Regression Analysis

Final Report, KU Leuven, Fall 2019

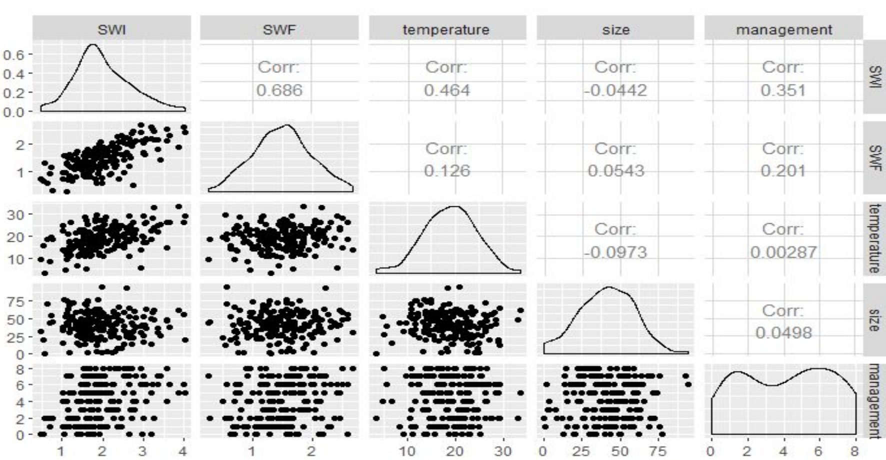
Kendall Brown r0773111

Introduction

The purpose of this report is to analyze the invertebrate data set provided on toledo during the fall semester of KU Leuven's 2019 Regression Analysis course. The data set contains six measurements, the Shannon-Wiener Index (SWI), Shannon-Weiner Floristic Index (SWF), the temperature at the time of sampling measured in celsius, the size of the sampling patch measured in square-meters, the number of years the sampling patch has been subject to management, and the duration of the sampling event. Four-hundred samples were gathered and will be subsetted into equally large training and validation sets for the purposes of model building and testing. As per assignment instructions, duration shall not be used as a possible indicator for SWI.

Descriptive Analysis

Seen below is a graphic produced with the training data to visualize the variables which may be influential in the model building process. As it can be seen in the density plots along the diagonal, the variables are normally distributed with the exception of management. The calculated correlation measurements along with the scatter-plots show SWI and SWF share a strong correlation with each other of .686. Temperature shares a strong relation with SWI as well with a correlation of .464. Management follows with a slight correlation of .351 with SWI and a small correlation of .201 with SWF. This slight relation between management and SWF may be the source of some unwanted multicollinearity. The measurement of plot size appears to be nothing but random noise sharing no significant relation between any of the other variables.



Basic Model

Before building a more complicated model, the training data shall be used to create and evaluate a base model. This model will not consider interaction or polynomial effects and will be drafted according to the following formula.

SWI~SWF+temperature+size+management

Initial results prove to be rather poor. As expected the size of the sample plot does not bear significance when used as a predictor of SWI having a regression coefficient of near 0. The SWF, temperature, and management do prove to be more valuable predictors with respective regression coefficients of .835, .049 and .063 and an intercept of -.357. This does not translate into strong predictive power as the model, although highly significant, only achieves a coefficient of determination value of .66. Moreover, diagnostics show a strong indication of heteroscedasticity indicating a transform is needed. There does not appear to be any indication of residual non-normality and independence is to be assumed as per assignment details. Potential multicollinearity was observed earlier. Fortunately it is by no means severe as formal tests found within the attached code provided little evidence to suggest multicollinearity. Multiple outliers are detected via the standardized residual plots. These samples have indices 69, 106, 161, 178, 185, and 200. By examining both the sample influence plot and Cook's distance bar chart it is quite clear that sample 106 is a highly significant outlier and is likely influencing the performance of the model. Fitting the test data to the model leads to a prediction root mean squared error of .448. Additionally, SWF apparently does not share a linear relation with SWI. This will be explored later.

Advanced Models

In order to address the issues found with the base model, several techniques are to be employed in hopes of adding power to the model. To begin the variable size shall be removed from the training data as it is clearly unimportant. Next a box-cox transformation shall be employed to address previously observed heteroscedasticity. The transformation process will consider all possible main effect interactions when calculating lambda. The results of the transform gives a lambda value of .4545 and applying this transform to the model corrects the problem of heteroscedasticity observed earlier.

The addition of interaction effects has injected a fair bit of unwanted variables into the model. To address this problem model trimming methods were employed and resulted in the following interaction effects model (the forward selection, backward elimination, and both-ways selection methods resulted in the same model).

SWI^lambda ~ SWF + temperature + management + SWF:management

Model diagnostics show heteroscedasticity has been eliminated, normality, maintained, independence is still assumed to be true, and influential outliers still exist within the training data. The coefficient of determination does not see a significant increase only rising to .67. Regression coefficients for this model are as follows. Intercept: .504, SWF: .311, Temperature:

.015, Management: .04, and SWF*Management: -.013. This model lacks substantial improvement over the base model discussed earlier and requires further molding.

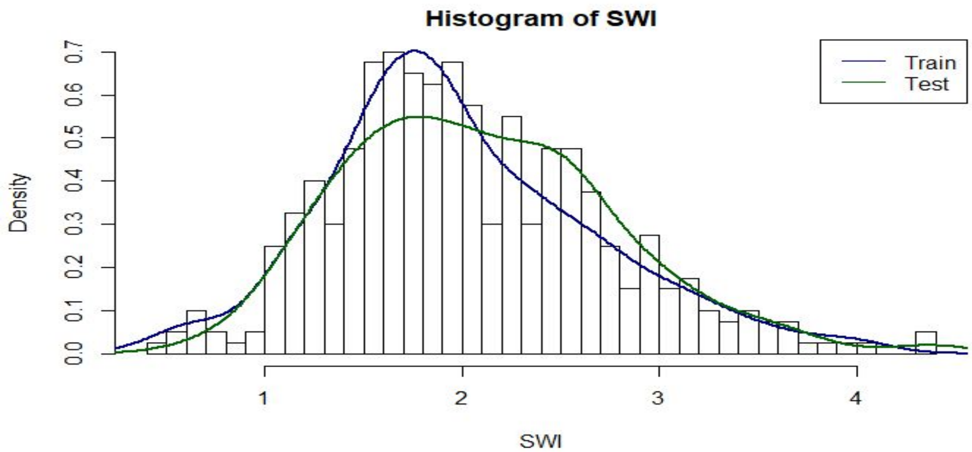
Based on the diagnostic plots from the base model the nonlinear relation between SWI and SWF shall now be explored. To accomplish this task a new polynomial model was constructed taking the square of SWF as a possible predictor of SWI. Interaction significance was also be explored during the model building process. The results of forward selection and backward elimination model building and comparison resulted in the following model.

SWI^.7 ~ SWF^2 + Temperature + Management

Of note, the only interaction effect previously held to be significant has been dropped from the model leaving only the square of SWF and the other two main effects. Evaluation of model diagnostics show that heteroscedasticity of the errors is present within this model. Implementation of a box-cox transform with lambda=.7 solved this issue. Regression coefficients associated with this model are as follows. Intercept: .546, SWF^2: .156, Temperature: .027, Management: .037. The adjusted coefficient of determination has a value of .687 a slight improvement over the base model. The prediction root mean square error sees a quite substantial improvement dropping down to .242 from .448 as is expected with a rise in the adjusted coefficient of determination. This model shall be taken to be the final model for the purpose of this report.

At the time of building this model an alternative model was created as well. As an alternative model, an iteratively reweighted least squares model without the box cox transformation and the same inputs as the final model was considered. After a single iteration there was no real improvement to the model. As expected the normality of the errors is more pronounced than it was in the box-cox model. Regression coefficients are Intercept: .142, SWF^2: .289, temperature: .046, management: .061. These weights did improve the model's adjusted coefficient of determination to .73, however this did not lead to stronger prediction performance. Calculating the prediction root mean squared error of the predicted values versus the measured results measures to .434. One of the advantages of taking the WLS regression model is that the model handles influential outliers much more appropriately. In this scenario it does not provide meaningful improvement, but the WLS regression should still be considered a fine alternative approach to an OLS model.

To gain insight into what can be done to improve the model, the models were trained on the test data and had their performance evaluated. As expected with such high RMSEs, the models do not generalize well to new data. This is interesting as both models are quite small in size and should be resistant to overfitting. Comparing a basic first order model trained on the test data to that trained on the training data reveals that the model trained on the test data appears to deviate quite harshly from that trained on the training data After running two-sided t-tests of each variable within the data sets it can be claimed that the variable SWI has noticeable variation of distribution between the two data sets. This is further seen in the following plot of the density histogram of SWI overlayed with the densities of both the training and test data.



