaa)[yhat1<36.15])
```

```
##
##  F test to compare two variances
##
## data:  residuals(wocanfaa)[yhat1 > 36.15] and residuals(wocanfaa)[yhat1 < 36.15]
## F = 2.3784, num df = 346, denom df = 476, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.957867 2.898125
## sample estimates:
## ratio of variances
##           2.378379
```

Our constant variance test shows that there is significant difference between constants. In other words, the null hypothesis, variance is constant, is rejected.

normal errors

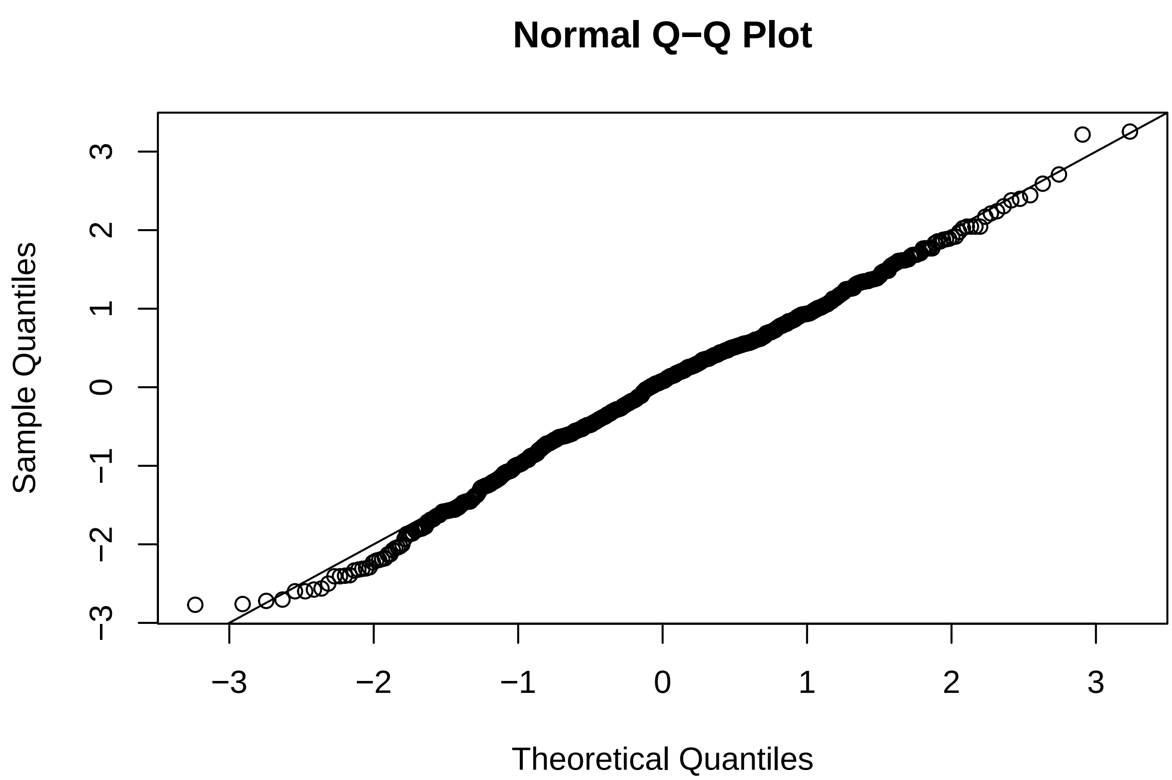
```
qqnorm(residuals(wocanfaa), ylab = "Residuals", main = "")
qqline(residuals(wocanfaa))
```



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

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(wocanfaa)
## W = 0.99482, p-value = 0.006674
```

```r
qqnorm(rstandard(wocanfaa))
abline(0,1)
```




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

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(wocanfaa)
## W = 0.99482, p-value = 0.006674
```

We have our P values less than 0.05. The null hypothesis, residuals are normal, is rejected.

Leverages and Outliers

```
# leverage points
n <- nrow(data_tr)
hatv <- hatvalues(wocanfaa)
p <- sum(hatv)
which(hatv > 2*p/n)
```

```
##  27  56  63  90  97 118 126 127 131 136 170 197 203 222 307 308 309 326 346 357
##  27  56  63  90  97 118 126 127 131 136 170 197 203 222 307 308 309 326 346 357
## 370 371 379 383 389 418 435 443 450 461 470 480 488 539 542 553 568 571 585 591
## 370 371 379 383 389 418 435 443 450 461 470 480 488 539 542 553 568 571 585 591
## 601 602 674 675 682 737 751 762 775 803
## 601 602 674 675 682 737 751 762 775 803
```

get outlier

