MA4142: Regression and Multivariate Analysis

Exploring Simple Regression

Authors:

Tanmay Goyal - AI20BTECH11021, Tanay Yadav - AI20BTECH11026

- **Installing and loading the libraries**

In [ ]:
```
install.packages("mltools")
install.packages("psych")
library(mltools)
library(data.table)
library(psych)
```

- **Reading the Dataset**

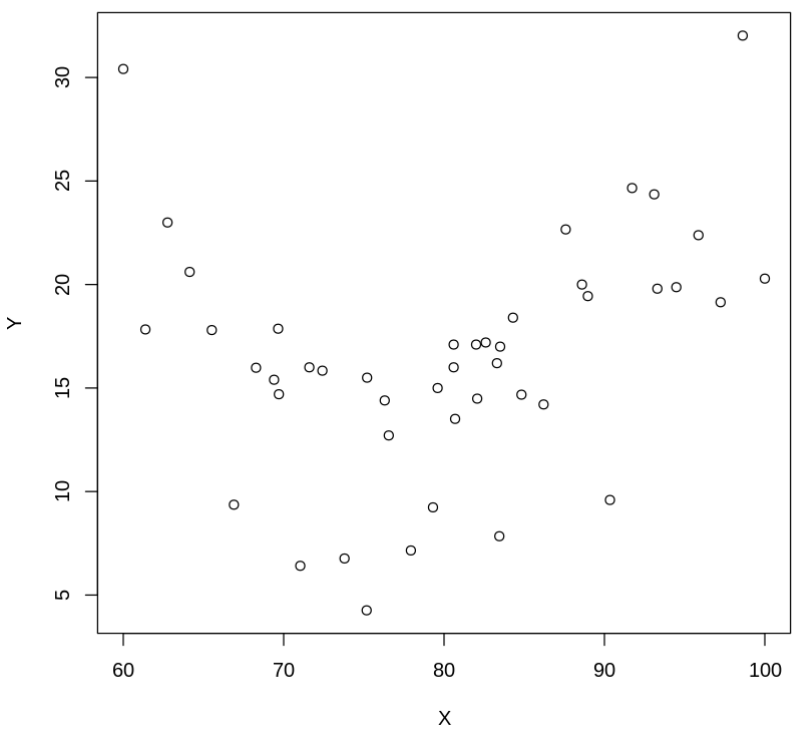
In [ ]:
```
data = read.csv("Crickets.csv")
head(data)
```

A data.frame: 6 × 3

	X.1	X	Y
	<int>	<dbl>	<dbl>
1	0	88.6	20.0
2	1	71.6	16.0
3	2	93.3	19.8
4	3	84.3	18.4
5	4	80.6	17.1
6	5	75.2	15.5

- **Visualizing the dataset**

In [ ]:
```
plot( Y ~ X , data = data)
```



- **Fitting a linear model ($y = ax + b$)**

In [ ]:
```
model = lm(Y ~ X, data = data)
model
```

```
Call:
lm(formula = Y ~ X, data = data)

Coefficients:
(Intercept)            X
     5.6729       0.1358
```

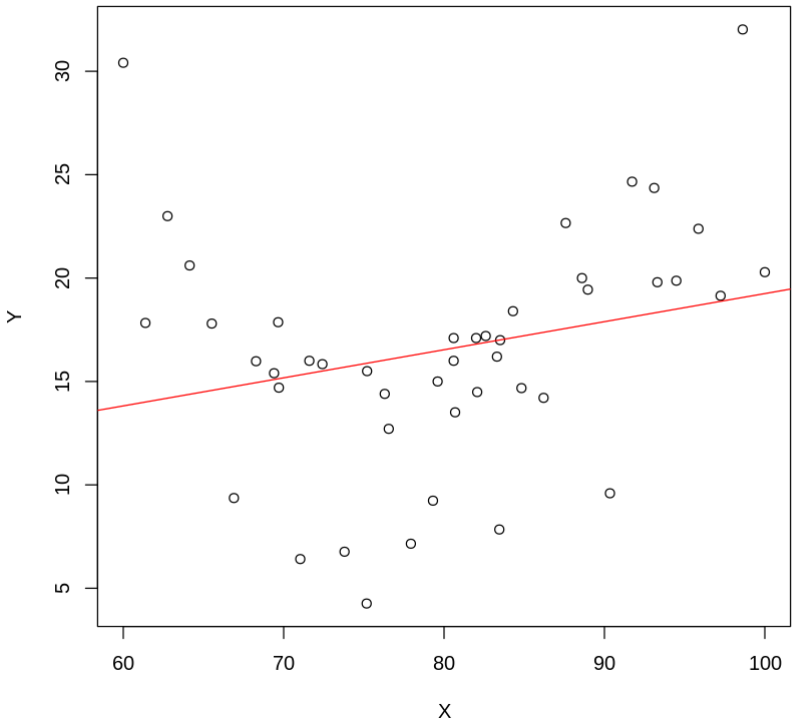
- **Summarizing the linear model**

In [ ]:
```
summary(model)
```

```
Call:
lm(formula = Y ~ X, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-11.6178  -2.5117   0.3036   2.2965  16.5917

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.67290    6.54598   0.867    0.391
X            0.13579    0.08112   1.674    0.101

Residual standard error: 5.682 on 43 degrees of freedom
Multiple R-squared:  0.06117,   Adjusted R-squared:  0.03934
F-statistic: 2.802 on 1 and 43 DF,  p-value: 0.1014
```

- **Visualizing the model**

In [ ]:
```
plot(Y ~ X, data=data) + abline(lm(Y ~ X, data=data) , col = 'red')
```



- **Fitting a 2nd-degree polynomial (** $y = ax^2 + bx + c$ **)**

In [ ]:
```
model2 = lm(Y ~ X + I(X^2) , data = data)
model2
```

```
Call:
lm(formula = Y ~ X + I(X^2), data = data)

Coefficients:
(Intercept)            X       I(X^2)
  193.74878     -4.64715      0.02989
```

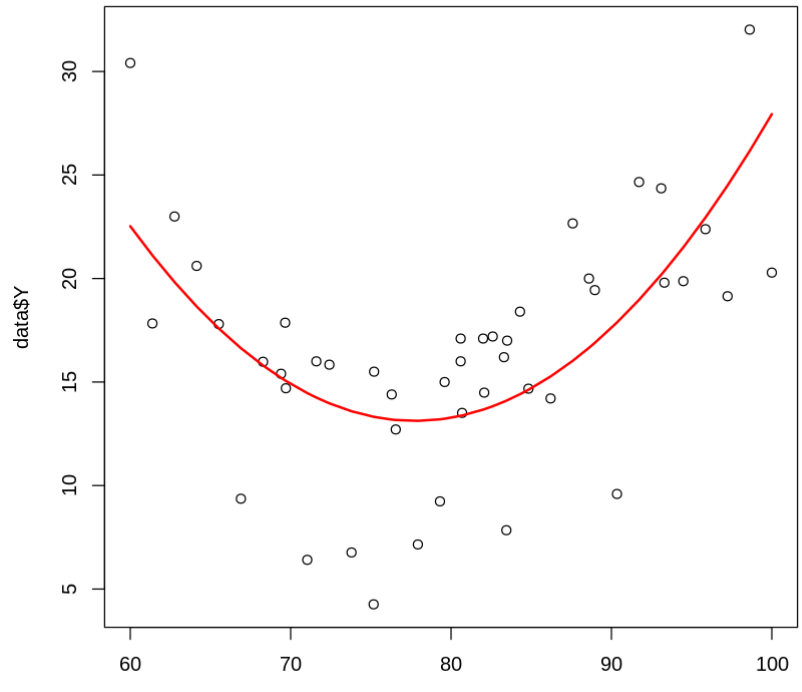
- **Summarizing the polynomial model**

In [ ]:
```
summary(model2)
```

```
Call:
lm(formula = Y ~ X + I(X^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0580 -1.6343  0.8026  2.8829  7.8854

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 193.748779  35.625451   5.438 2.54e-06 ***
X            -4.647150   0.898841  -5.170 6.12e-06 ***
I(X^2)        0.029891   0.005603   5.335 3.57e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.439 on 42 degrees of freedom
Multiple R-squared:  0.4404,    Adjusted R-squared:  0.4137
F-statistic: 16.52 on 2 and 42 DF,  p-value: 5.084e-06
```

- **Predicting and visualizing the polynomial model**

In [ ]:
```
pred <- predict(model2)
ix <- sort(data$X ,index.return=T)$ix

plot(data$X , data$Y) + lines(data$X[ix], pred[ix], col='red', lwd=2)
```



- **Fitting a 3rd-degree polynomial model ($y = ax^3 + bx^2 + cx + d$)**

In [ ]:
```
model3 = lm(Y ~ X + I(X^2) + I(X^3) , data = data)
model3
```

```
Call:
lm(formula = Y ~ X + I(X^2) + I(X^3), data = data)

Coefficients:
(Intercept)            X       I(X^2)       I(X^3)
 857.812862   -30.163313     0.352522    -0.001343
```

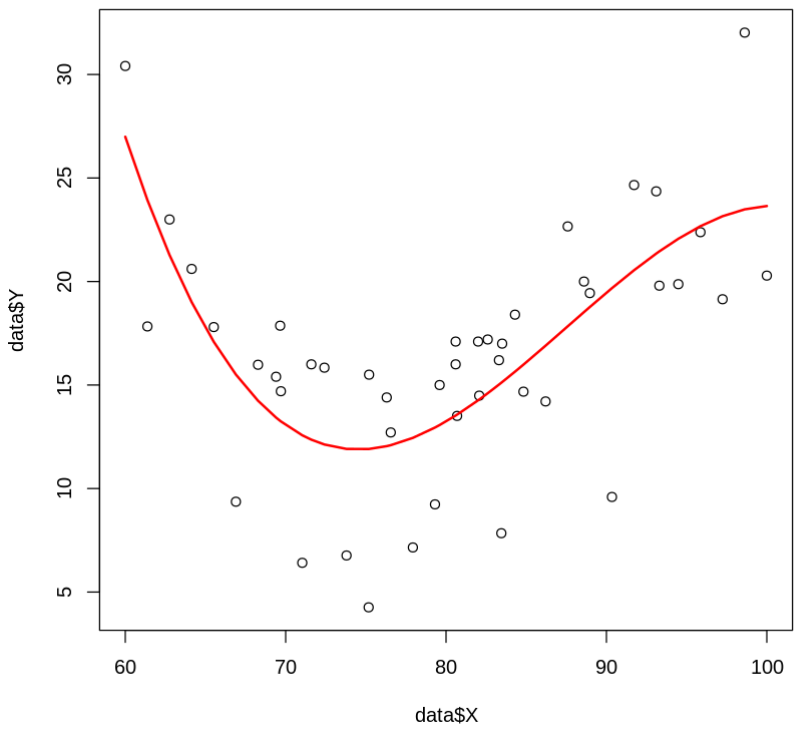
- **Summarizing the polynomial model**

In [ ]:
```
summary(model3)
```

```
Call:
lm(formula = Y ~ X + I(X^2) + I(X^3), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-10.087  -2.694   1.457   2.754   8.537

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 857.812862 252.909588   3.392  0.00155 **
X           -30.163313   9.669770  -3.119  0.00331 **
I(X^2)        0.352522   0.121916   2.892  0.00611 **
I(X^3)       -0.001343   0.000507  -2.649  0.01142 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.151 on 41 degrees of freedom
Multiple R-squared:  0.5221,    Adjusted R-squared:  0.4872
F-statistic: 14.93 on 3 and 41 DF,  p-value: 1.023e-06
```