Provided here is a table of the 95% confidence intervals for each regression coefficient of all four models. All models presented here yielded an F-test p-value of ~zero for the model and each regression coefficient. By examining these coefficients it can be claimed that the intercept results in the most drastic change in model performance as all other variables appear to be quite uniform in their intervals across all models.

Included in the table as well are the coefficients of a model trained on the entire data set with the model taking a box-cox transform(lambda=.4545) as that appeared to be quite impactful during the training session.

Model\Coef	Intercept	SWF^2	Temp	MGMT
Train Final	0.423-0.669	0.135-0.177	0.021-0.033	0.025-0.049
Test Final	0.337-0.654	0.148-0.193	0.024-0.038	0.023-0.049
Train Alt	-0.031-0.315	0.254-0.325	0.038-0.055	0.041-0.081
Test Alt	-0.336-0.222	0.268-0.352	0.047-0.070	0.047-0.094
Ult Model	0.706-0.815	0.08-0.097	0.013-0.018	0.015-0.025

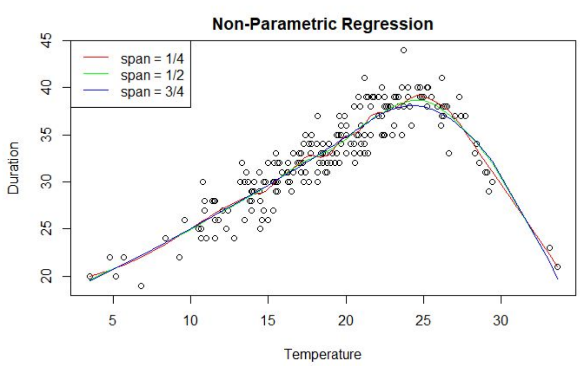
The ultimate model unsurprisingly has approximately the average performance of the training and test models. The adjusted coefficient of determination is calculated to be .64 with the regression coefficients being interpreted as such. WIth a mean intercept value of .76, the value of SWI^.4545 is expected to increase by .09, .02, and .02 points for each increase in SWF^2, temperature, and management respectively.

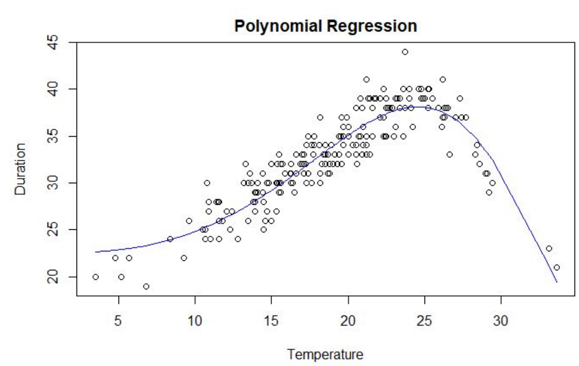
Non-Parametric Model

To further explore the data set a non-parametric model shall be created to determine if there is a relationship between duration and temperature. Examination of the scatter plot shows a potential inverse-parabolic relation indicating that when the outside temperature is nice(temperature ~between 20-27 degrees) the probability of conducting longer surveys increases. To test this observation, a non-parametric model and a quadratic regression model will be compared against each other to determine which models proves most powerful. Calculating several non-parametric regression lines reveals that a second order polynomial with a span of .25 proves to be a sufficient estimator of the relation. With independence assumed, this model passes the gaussian-markov assumptions of normality, linearity, and homoscedasticity.

When calculating a quadratic model it was discovered that a fourth degree polynomial estimates the function well. As it fails to pass the Gaussian-Markov assumption of normality a log transform was taken. The model equation shown here yields a coefficient of determination of .845, passing all Gaussian-Markov assumptions. Plot of fit and equation shown here.

ln(Duration) ~ Temperature^3 + Temperature^4





Based off of visual interpretation, it can be said that the non-parametric model performs better than the polynomial regression model, especially on the tails. To formally test which model does indeed perform better, the model RMSE of the fitted values against the measured were compared. The results of this test showed the non-parametric model outperforming the polynomial model by quite a large margin (non-parametric RMSE:1.78, polynomial RMSE:6.44). This was evident from the visualizations shown above as such it can be concluded that a non-parametric model does map this data set quite well.

Regression Analysis Final Project Code Apendix

Kendall Brown

2019-2020

Loading data, neccesary libraries, and partitioning data into training/test sets

```
rm(list=ls())
library(ggplot2)
library(MASS)
library(car)
library(lmtest)
library(olsrr)
library(mctest)
library(ppcor)
library(GGally)
library(dplyr)
library(Metrics)
library(ggeffects)
```

Code for tasks 1-5

Creating training and test data sets

```
data=read.table("C:/Users/kebro/OneDrive/KU Leuven/Regression/invertebrate.txt",header=T)
rnum=0773111
set.seed(rnum)
d.test <- sample(1:dim(data)[1], 200)
data.test <- data[d.test, ]
data.training <- data[-d.test, ]
n <- dim(data.training)[1]
p <- dim(data.training)[2]
```

Descriptive Analysis of training data

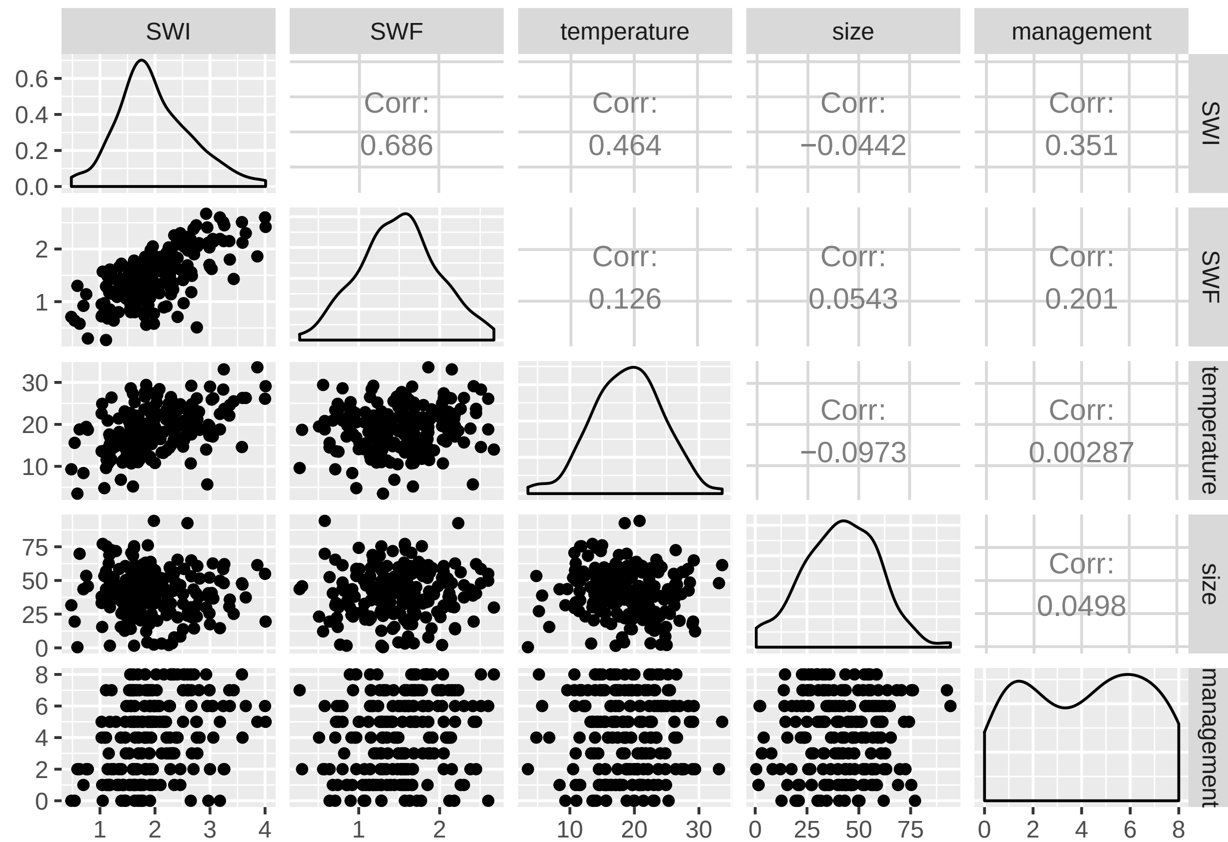
```
summary(data.training)
```

```
##       SWI              SWF          temperature         size
##  Min.   :0.480   Min.   :0.270   Min.   : 3.50   Min.   : 0.50
##  1st Qu.:1.567   1st Qu.:1.177   1st Qu.:15.38   1st Qu.:29.23
##  Median :1.895   Median :1.525   Median :19.05   Median :42.10
##  Mean   :1.994   Mean   :1.498   Mean   :18.95   Mean   :41.32
##  3rd Qu.:2.357   3rd Qu.:1.802   3rd Qu.:22.52   3rd Qu.:54.70
##  Max.   :4.010   Max.   :2.670   Max.   :33.60   Max.   :94.20
##    management       duration
##  Min.   :0.000   Min.   :19.00
##  1st Qu.:2.000   1st Qu.:30.00
##  Median :4.000   Median :33.00
##  Mean   :4.115   Mean   :32.75
##  3rd Qu.:6.000   3rd Qu.:37.00
##  Max.   :8.000   Max.   :44.00
```

```r
str(data.training)
```

```
## 'data.frame':    200 obs. of  6 variables:
##  $ SWI        : num  0.59 1.08 1.6 2.95 1.38 0.7 0.48 1.11 1.14 1.57 ...
##  $ SWF        : num  1.3 0.97 1.67 2.41 1.44 0.92 0.71 0.27 1.48 1.65 ...
##  $ temperature: num  3.5 4.8 5.2 5.7 6.8 8.4 9.3 9.6 10.5 10.7 ...
##  $ size       : num  0.5 53.3 27.2 38.8 15.5 43.5 31.6 43.6 52 70.5 ...
##  $ management : int  2 4 8 6 4 1 0 7 2 7 ...
##  $ duration   : int  20 22 20 22 19 24 22 26 25 25 ...
```

```r
ggpairs(data.training[,-ncol(data.training)])
```