```
n <- nrow(data_tr)
stud <- rstudent(wocanfaa)
stud[which.max(abs(stud))]
```

```
##      364
## 3.273975
```

```
qt(1-.05/(n*2),n-p-1)
```

```
## [1] 4.03121
```

```
which(abs(stud) > qt(1-.05/(n*2),n-p-1))
```

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

No outlier detected

```
x <- model.matrix(wocanfaa)[,-1]
vif(x)
```

```
##   Cement      BFS       FA    Water      Age
## 1.604532 1.447997 1.734676 1.193032 1.101126
```

```
max(vif(x))
```

```
## [1] 1.734676
```

Passed

Serial Correlation, Durbin Wtason test

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
dwtest(wocanfaa)
```

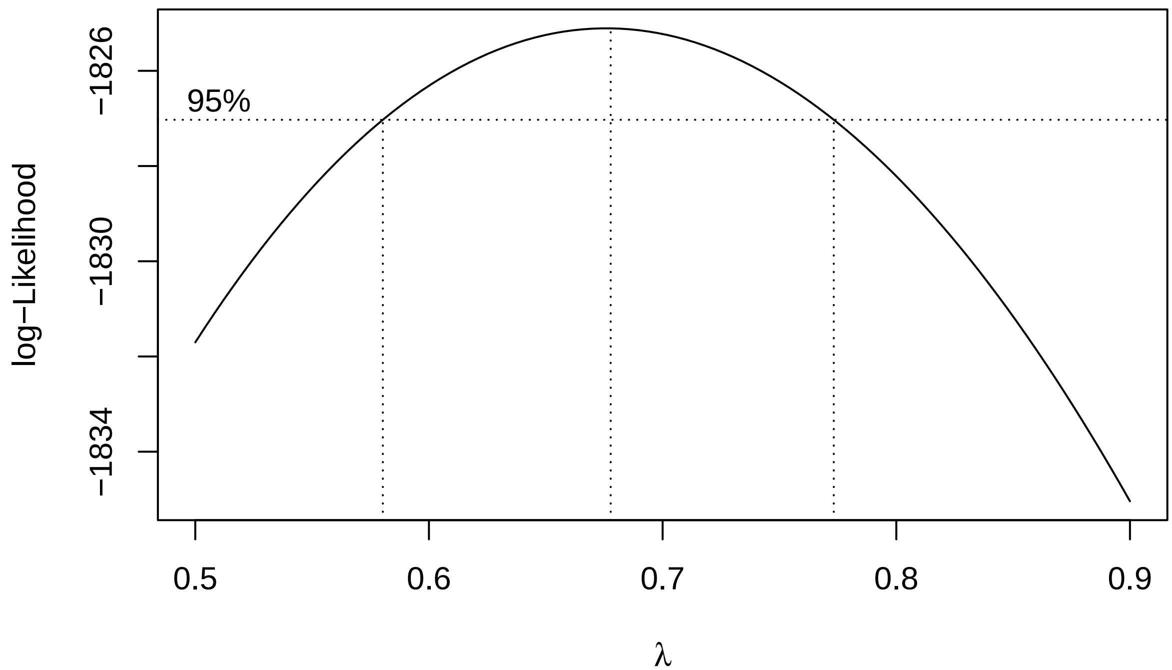
```
##
##  Durbin-Watson test
##
## data:  wocanfaa
## DW = 2.0569, p-value = 0.7929
## alternative hypothesis: true autocorrelation is greater than 0
```

Test Statistics with 2.0514 and P is greater than 0.05, fail to reject null

From previous diagnostics, we conclude that the transformation is needed

Transformation

```r
boxcox(wocanfaa, plotit = T, lambda = seq(0.5,0.9,by = 0.1))
```