- **Predicting and visualizing the polynomial model**

In [ ]:
```
pred <- predict(model3)
ix <- sort(data$X ,index.return=T)$ix

plot(data$X , data$Y) + lines(data$X[ix], pred[ix], col='red', lwd=2)
```



- **Fitting a 4th-degree polynomial model ($y = ax^4 + bx^3 + cx^2 + dx + e$)**

In [ ]:
```
model4 = lm(Y ~ X + I(X^2) + I(X^3) + I(X^4) , data = data)
model4
```

```
Call:
lm(formula = Y ~ X + I(X^2) + I(X^3) + I(X^4), data = data)

Coefficients:
(Intercept)            X       I(X^2)       I(X^3)       I(X^4)
  1.198e+03   -4.770e+01    6.880e-01   -4.169e-03    8.850e-06
```

- **Summarizing the model**

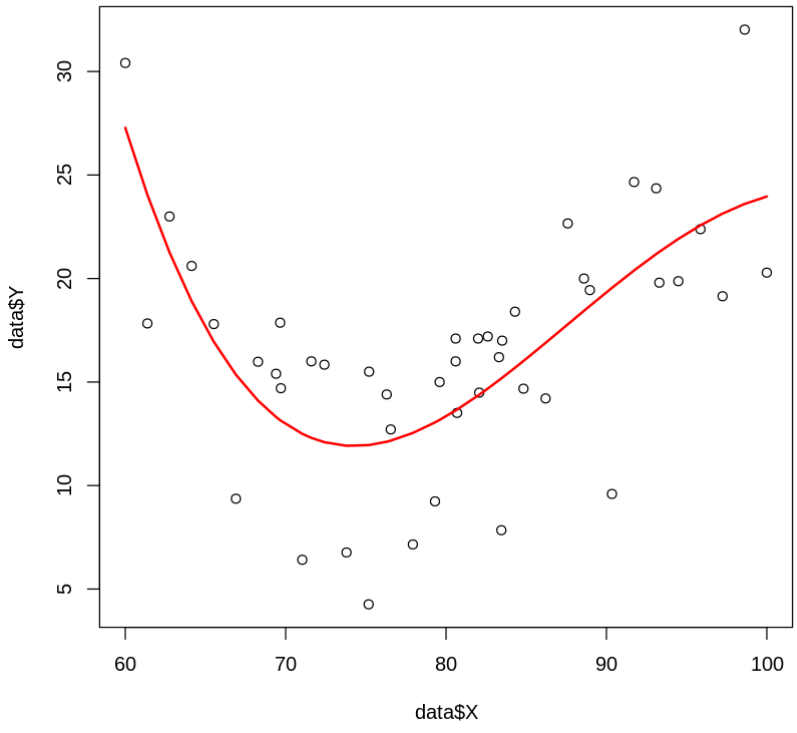
In [ ]:
```
summary(model4)
```

```
Call:
lm(formula = Y ~ X + I(X^2) + I(X^3) + I(X^4), data = data)

Residuals:
   Min     1Q Median     3Q    Max
-9.956 -2.685  1.560  2.710  8.418

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.198e+03  1.881e+03   0.637    0.528
X           -4.770e+01  9.648e+01  -0.494    0.624
I(X^2)       6.880e-01  1.841e+00   0.374    0.711
I(X^3)      -4.169e-03  1.548e-02  -0.269    0.789
I(X^4)       8.850e-06  4.844e-05   0.183    0.856

Residual standard error: 4.201 on 40 degrees of freedom
Multiple R-squared:  0.5225,    Adjusted R-squared:  0.4748
F-statistic: 10.94 on 4 and 40 DF,  p-value: 4.343e-06
```

The 3rd-degree polynomial model shows a better r-squared value as compared to the 4th-degree polynomial regression model. Hence, it fits the dataset better.

- **Predicting and visualizing the polynomial model**

In [ ]:
```
pred <- predict(model4)
ix <- sort(data$X ,index.return=T)$ix

plot(data$X , data$Y) + lines(data$X[ix], pred[ix], col='red', lwd=2)
```


data$Y
data$X
5
10
15
20
25
30
60
70
80
90
100

Mutiple Regression and Regression Diagnostics

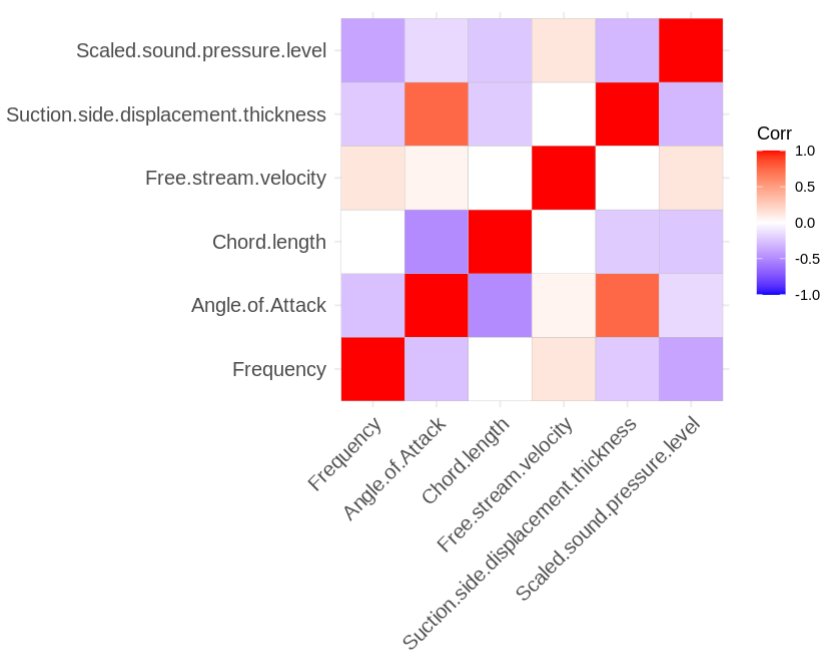
- **Installing and loading the required libraries**

In [ ]:
```
install.packages("ggcorrplot")
install.packages("psych")
install.packages("dplyr")
install.packages("ggpubr")
install.packages("regclass")
install.packages("lmtest")
install.packages("ridge")
```

In [ ]:
```
library(ggcorrplot)
library(psych)
library(dplyr)
library(ggpubr)
library(regclass)
library(lmtest)
# library(ridge)
```

- **Finding the Correlation Plot**

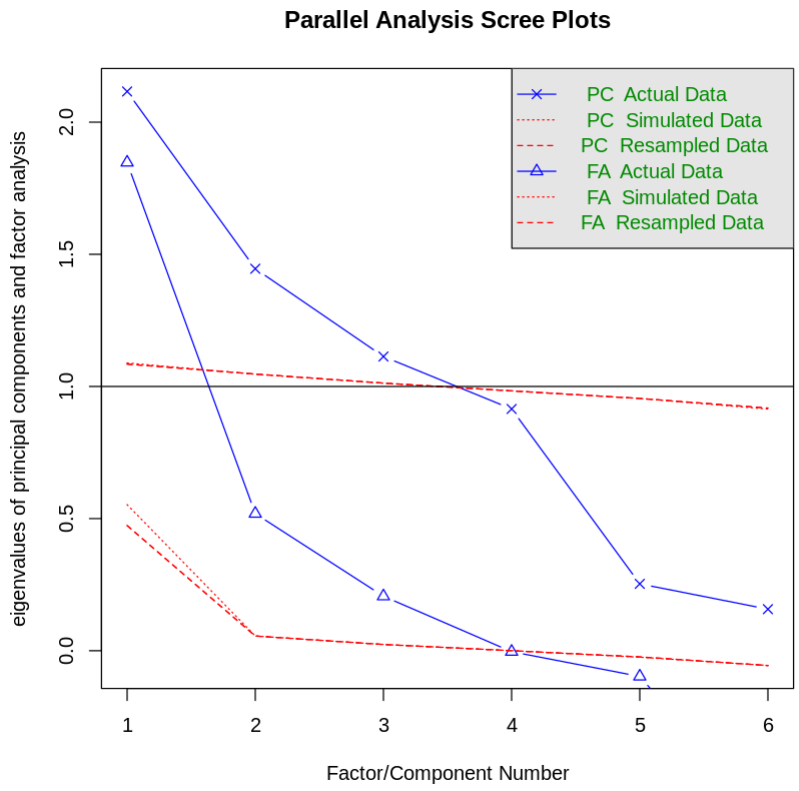
In [ ]:
```
data = read.csv("airfoil_self_noise.csv")
data_corr = cor(data)
ggcorrplot(data_corr)
```



We see that the variable Free Stream Velocity is uncorrelated with other variables. This shall play an inmportant role during Factor Analysis.