From the ggpairs plot all data seems normally distributed with the exception of management, generally weak corelations between variables

Fitting a first order linear model predicting SWI as a function of all variables -duration

```r
fit.basic=lm(SWI~.-duration,data=data.training)
fit.basic.sum=summary(fit.basic)
fit.basic.sum
```

```
##
## Call:
## lm(formula = SWI ~ . - duration, data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02005 -0.27531 -0.02608  0.24343  1.53643
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.357059   0.147639  -2.418   0.0165 *
## SWF          0.834942   0.059745  13.975  < 2e-16 ***
## temperature  0.048518   0.005275   9.198  < 2e-16 ***
## size        -0.001927   0.001576  -1.222   0.2231
## management   0.063350   0.011409   5.553 9.13e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3961 on 195 degrees of freedom
## Multiple R-squared:  0.6694, Adjusted R-squared:  0.6626
## F-statistic:  98.7 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
#model coefficients and their 95% confidence intervals
coefficients(fit.basic)
```

```
##  (Intercept)          SWF  temperature         size   management
## -0.357058626  0.834941694  0.048517959 -0.001926504  0.063350391
```

```r
confint(fit.basic)
```

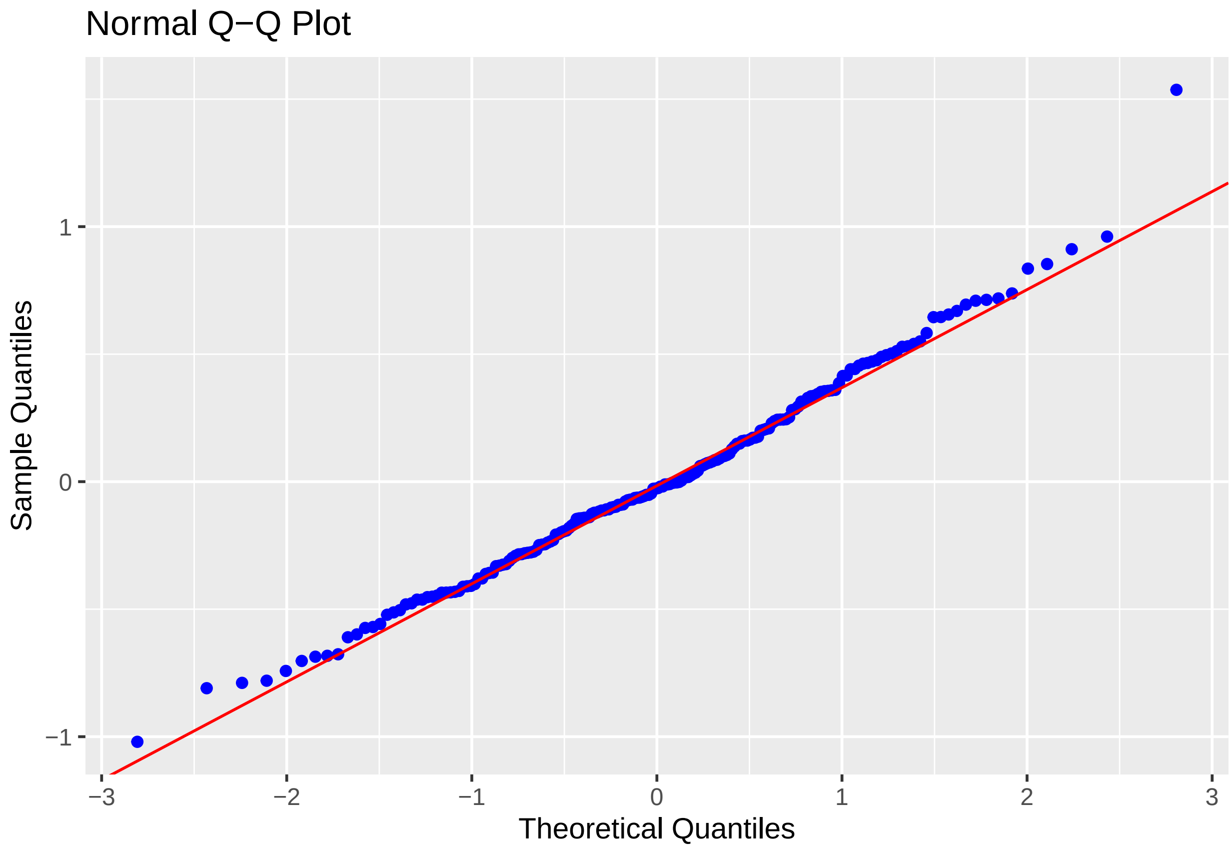
```
##                    2.5 %       97.5 %
## (Intercept) -0.648232700 -0.065884552
## SWF          0.717111387  0.952772002
## temperature  0.038114533  0.058921385
## size        -0.005035407  0.001182398
## management   0.040849086  0.085851697
```

```r
#ANOVA analysis of model
anova(fit.basic)
```