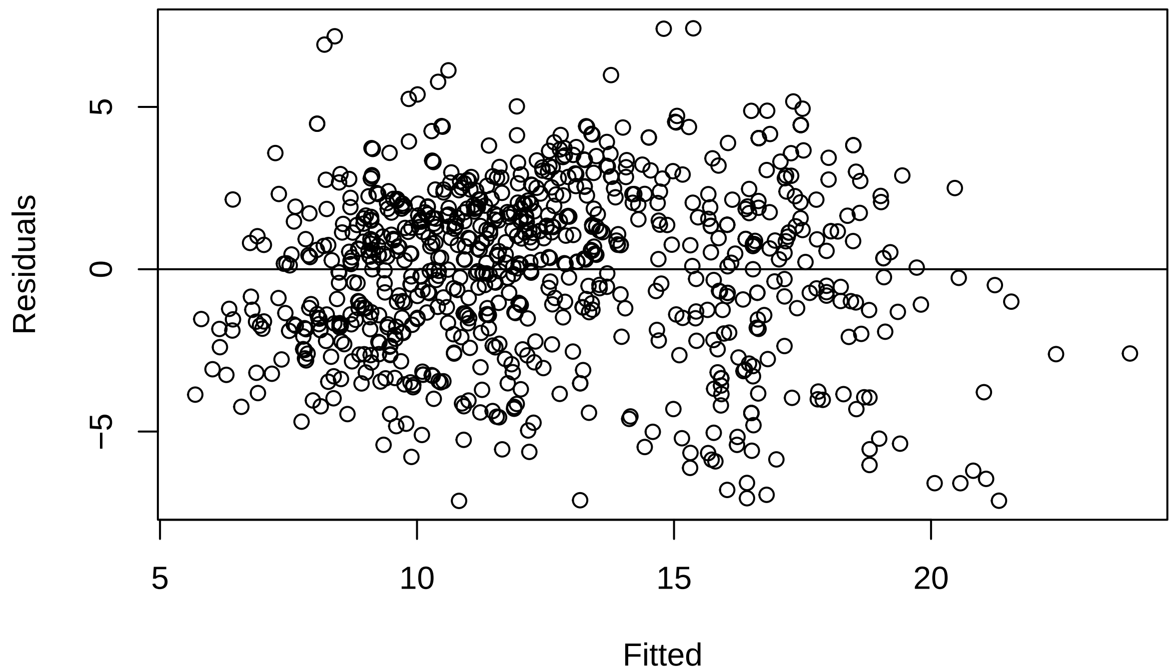
We see that the interval is approximately from 0.61 to 0.81, we can choose 0.7 as our lambda value

```
trans=(lm(Strength^0.71 ~ Cement + Water + BFS + FA + Age, data_tr))
#trans = lm(log(Strength) ~  Cement + Water + Superplasticizer + BFS + FA + Age, data)
#trans = lm(Strength~ polym(Cement, Water, Superplasticizer, BFS, FA, Age,degree = 2), data)
```

2nd round of diagnostic

Constant Variance #2

```
plot(fitted(trans), residuals(trans), xlab = "Fitted", ylab = "Residuals")
abline(h=0)
```



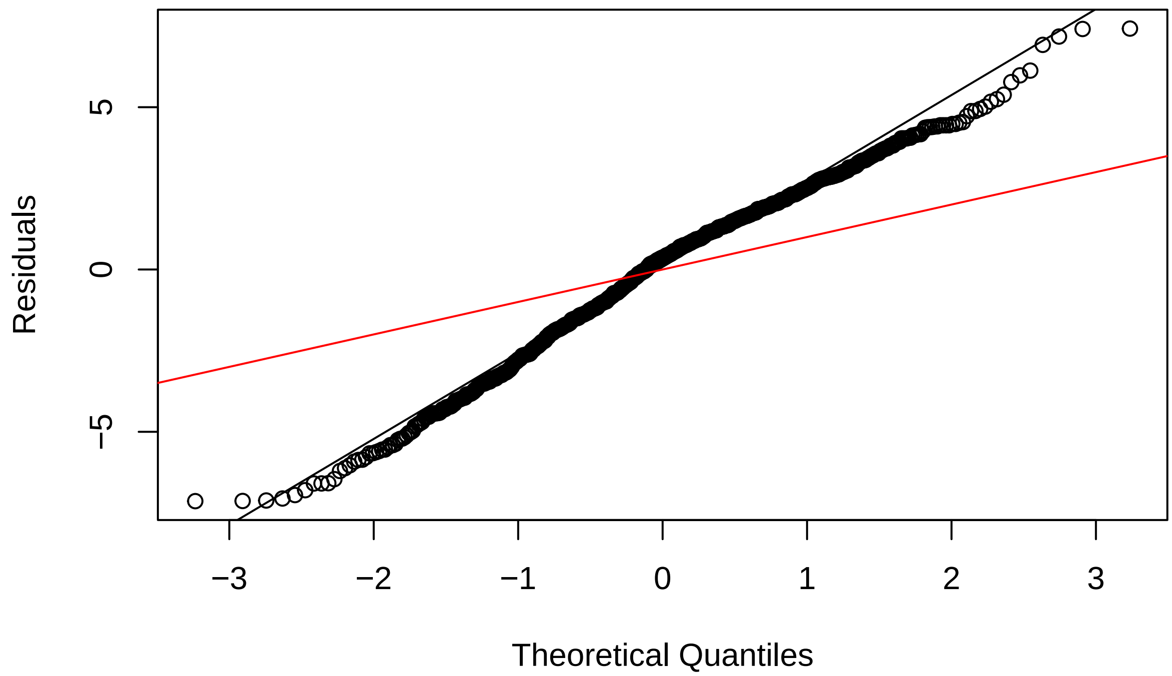

```
resi1 <- residuals(trans)
yhat1 <- fitted(trans)


var.test(residuals(trans)[yhat1>mean(yhat1)], residuals(trans)[yhat1<mean(yhat1)])
```

```
##
##  F test to compare two variances
##
## data:  residuals(trans)[yhat1 > mean(yhat1)] and residuals(trans)[yhat1 < mean(yhat1)]
## F = 1.6346, num df = 355, denom df = 467, p-value = 6.818e-07
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.346135 1.990028
## sample estimates:
## ratio of variances
##           1.634561
```