In [ ]:
```
fa.parallel(data, n.iter=100)
```

```
Warning message in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
“The estimated weights for the factor scores are probably incorrect.  Try a different factor score estimation method.”
Warning message in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, :
“An ultra-Heywood case was detected.  Examine the results carefully”
Parallel analysis suggests that the number of factors =  3  and the number of components =  3
```

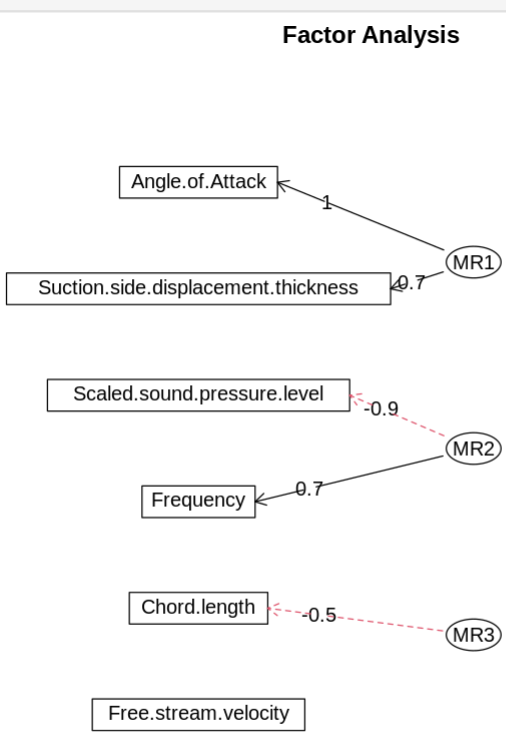


Principal Components predicts 3 components while Factor Analysis predicts 3 factors.

In [ ]:
```
fa = fa(data , nfactors = 3 , method = "both")
fa$loadings
```

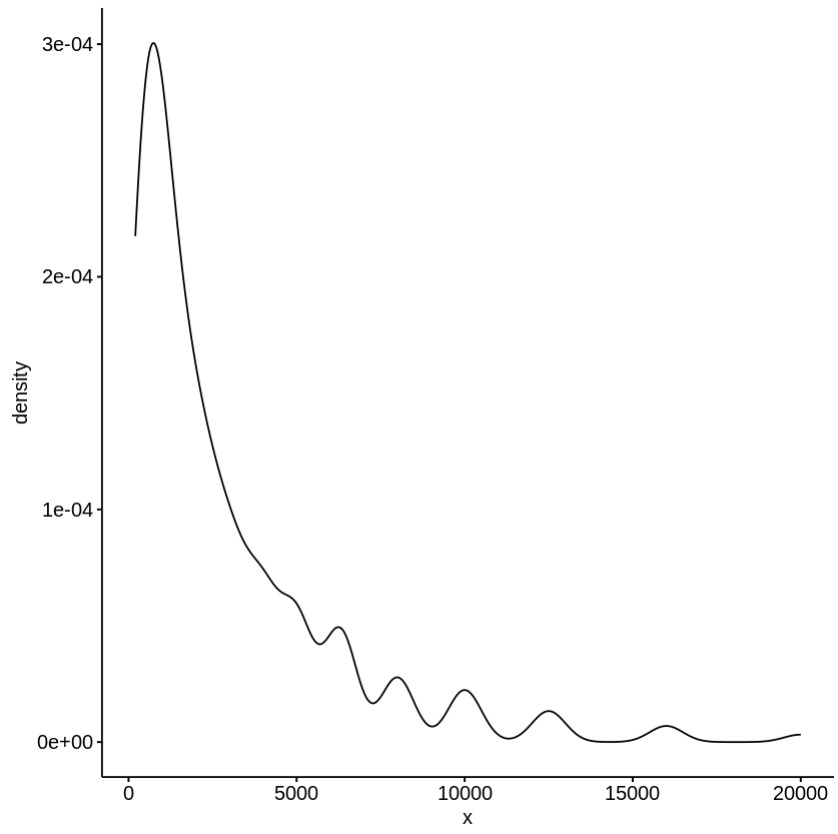
```
Loading required namespace: GPArotation

Warning message in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, :
“I am sorry, to do these rotations requires the GPArotation package to be installed”
Warning message in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
“The estimated weights for the factor scores are probably incorrect.  Try a different factor score estimation method.”
Warning message in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, :
“An ultra-Heywood case was detected.  Examine the results carefully”
Loadings:
                                  MR1    MR2    MR3
Frequency                         -0.423  0.705  0.573
Angle.of.Attack                    0.991         0.180
Chord.length                      -0.415  0.130 -0.451
Free.stream.velocity                             0.186
Suction.side.displacement.thickness  0.750  0.195
Scaled.sound.pressure.level       -0.139 -0.922  0.368

                 MR1   MR2   MR3
SS loadings    1.914 1.409 0.742
Proportion Var 0.319 0.235 0.124
Cumulative Var 0.319 0.554 0.678
```

In [ ]:
```
fa.diagram(fa$loadings)
```



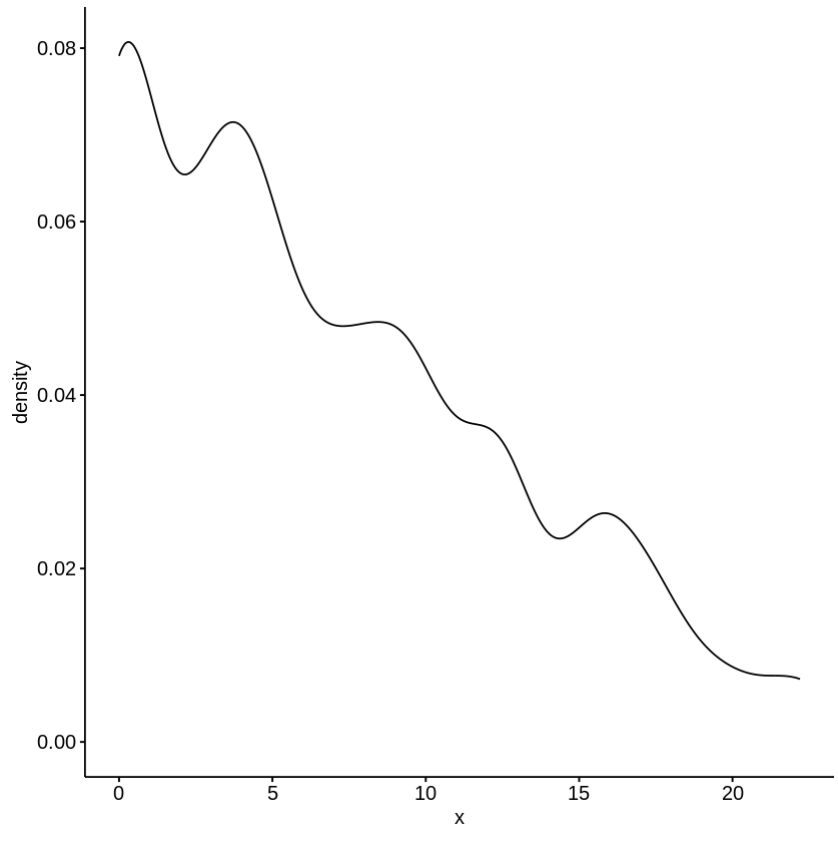
- **Density Graphs for each feature variable**

In [ ]:
```
ggdensity(data$Frequency)
```

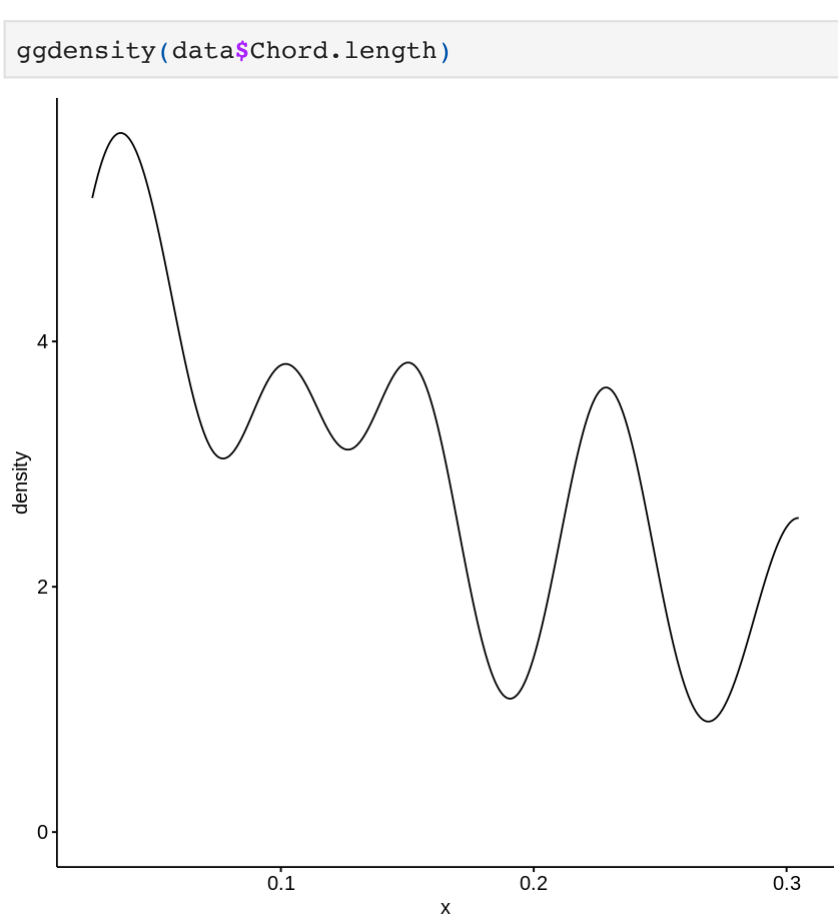


This looks like a right-tailed graph. (Important for when we do Box-Cox Transformation.)