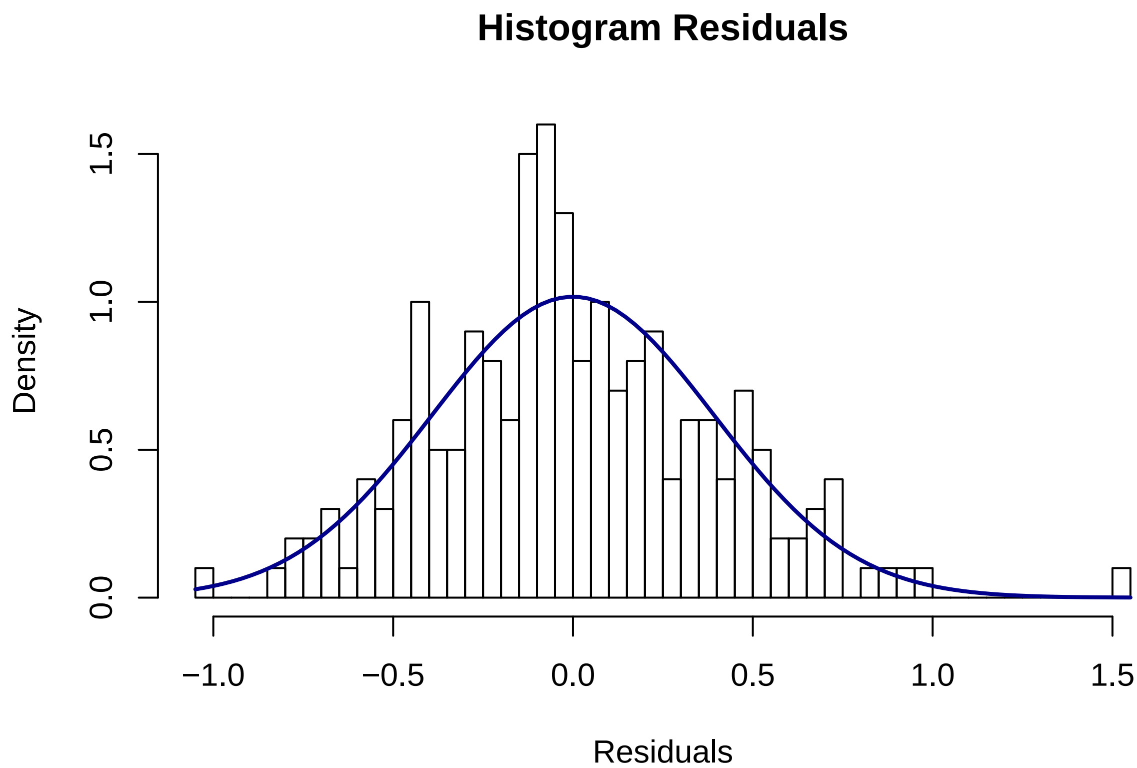
```
## Analysis of Variance Table
##
## Response: SWI
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## SWF           1 43.515  43.515 277.3264 < 2.2e-16 ***
## temperature   1 13.431  13.431  85.5981 < 2.2e-16 ***
## size          1  0.161   0.161   1.0291    0.3116
## management    1  4.838   4.838  30.8309 9.126e-08 ***
## Residuals   195 30.597   0.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing model assumptions

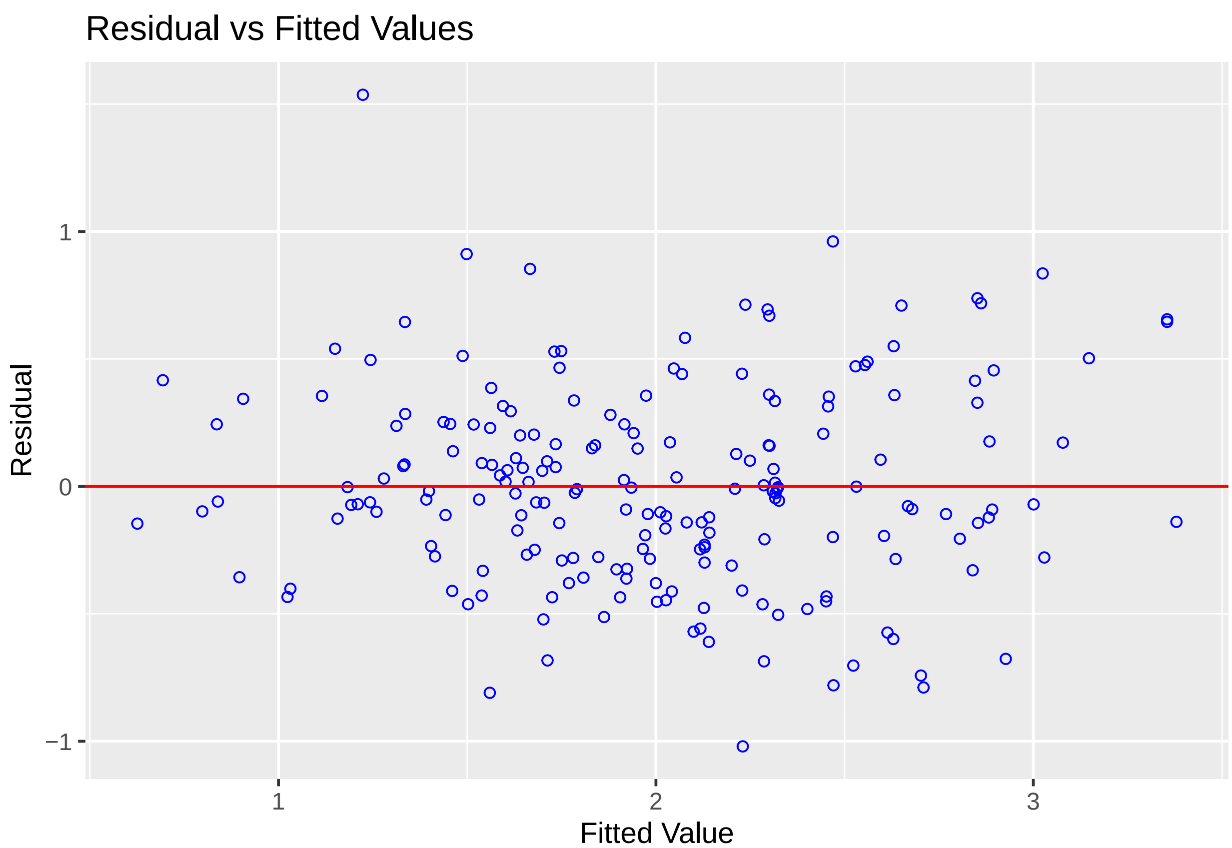
```r
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.basic)
```




```
hist(fit.basic$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.basic$residuals), sd=sd(fit.basic$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.basic)
```



First order model shows possible non-constant variance, normality appears to hold true, linearity is met.

Formal Tests of assumptions

```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.basic)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.496839, Df = 1, p = 0.061486
```

```
#Normality of errors tests
ols_test_normality(fit.basic)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9897         0.1589
## Kolmogorov-Smirnov        0.0513         0.6690
## Cramer-von Mises          29.964         0.0000
## Anderson-Darling          0.4314         0.3030
## -----------------------------------------------
```

```
ols_test_correlation(fit.basic)
```

```
## [1] 0.9939204
```

Tests for multi-colinearity

```r
#Covariance and correlation matracies of each predictor variable
cov(data.training[,c(-1,-6)])
```

```
##                   SWF temperature        size management
## SWF         0.2347857  0.32936558   0.4723405 0.24435930
## temperature 0.3293656 29.12542688  -9.4289621 0.03900251
## size        0.4723405 -9.42896206 322.2585023 2.24644975
## management  0.2443593  0.03900251   2.2464497 6.32339196
```

```r
cor(data.training[,c(-1,-6)])
```

```
##                    SWF  temperature        size  management
## SWF         1.00000000  0.125952345  0.05430217 0.200547852
## temperature 0.12595235  1.000000000 -0.09732525 0.002873964
## size        0.05430217 -0.097325254  1.00000000 0.049764479
## management  0.20054785  0.002873964  0.04976448 1.000000000
```

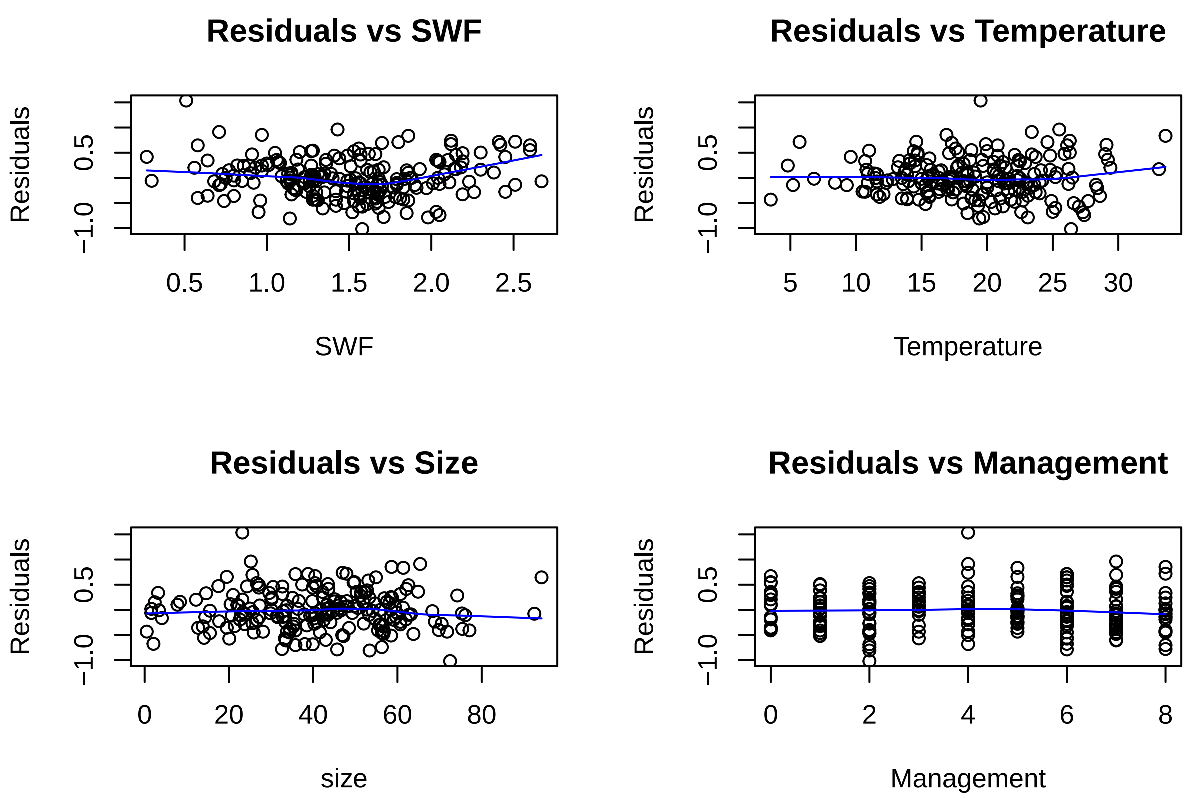
```r
#formal multicolinearity tests
omcdiag(data.training[,c(-1,-6)],data.training[,1])
```