Our constant variance test shows thet there is significant difference between constants. In other words, the null hypothesis, variance is constant, is rejected.

normal errors #2

```
qqnorm(residuals(trans), ylab = "Residuals", main = "")
qqline(residuals(trans))
abline(0,1, col ="red")
```




```
shapiro.test(residuals(trans))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(trans)
## W = 0.98861, p-value = 5.115e-06
```

We have our P values less than 0.05. The null hypothesis, residuals are normal, is rejected. Test failed

Leverages and Outliers #2

```
n <- nrow(data_tr)
hatv <- hatvalues(trans)
p <- sum(hatv)
which(hatv > 2*p/n)
```

```
##  27  56  63  90  97 118 126 127 131 136 170 197 203 222 307 308 309 326 346 357
##  27  56  63  90  97 118 126 127 131 136 170 197 203 222 307 308 309 326 346 357
## 370 371 379 383 389 418 435 443 450 461 470 480 488 539 542 553 568 571 585 591
## 370 371 379 383 389 418 435 443 450 461 470 480 488 539 542 553 568 571 585 591
## 601 602 674 675 682 737 751 762 775 803
## 601 602 674 675 682 737 751 762 775 803
```

```
n <- nrow(data_tr)
stud <- rstudent(trans)
stud[which.max(abs(stud))]
```

```
##      364
## 2.842896
```

```
qt(1-.05/(n*2),n-p-1)
```

```
## [1] 4.03121
```

```
which(abs(stud) > qt(1-.05/(n*2),n-p-1))
```

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

Still, no outliers detected

Serial Correlation, Durbin Watson test #2

```
library(lmtest)
dwtest(trans)
```

```
##
##  Durbin-Watson test
##
## data:  trans
## DW = 2.0417, p-value = 0.7248
## alternative hypothesis: true autocorrelation is greater than 0
```

p-value is greater than 0.05,test passed

```
x <- model.matrix(trans)[,-1]
vif(x)
```

```
##   Cement    Water      BFS       FA      Age
## 1.604532 1.193032 1.447997 1.734676 1.101126
```

```
max(vif(x))
```

```
## [1] 1.734676
```

Passed

Some conclusion…

```
###prediction
trans_final=(lm(Strength^0.7 ~ Cement + Water + Superplasticizer + BFS + FA + Age, data_tt))
x <- model.matrix(trans_final)
x0 <- apply(x,2,median) # get median characteristics

pred1 <- predict(trans_final, data.frame(t(x0)), interval = "p")
pred1
```

```
##        fit      lwr      upr
## 1 9.306706 3.942732 14.67068
```

```
confident1 <- predict(trans_final, data.frame(t(x0)), interval = "c")
confident1
```

```
##        fit      lwr      upr
## 1 9.306706 8.422253 10.19116
```

The prediction is based on the model that could not pass the diagnostic, our transformation failed. It is not a good model to predict the strength of concrete.