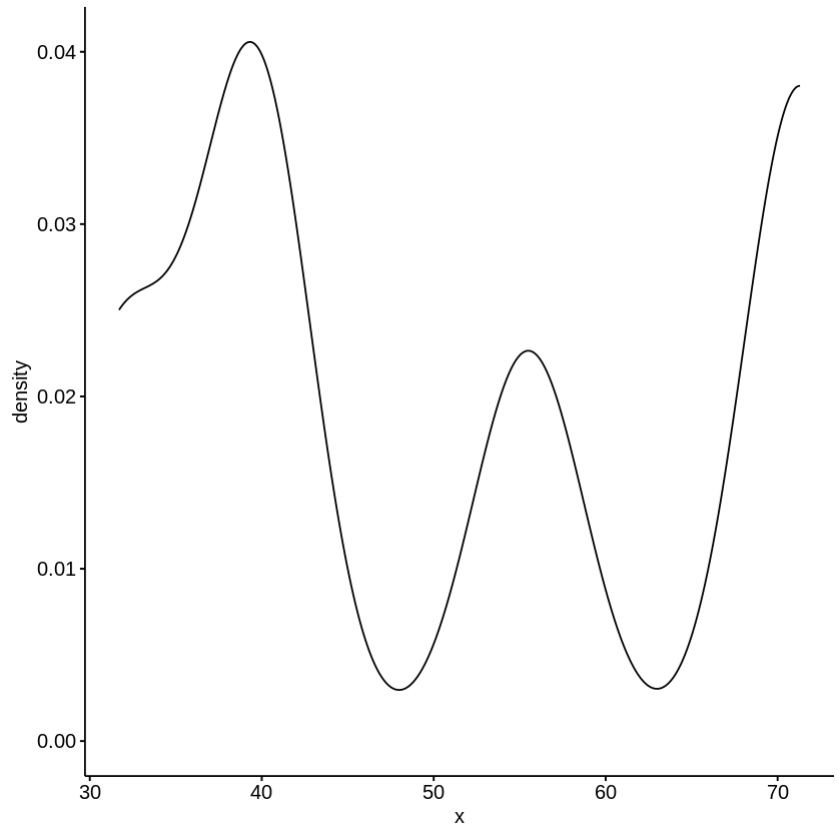
In [ ]:
```
ggdensity(data$Angle.of.Attack)
```



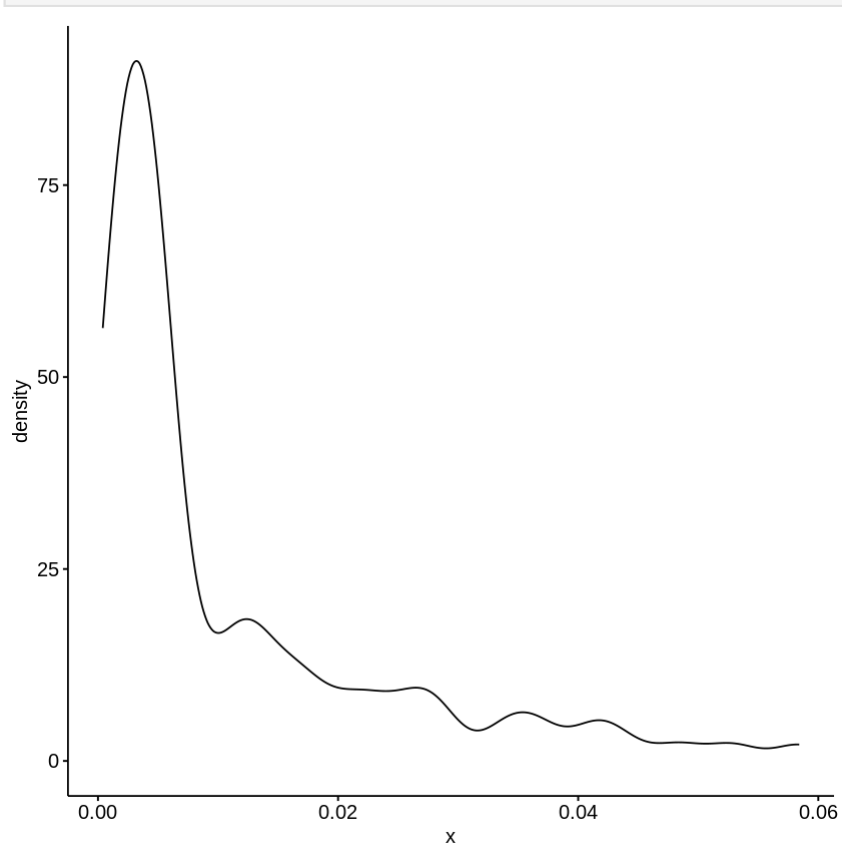
In [ ]:
```
ggdensity(data$Chord.length)
```



In [ ]:
```
ggdensity(data$Free.stream.velocity)
```



In [ ]:
```
ggdensity(data$Suction.side.displacement.thickness)
```



Again, this looks like a right-tailed graph.

We now perform Shapiro-Wilk's test for checking normality.

In [ ]:
```
shapiro.test(data$Frequency)
```

```
        Shapiro-Wilk normality test

data:  data$Frequency
W = 0.7635, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(data$Angle.of.Attack)
```

```
        Shapiro-Wilk normality test

data:  data$Angle.of.Attack
W = 0.91408, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(data$Chord.length)
```

```
        Shapiro-Wilk normality test

data:  data$Chord.length
W = 0.88483, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(data$Free.stream.velocity)
```

```
        Shapiro-Wilk normality test

data:  data$Free.stream.velocity
W = 0.81266, p-value < 2.2e-16
```

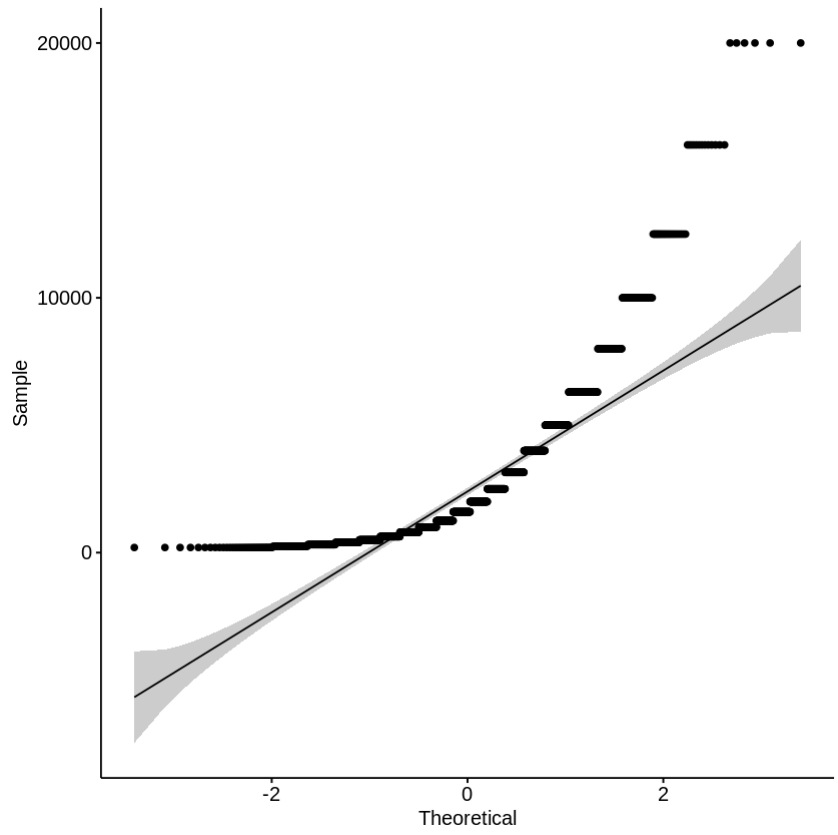
In [ ]:
```
shapiro.test(data$Suction.side.displacement.thickness)
```

```
        Shapiro-Wilk normality test

data:  data$Suction.side.displacement.thickness
W = 0.75184, p-value < 2.2e-16
```

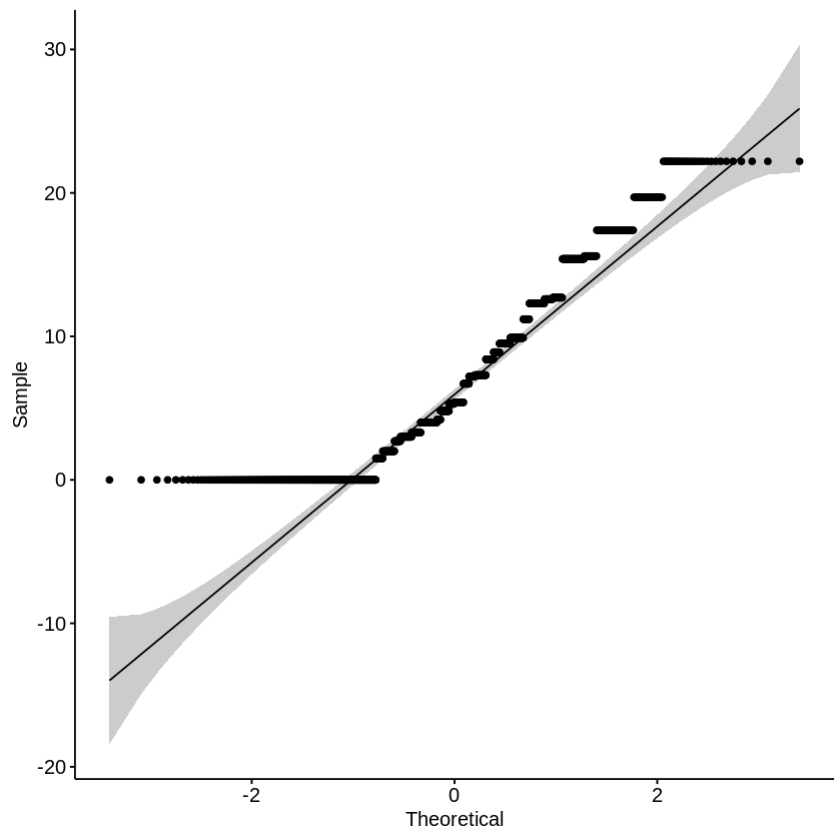
We find that none of the columns are Normal based on the Shapiro-Wilk Test. This can be said due to the p-value, which is lower than 0.05, which causes us to reject the null-hypothesis, which states that the data is not normally distributed.

We shall also make the Q-Q plots to check the normality.