```
##
## Call:
## omcdiag(x = data.training[, c(-1, -6)], y = data.training[, 1])
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.9295         0
## Farrar Chi-Square:        14.3796         1
## Red Indicator:             0.1088         0
## Sum of Lambda Inverse:     4.1503         0
## Theil's Method:           -1.8644         0
## Condition Number:         13.0682         0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

```r
#no severe multicolinearity discovered
```

```r
par(mfrow=c(2,2))
#Checking linearity of SWF
plot(data.training$SWF,fit.basic$residuals,
     main="Residuals vs SWF",xlab="SWF",ylab="Residuals")
lines(lowess(fit.basic$residuals~data.training$SWF),col="blue")

#Checking linearity of Temperature
plot(data.training$temperature,fit.basic$residuals,
     main="Residuals vs Temperature",xlab="Temperature",ylab="Residuals")
lines(lowess(fit.basic$residuals~data.training$temperature),col="blue")

#Checking linearity of Size
plot(data.training$size,fit.basic$residuals,
     main="Residuals vs Size",xlab="size",ylab="Residuals")
lines(lowess(fit.basic$residuals~data.training$size),col="blue")
```

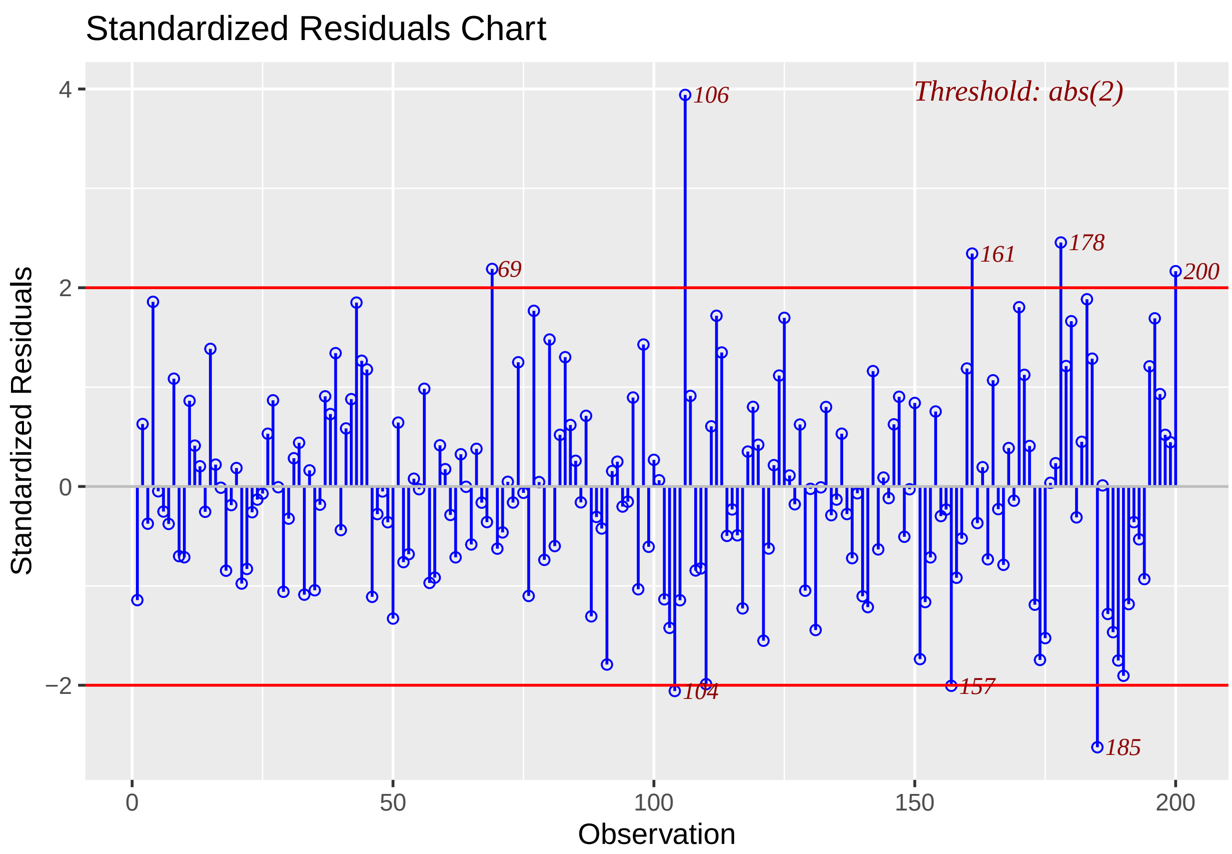
```r
#Checking linearity of Management
plot(data.training$management,fit.basic$residuals,
     main="Residuals vs Management",xlab="Management",ylab="Residuals")

lines(lowess(fit.basic$residuals~data.training$management),col="blue")
```



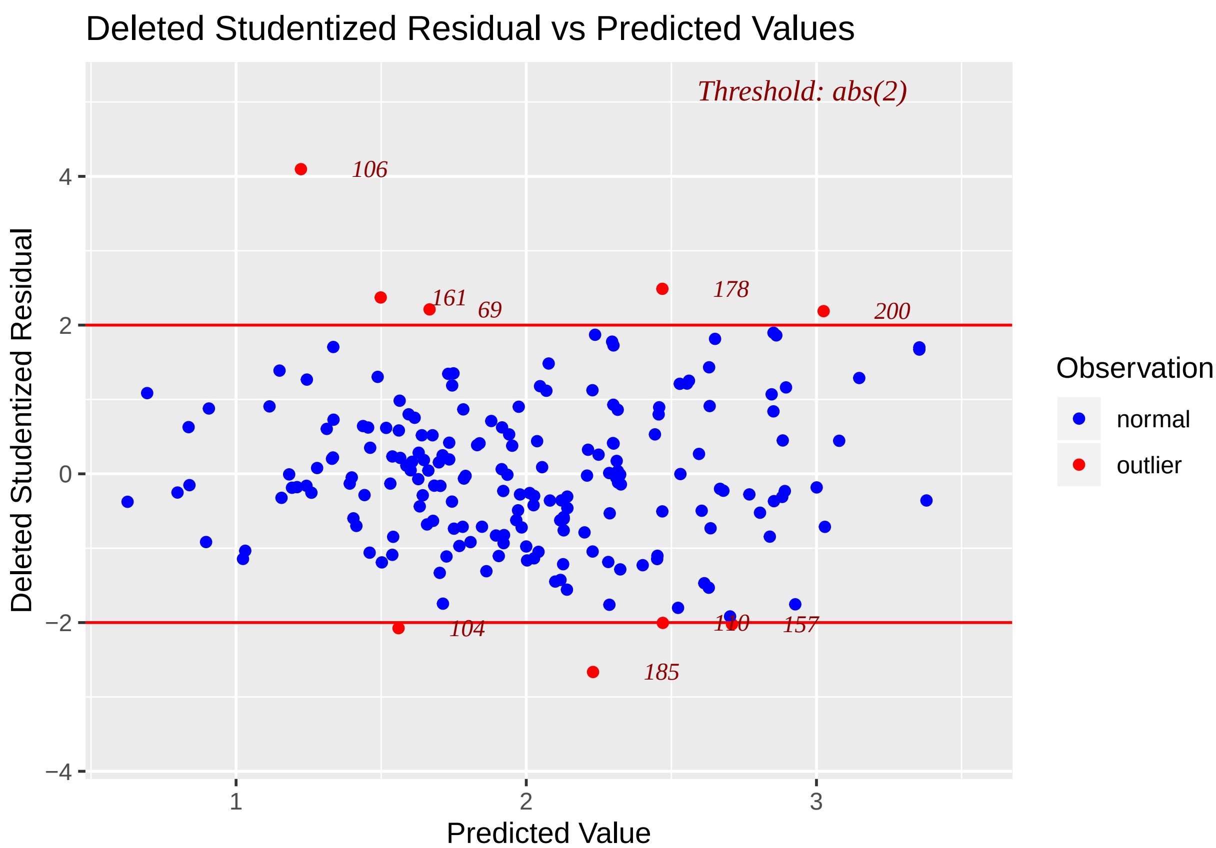
SWF appears to be nonlinear. Temperature and size may be non-linear as well. Managment is seemingly linear.

Checking for outliers and heavily influential observations

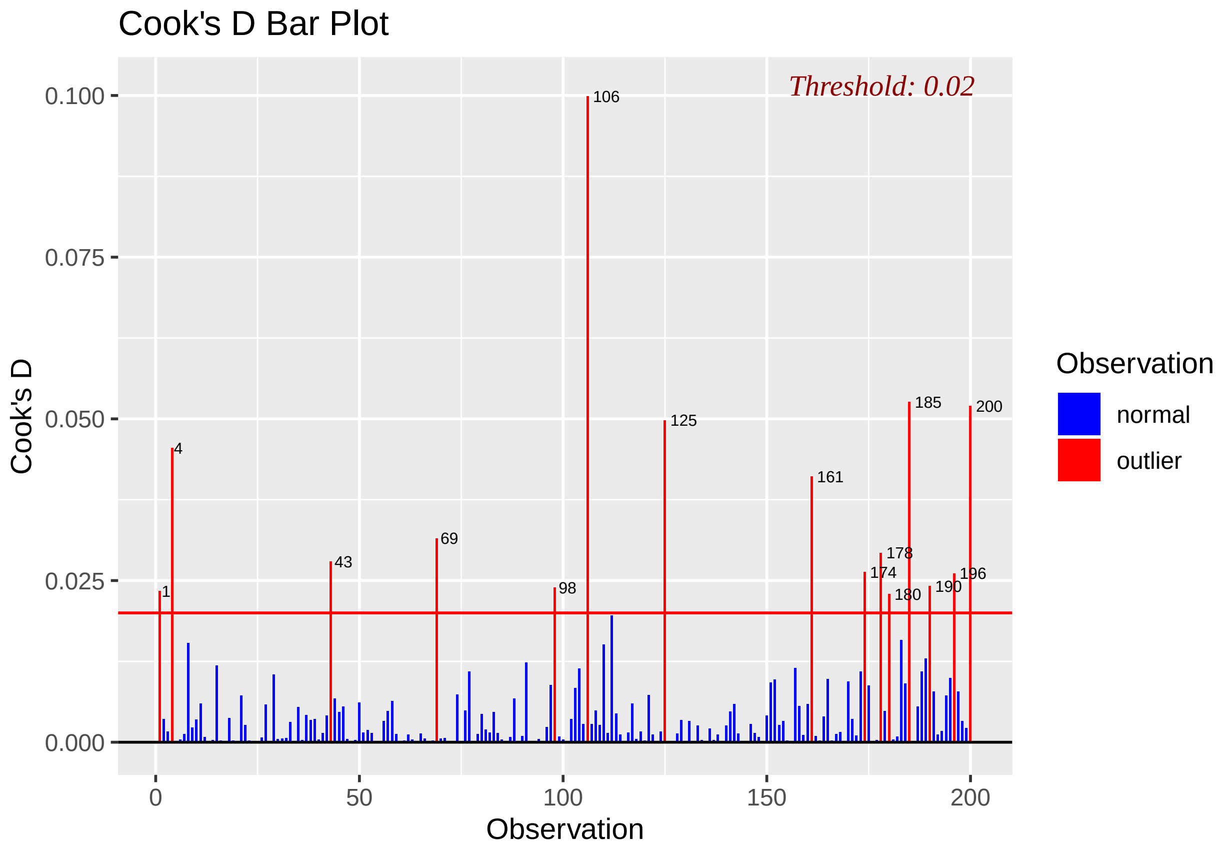
```r
mod=fit.basic
#Standardized Residual plot
ols_plot_resid_stand(mod)
```



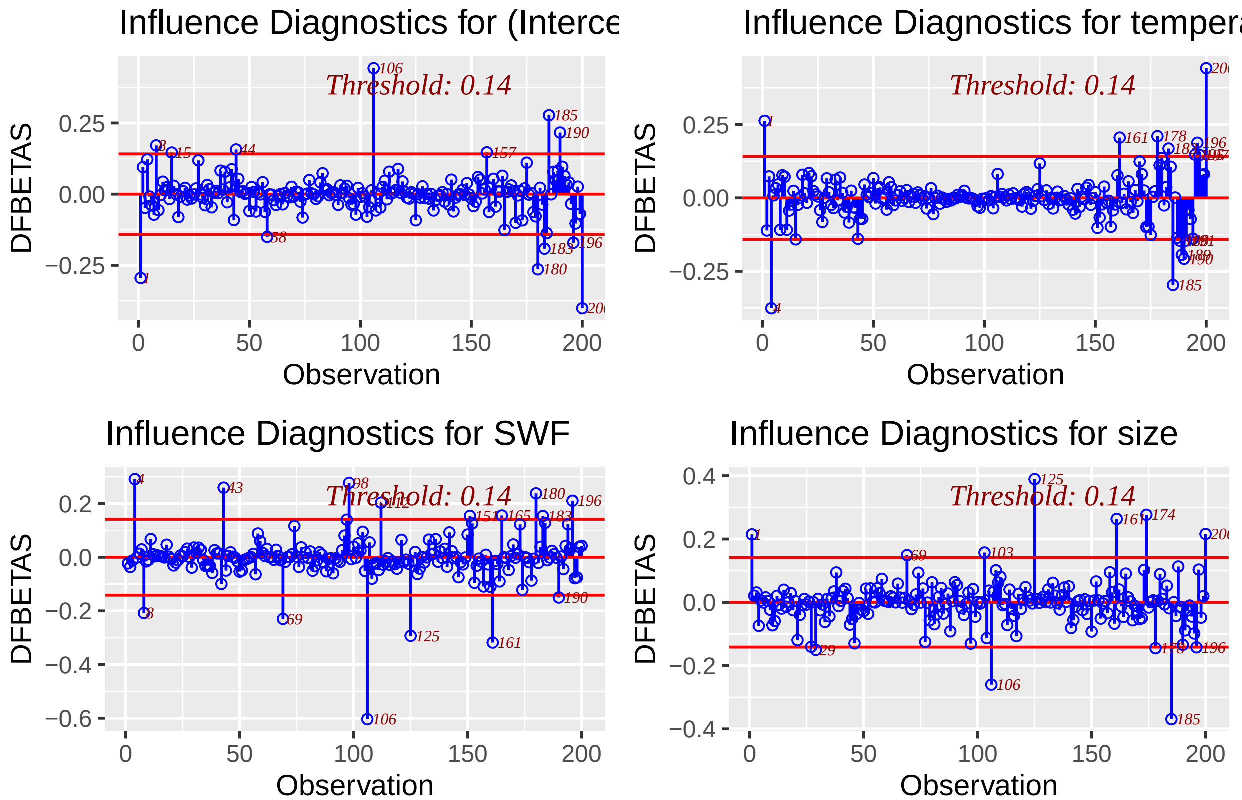

```r
#studentized residual plot
ols_plot_resid_stud_fit(mod)
```

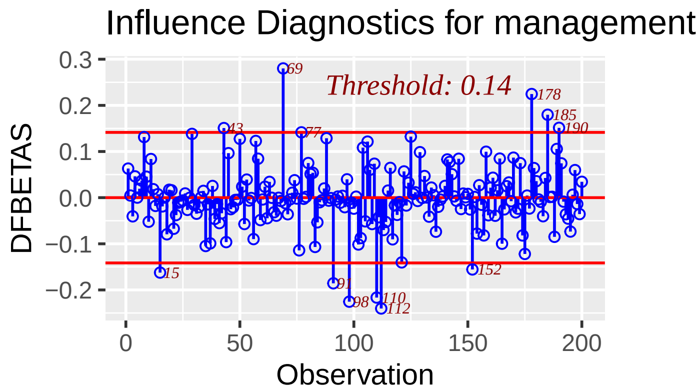



```r
#cooks distance plot
ols_plot_cooksd_bar(mod)
```