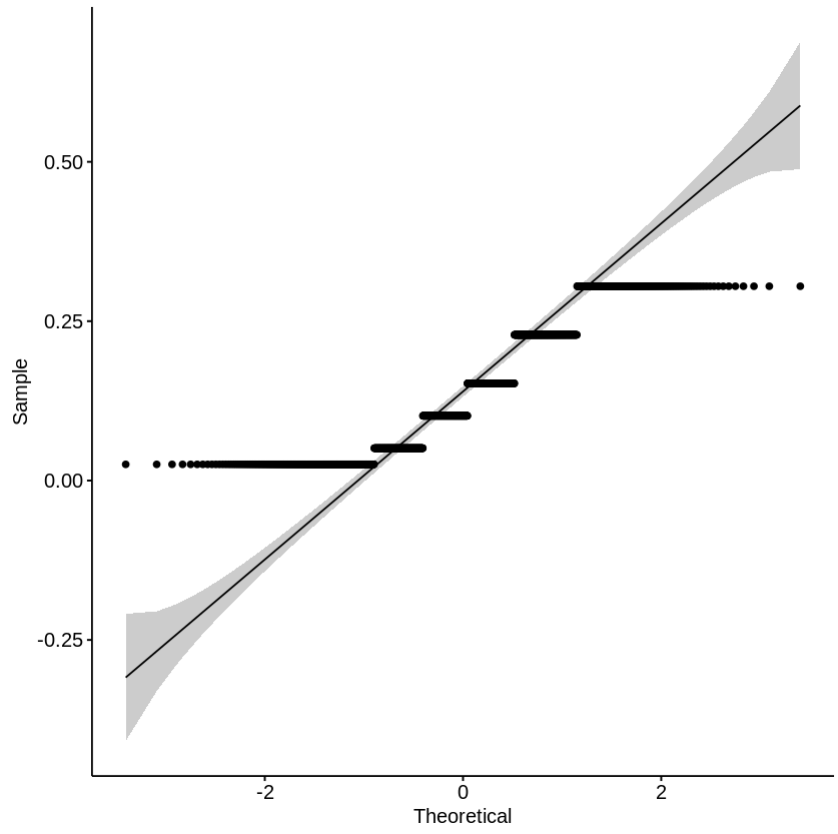
In [ ]:
```
ggqqplot(data$Frequency)
```



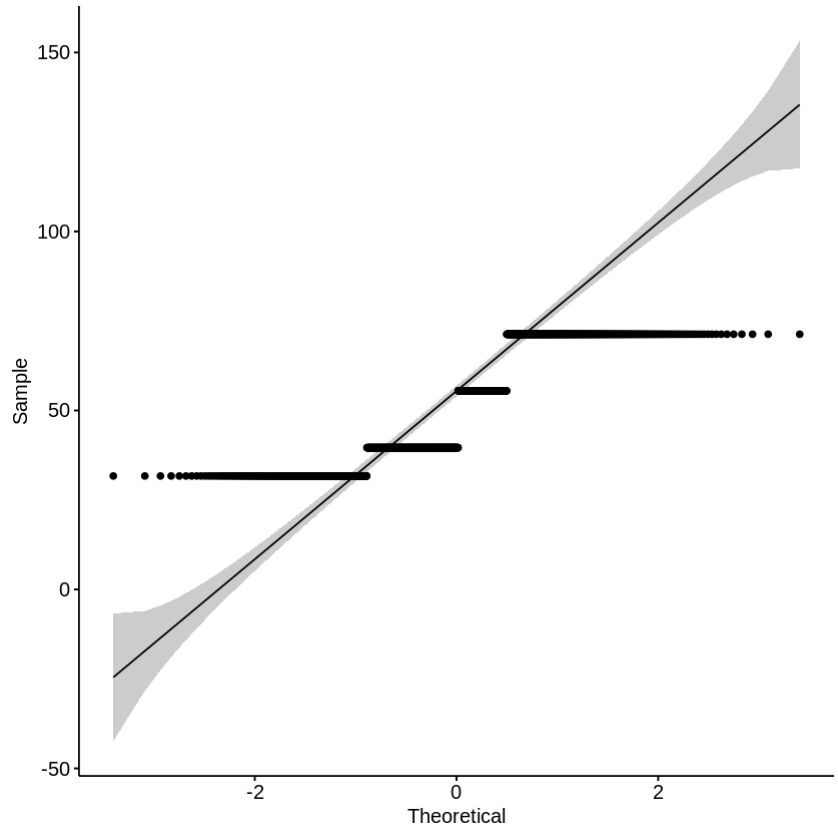
In [ ]:
```
ggqqplot(data$Angle.of.Attack)
```



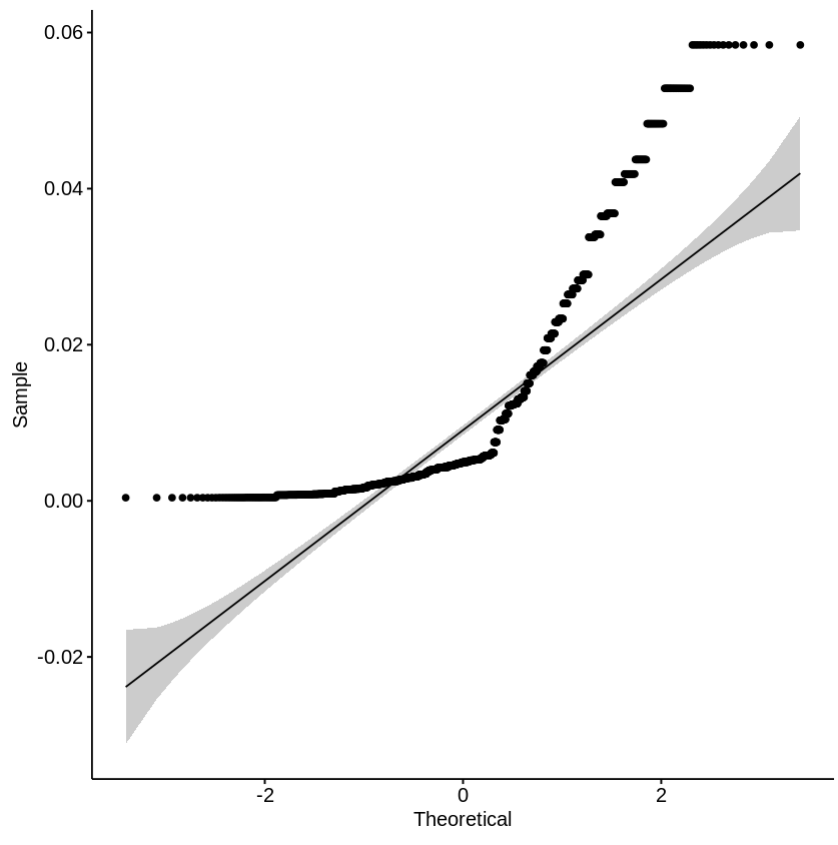
In [ ]:
```
ggqqplot(data$Chord.length)
```



In [ ]:
```
ggqqplot(data$Free.stream.velocity)
```



In [ ]:
```
ggqqplot(data$Suction.side.displacement.thickness)
```



A Q-Q plot assumes the second distribution to be normal and plots the quartiles for it, which lies on a line with slope approximately 45 degrees. If the data also lies on a similar line, then we can say it is normally distributed.

We shall now fit the Linear Regression model.

In [ ]:
```
model = lm(Scaled.sound.pressure.level ~ ., data = data)
```

```
model
```

```
Call:
lm(formula = Scaled.sound.pressure.level ~ ., data = data)

Coefficients:
                           (Intercept)                                Frequency
                             1.328e+02                               -1.282e-03
                       Angle.of.Attack                            Chord.length
                            -4.219e-01                               -3.569e+01
                  Free.stream.velocity  Suction.side.displacement.thickness
                             9.985e-02                               -1.473e+02
```

In [ ]:
```
summary(model)
```

```
Call:
lm(formula = Scaled.sound.pressure.level ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-17.480  -2.882  -0.209   3.152  16.064

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                       1.328e+02  5.447e-01  243.87   <2e-16 ***
Frequency                        -1.282e-03  4.211e-05  -30.45   <2e-16 ***
Angle.of.Attack                  -4.219e-01  3.890e-02  -10.85   <2e-16 ***
Chord.length                     -3.569e+01  1.630e+00  -21.89   <2e-16 ***
Free.stream.velocity              9.985e-02  8.132e-03   12.28   <2e-16 ***
Suction.side.displacement.thickness -1.473e+02  1.501e+01   -9.81   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.809 on 1497 degrees of freedom
Multiple R-squared:  0.5157,    Adjusted R-squared:  0.5141
F-statistic: 318.8 on 5 and 1497 DF,  p-value: < 2.2e-16
```

We see that the p-values for all variables is less than 0.05. Thus, we can reject the null hypothesis for each of the variables, which states that the coefficient for that variable is 0 and thus, there is no significant relationship between the variable and the target. Since we are rejecting the null hypothesis, we can say there is a significant relationship between the two.

We also get an adjusted R-squared value of 0.51, which is good enough for starters, but should be improved upon.

We also perform VIF (Variance Inflation Factor) on the model:

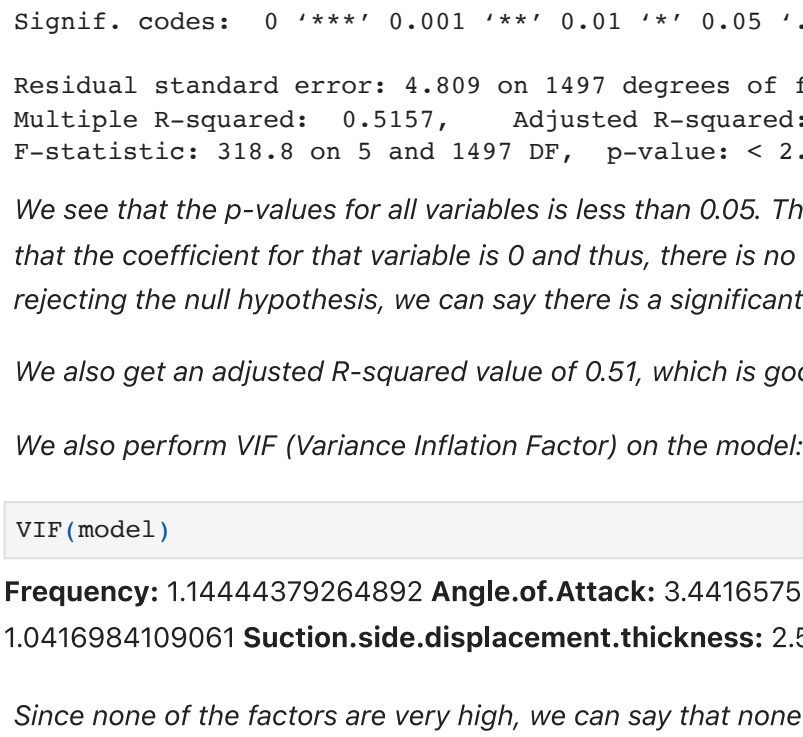
In [ ]:
```
VIF(model)
```

Frequency: 1.14444379264892 **Angle.of.Attack:** 3.44165751926982 **Chord.length:** 1.51075429995686 **Free.stream.velocity:** 1.0416984109061 **Suction.side.displacement.thickness:** 2.53212699073153

Since none of the factors are very high, we can say that none of the feature variables can be explained in terms of the other, and hence, each of these independent variables are uncorrelaed with each other.