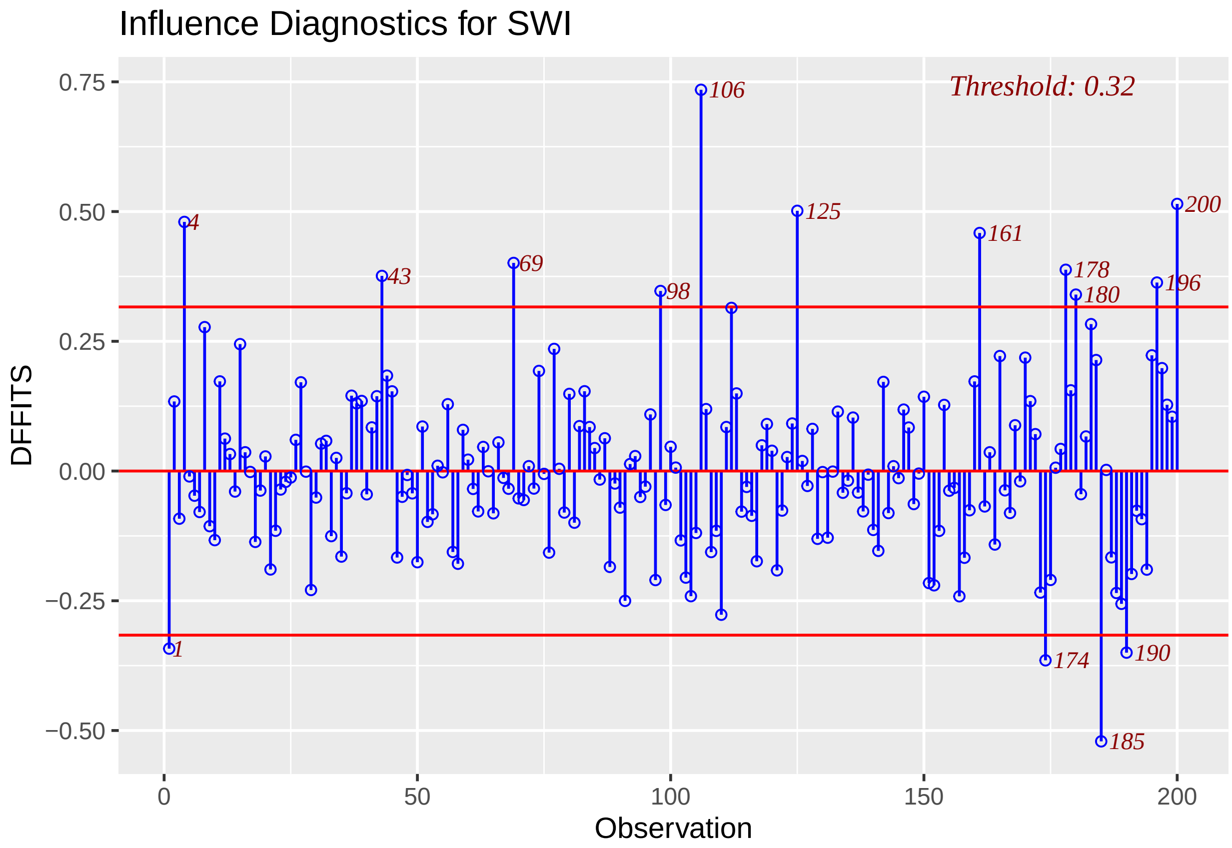



```r
#DFBetas for each variable
ols_plot_dfbetas(mod)
```

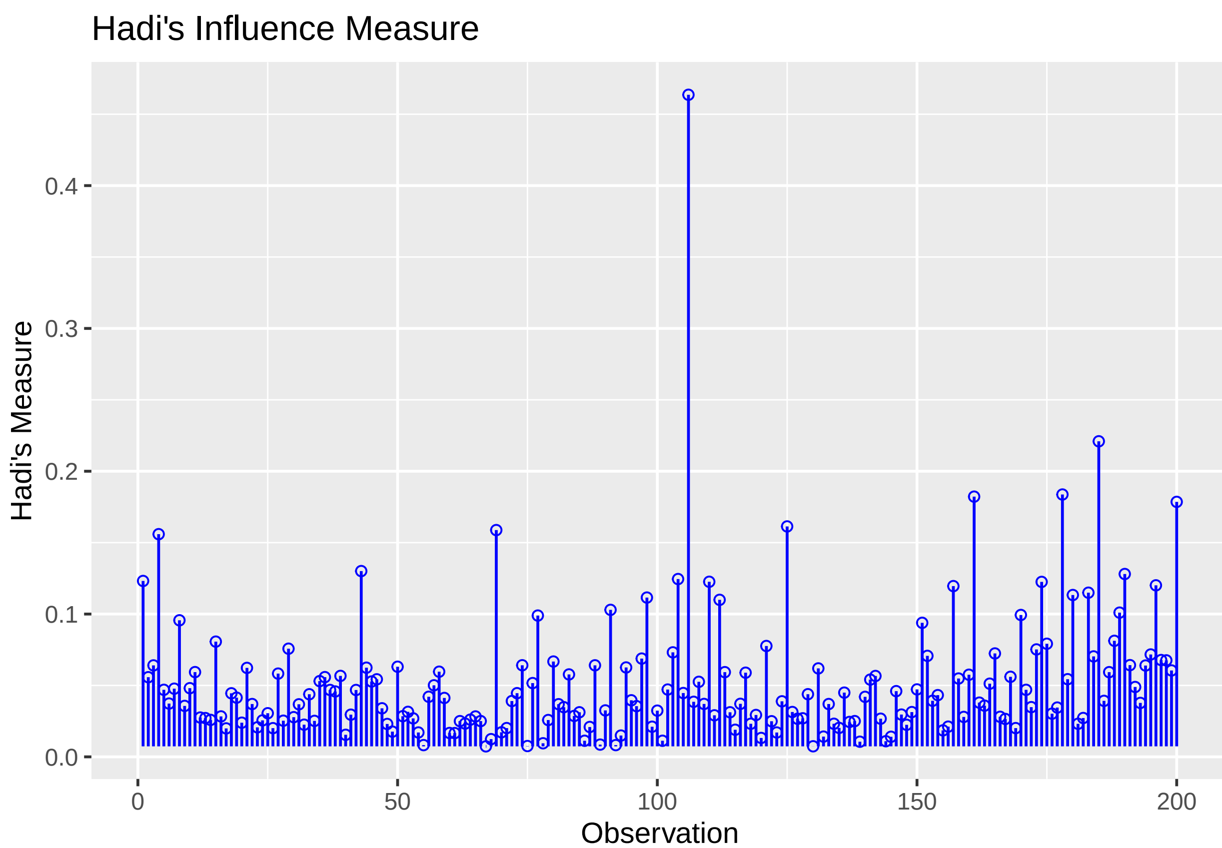

Influence Diagnostics for (Interce
Influence Diagnostics for tempera
Influence Diagnostics for SWF
Influence Diagnostics for size
Threshold: 0.14
DFBETAS
Observation




```r
#Difference in fit chart for each sample
ols_plot_dffits(mod)
```




```
#Plot for observation influence using hadi's distance
ols_plot_hadi(mod)
```



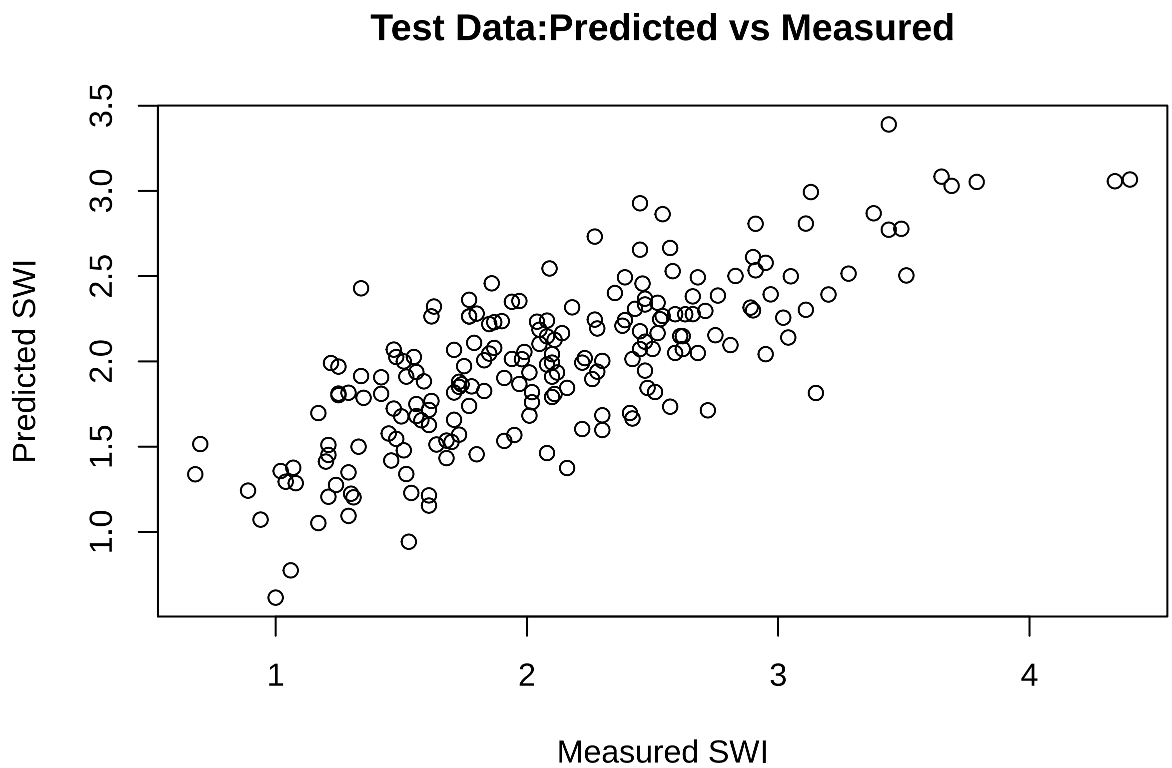
A several outliers are detected in sample index 69, 106, 161, 178, 185, and 200. Samples 104, 110, and 157 are potential outliers. Sample 169 proves to be highly influential