We now check for Heteroschadasticity by plotting the residuals and the fitted values.

In [ ]:
```
plot(predict(model) , residuals(model))
```



We find that there is a kind-of cone structure that is being formed. Thus, we can not simply assume the data to be Homoschadistic.

We also check for the Autocorrelation using the Durbin-Watson test.

In [ ]:
```
dwtest(formula = model , alternative = "two.sided")
```

```
        Durbin-Watson test

data:  model
DW = 0.44743, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is not 0
```

Thus, our data does seem to be correlated.

We now apply the Box-Cox transformation. Note that it doesn't always work the way we wish to, as we shall see soon.

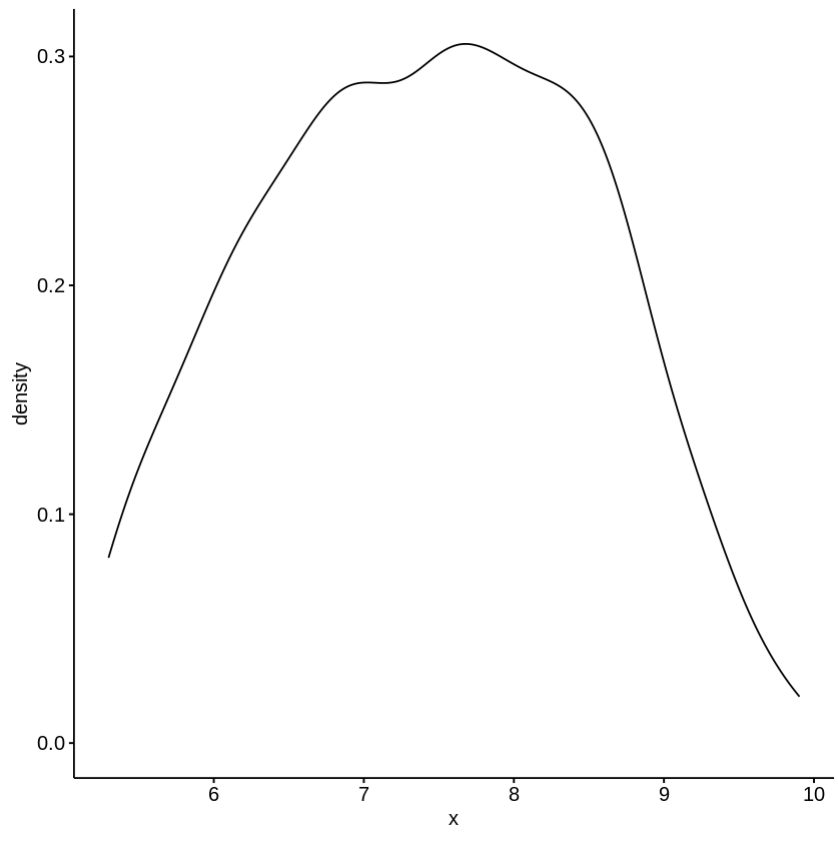
In [ ]:
```
# library(caret)

newSSPL = caret::BoxCoxTrans(data$Scaled.sound.pressure.level)
newF = caret::BoxCoxTrans(data$Frequency)
newAA = caret::BoxCoxTrans(data$Angle.of.Attack)
newFsv = caret::BoxCoxTrans(data$Free.stream.velocity)
newC = caret::BoxCoxTrans(data$Chord.length)
newSSdt = caret::BoxCoxTrans(data$Suction.side.displacement.thickness)

newdata = cbind(Frequency = predict(newF , data$Frequency), Angle.of.Attack = predict(newAA,data$Angle.of.Attack) , Fr
Suction.side.displacement.thickness = predict(newSSdt , data$Suction.side.displacement.thickness),
                Scaled.sound.pressure.level = predict(newSSPL , data$Scaled.sound.pressure.level))

newdata = data.frame(newdata)
```

```
Loading required package: lattice


Attaching package: ‘lattice’

The following object is masked from ‘package:regclass’:

    qq


Warning message in system("timedatectl", intern = TRUE):
“running command 'timedatectl' had status 1”
Attaching package: ‘caret’

The following object is masked from ‘package:VGAM’:

    predictors
```

Plotting the densities of the variables after Box-Cox transformation.

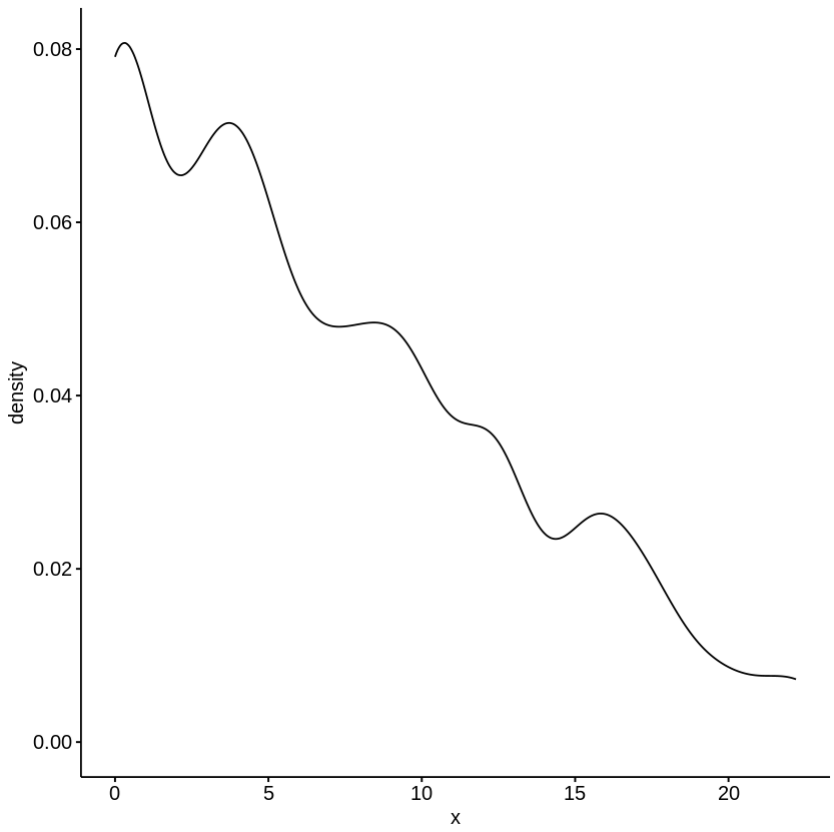
We shall now check the density graphs for each of the variables.

In [ ]:
```
ggdensity(newdata$Frequency)
```



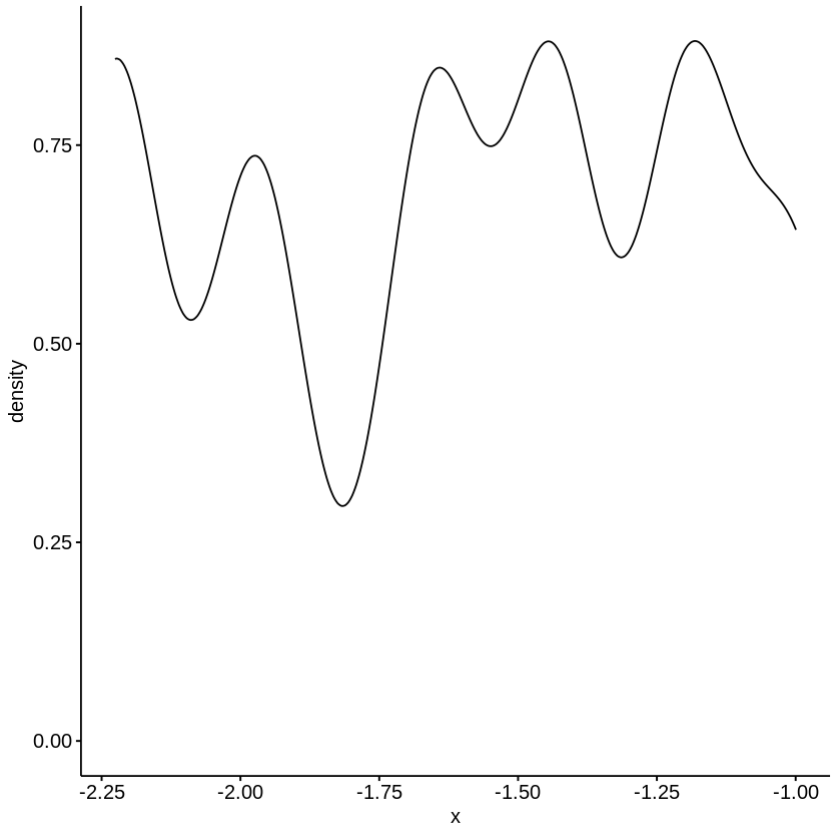
This was a right-tailed distribution. It does look Normal now.

In [ ]:
```
ggdensity(newdata$Angle.of.Attack)
```



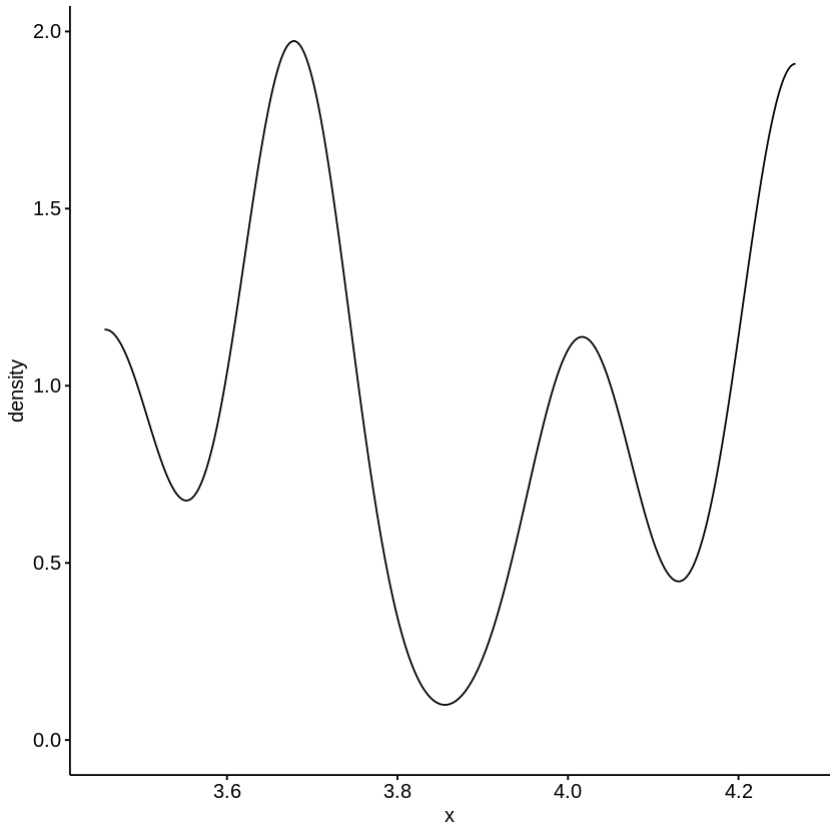
This remains the same.

In [ ]:
```
ggdensity(newdata$Chord.length)
```



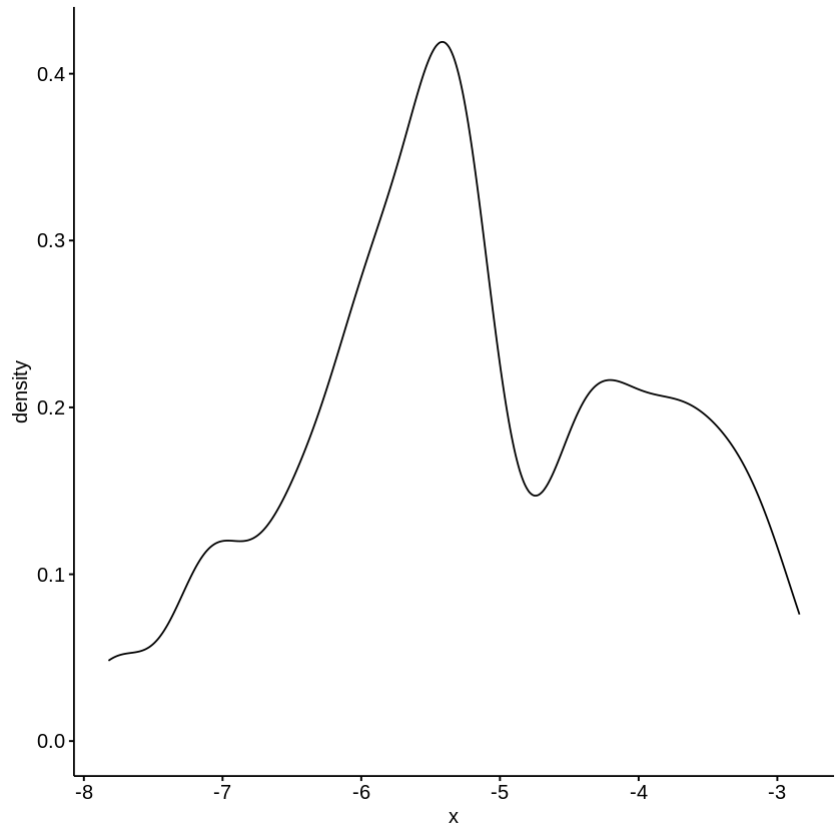
This, too, remains the same.

In [ ]:
```
ggdensity(newdata$Free.stream.velocity)
```



Again, the same.

In [ ]:
```
ggdensity(newdata$Suction.side.displacement.thickness)
```



This does look closer to Normal than before.

We shall again perform Shapiro-Wilk's test:

In [ ]:
```
shapiro.test(newdata$Frequency)
```

```
        Shapiro-Wilk normality test

data:  newdata$Frequency
W = 0.97939, p-value = 7.393e-14
```

Even though the p-value is lower than 0.05, it has still increased (previously it was of the order of -16).

In [ ]:
```
shapiro.test(newdata$Angle.of.Attack)
```

```
        Shapiro-Wilk normality test

data:  newdata$Angle.of.Attack
W = 0.91408, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(newdata$Chord.length)
```

```
        Shapiro-Wilk normality test

data:  newdata$Chord.length
W = 0.90206, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(newdata$Free.stream.velocity)
```

```
        Shapiro-Wilk normality test

data:  newdata$Free.stream.velocity
W = 0.83073, p-value < 2.2e-16
```

In [ ]:
```
shapiro.test(newdata$Suction.side.displacement.thickness)
```

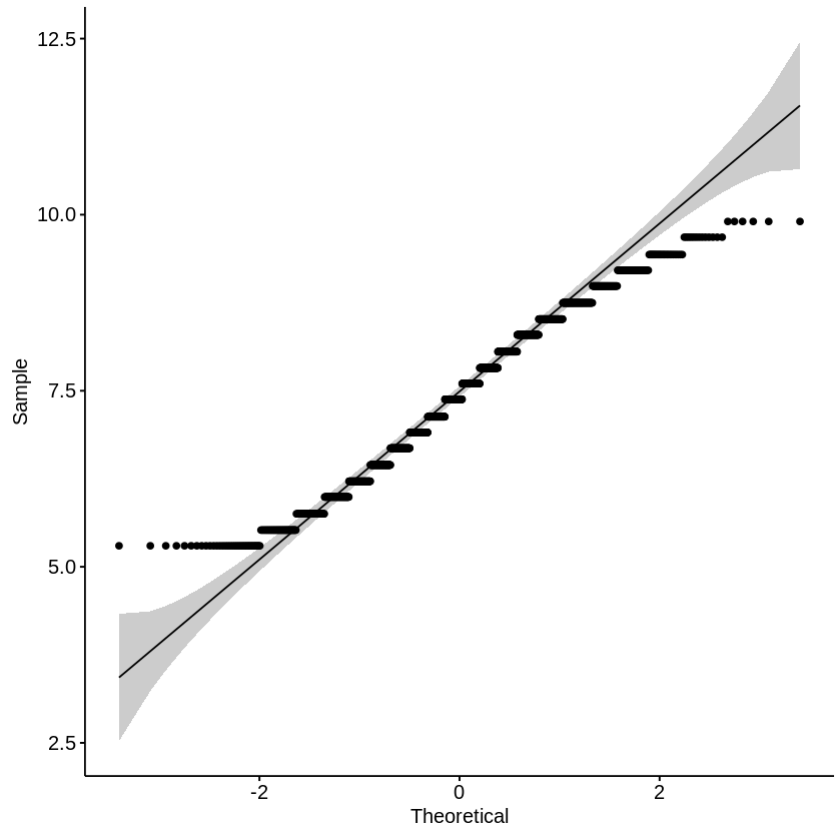
```
        Shapiro-Wilk normality test

data:  newdata$Suction.side.displacement.thickness
W = 0.97522, p-value = 2.025e-15
```

Again, here, even though the p-value is still lesser than 0.05, it has increased from the previous time.

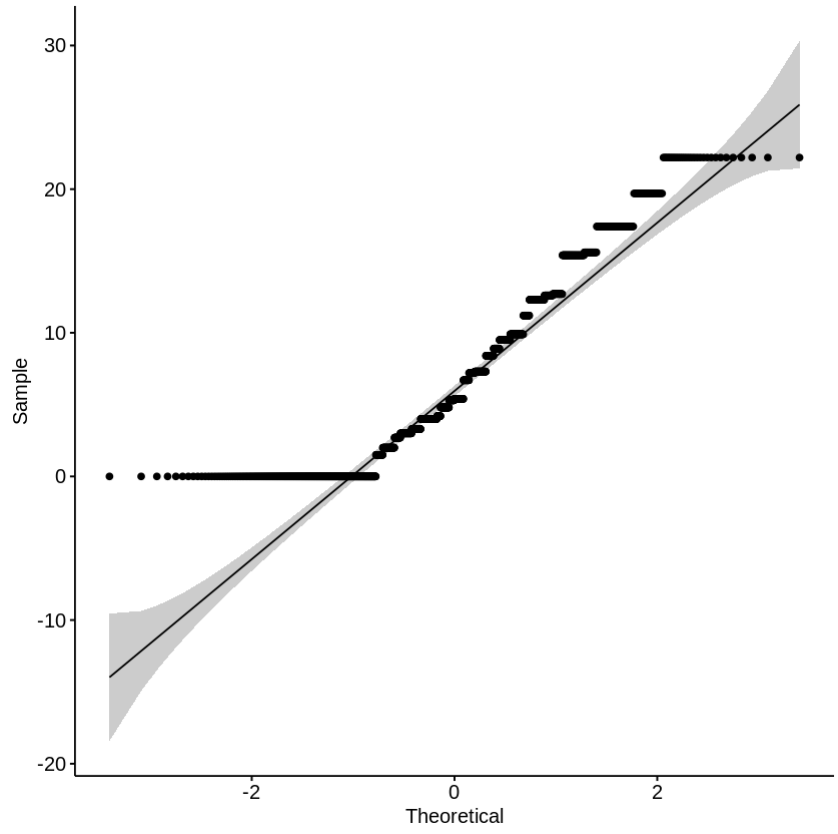
However, our variables are still not Normal as per the Shapiro-Wilk test.

Plotting the Q-Q curves:

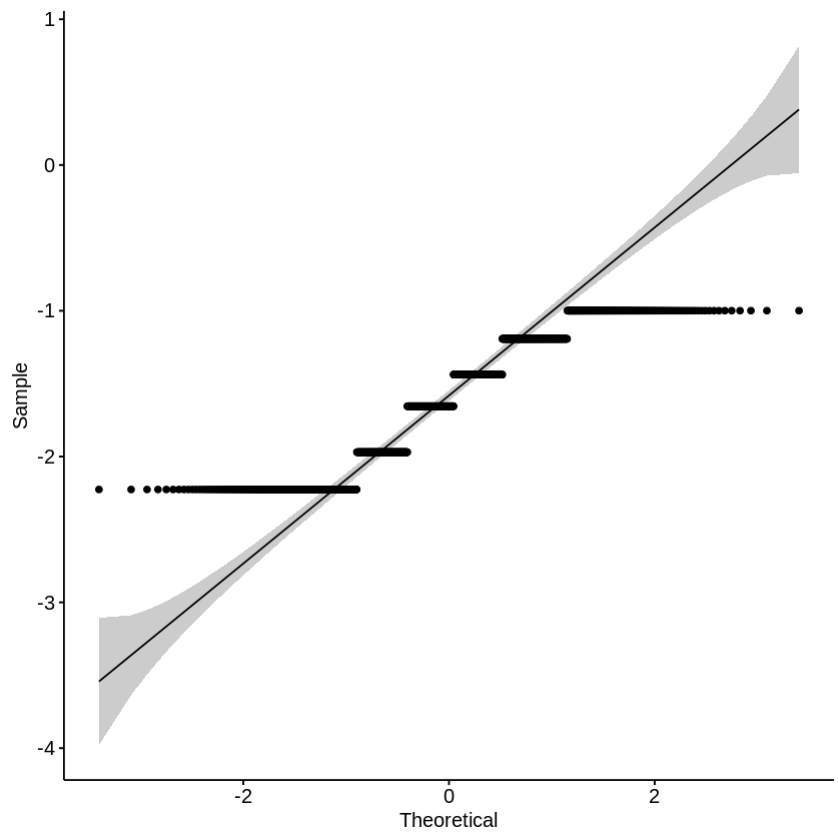
In [ ]:
```
ggqqplot(newdata$Frequency)
```



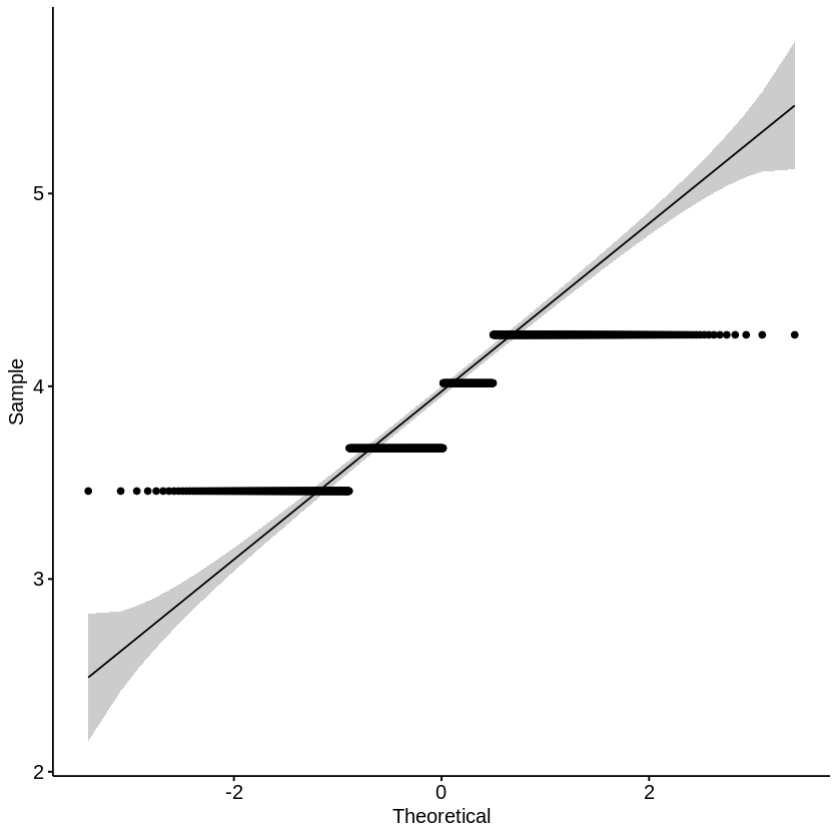
In [ ]:
```
ggqqplot(newdata$Angle.of.Attack)
```



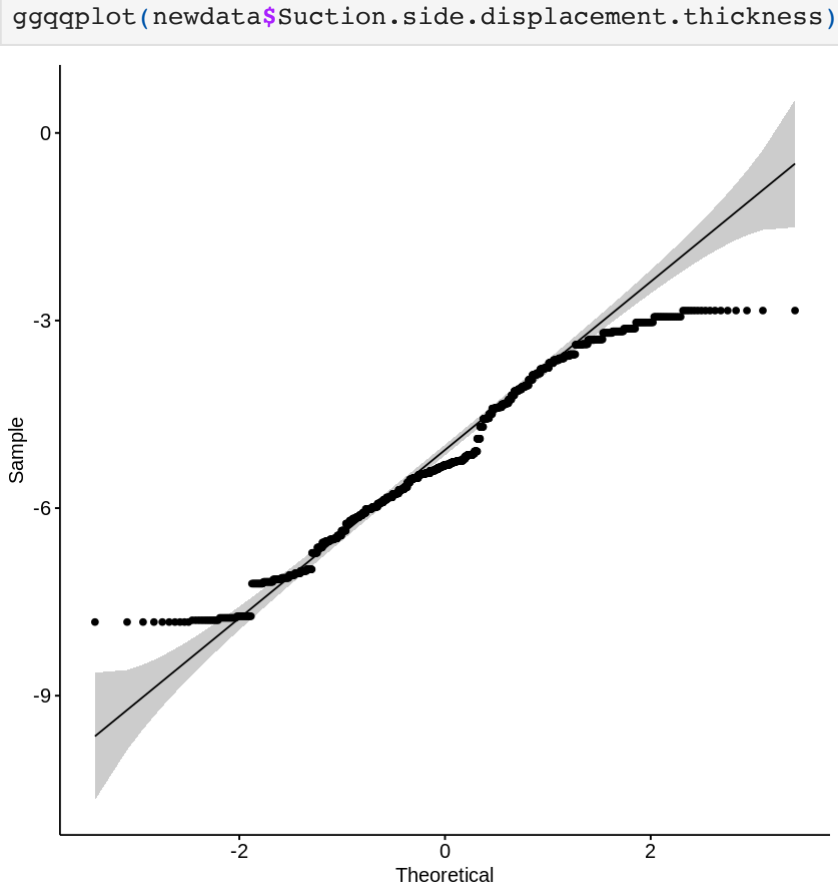
In [ ]:
```
ggqqplot(newdata$Chord.length)
```



In [ ]:
```
ggqqplot(newdata$Free.stream.velocity)
```



In [ ]:
```
ggqqplot(newdata$Suction.side.displacement.thickness)
```



Again, the first and last graphs are much closer to normality, however, we still cannot call them to be normally distributed.

We shall now run the Linear Regression Model on the new variables that were transformed using the Box-Cox Transformation.

In [ ]:
```
model2 <- lm(newdata$Scaled.sound.pressure.level ~ . , data = newdata)
model2
```

```
Call:
lm(formula = newdata$Scaled.sound.pressure.level ~ ., data = newdata)

Coefficients:
                         (Intercept)                            Frequency
                             6878.77                              -445.94
                     Angle.of.Attack                  Free.stream.velocity
                              -47.99                               583.49
                       Chord.length  Suction.side.displacement.thickness
                             -865.12                              -177.65
```

In [ ]:
```
summary(model2)
```

```
##
## Call:
## lm(formula = newdata$Scaled.sound.pressure.level ~ ., data = newdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2634.45  -364.81    31.63   432.16  2206.59
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        6878.77     274.76  25.036  < 2e-16 ***
## Frequency                          -445.94      16.16 -27.587  < 2e-16 ***
## Angle.of.Attack                     -47.99      11.45  -4.191 2.94e-05 ***
## Free.stream.velocity                583.49      54.12  10.781  < 2e-16 ***
## Chord.length                       -865.12      88.14  -9.815  < 2e-16 ***
## Suction.side.displacement.thickness -177.65      48.37  -3.673 0.000248 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 628.2 on 1497 degrees of freedom
## Multiple R-squared:  0.4578, Adjusted R-squared:  0.456
## F-statistic: 252.8 on 5 and 1497 DF,  p-value: < 2.2e-16
```

Again, we find that all the p-values for the variables is lesser than 0.05. Thus, we reject the null hypothesis that these variables do not contribute significantly to the target.

We get a adjusted R-squared value of 0.45, which is lower than last time.

```
dwtest(model2 , alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 0.34446, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is not 0
```

Finally, we shall try the Ridge Regression:

```
model_ridge <- linearRidge(Scaled.sound.pressure.level~. , data = data )
summary(model_ridge)
```

```
##
## Call:
## linearRidge(formula = Scaled.sound.pressure.level ~ ., data = data)
##
##
## Coefficients:
##                                     Estimate Scaled estimate
## (Intercept)                        1.306e+02              NA
## Frequency                         -9.692e-04      -1.184e+02
## Angle.of.Attack                   -2.390e-01      -5.482e+01
## Chord.length                      -2.468e+01      -8.949e+01
## Free.stream.velocity               7.229e-02       4.363e+01
## Suction.side.displacement.thickness -1.452e+02      -7.402e+01
##                                   Std. Error (scaled) t value (scaled)
## (Intercept)                                        NA               NA
## Frequency                                   4.215e+00            28.09
## Angle.of.Attack                             4.609e+00            11.89
## Chord.length                                4.298e+00            20.82
## Free.stream.velocity                        4.139e+00            10.54
## Suction.side.displacement.thickness         4.495e+00            16.47
##                                   Pr(>|t|)
## (Intercept)                             NA
## Frequency                           <2e-16 ***
## Angle.of.Attack                     <2e-16 ***
## Chord.length                        <2e-16 ***
## Free.stream.velocity                <2e-16 ***
## Suction.side.displacement.thickness <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Ridge parameter: 0.204241, chosen automatically, computed using 3 PCs
##
## Degrees of freedom: model 3.804 , variance 3.018 , residual 4.59
```

Once again, we reject the null hypothesis that the variables do not contribute significantly to the target variable. We achieve an adjusted R-squared value of 0.481 as calculated in the cell below.

```
y_pred <- predict(model_ridge , newdata = data[,1:5])
y<- data$Scaled.sound.pressure.level
sst <- sum((y - mean(y))^2)
sse <- sum((y_pred - y)^2)
R_square <- 1 - sse/sst * (1502 / 1496)
R_square
```

```
## [1] 0.4816331
```