Evaluating test data performance with RMSE and a plot.

```
results.basic=predict(fit.basic,data.test)
rmse(results.basic,data.test$SWI)
```

```
## [1] 0.4481482
```

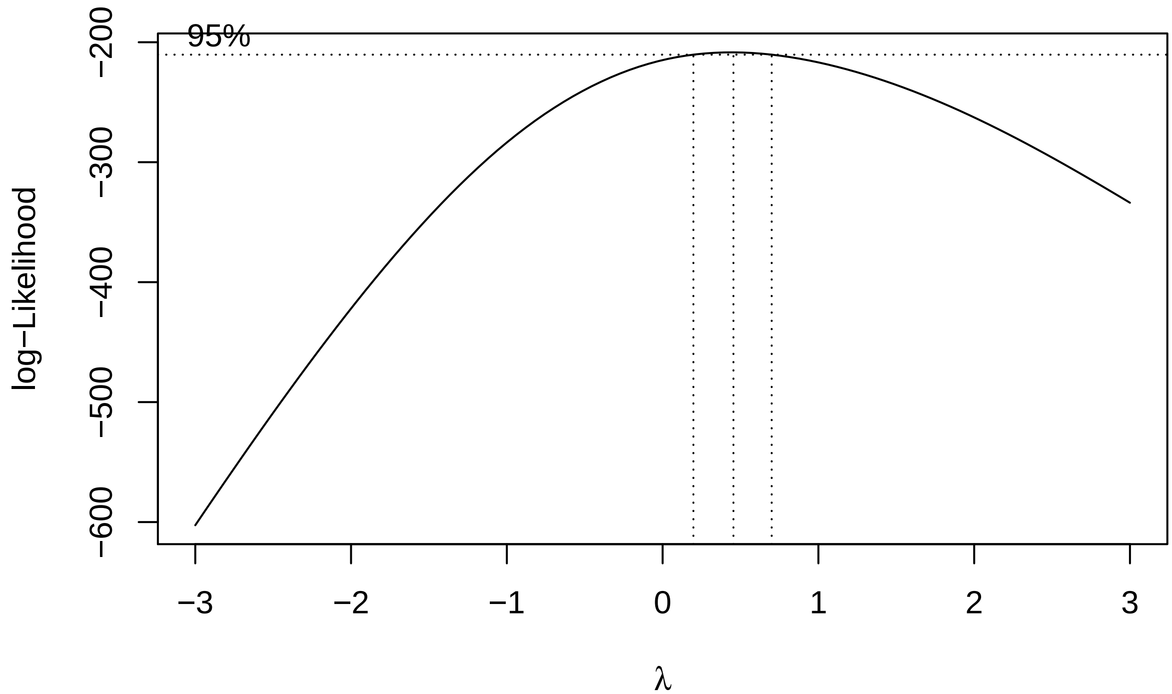
```
plot(data.test$SWI,results.basic,
     xlab="Measured SWI",ylab="Predicted SWI",main="Test Data:Predicted vs Measured")
```



The basic model has a poor mapping to the actual data. Problems with assumptions need to be adressed.

Box-Cox Transform to address homoscedasticity.Size removed because its insignificant, full model with interaction effects considered for ox cox transform

```
fullmodel=lm(SWI~SWF*temperature*management,data=data.training)
bc=boxcox(fullmodel,lambda = seq(-3,3))
```




```r
lambda=bc$x[which(bc$y==max(bc$y))]
#measured lambda
lambda
```

```
## [1] 0.4545455
```

```r
fit.bc=lm(SWI^lambda~SWF*temperature*management,data=data.training)
summary(fit.bc)
```

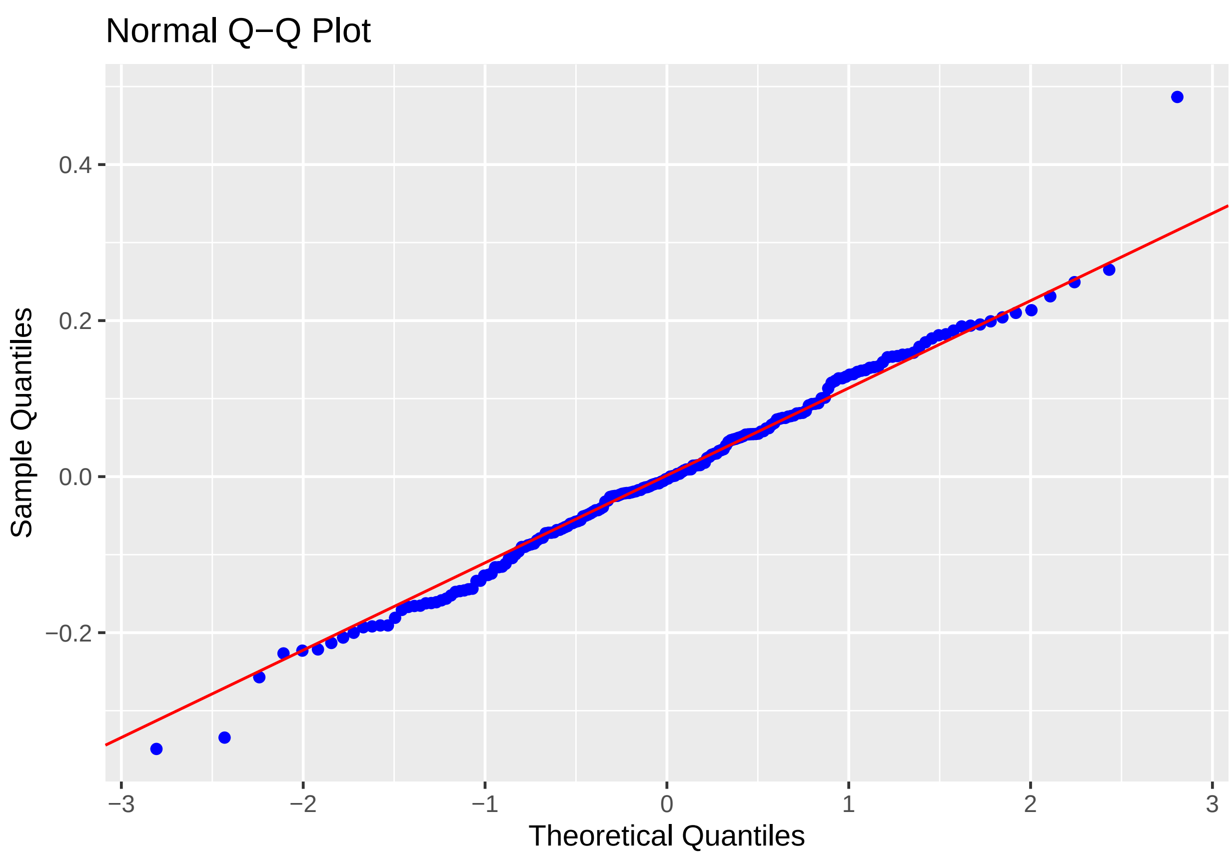
```
##
## Call:
## lm(formula = SWI^lambda ~ SWF * temperature * management, data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34903 -0.07397 -0.00286  0.07710  0.48664
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.260388   0.210552   1.237   0.2177
## SWF                          0.486445   0.150045   3.242   0.0014 **
## temperature                  0.027846   0.010756   2.589   0.0104 *
## management                   0.066891   0.041767   1.602   0.1109
## SWF:temperature             -0.008817   0.007499  -1.176   0.2411
## SWF:management              -0.032088   0.027864  -1.152   0.2509
## temperature:management      -0.001350   0.002187  -0.617   0.5379
## SWF:temperature:management   0.000984   0.001446   0.681   0.4970
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1238 on 192 degrees of freedom
## Multiple R-squared:  0.6804, Adjusted R-squared:  0.6688
## F-statistic:  58.4 on 7 and 192 DF,  p-value: < 2.2e-16
```

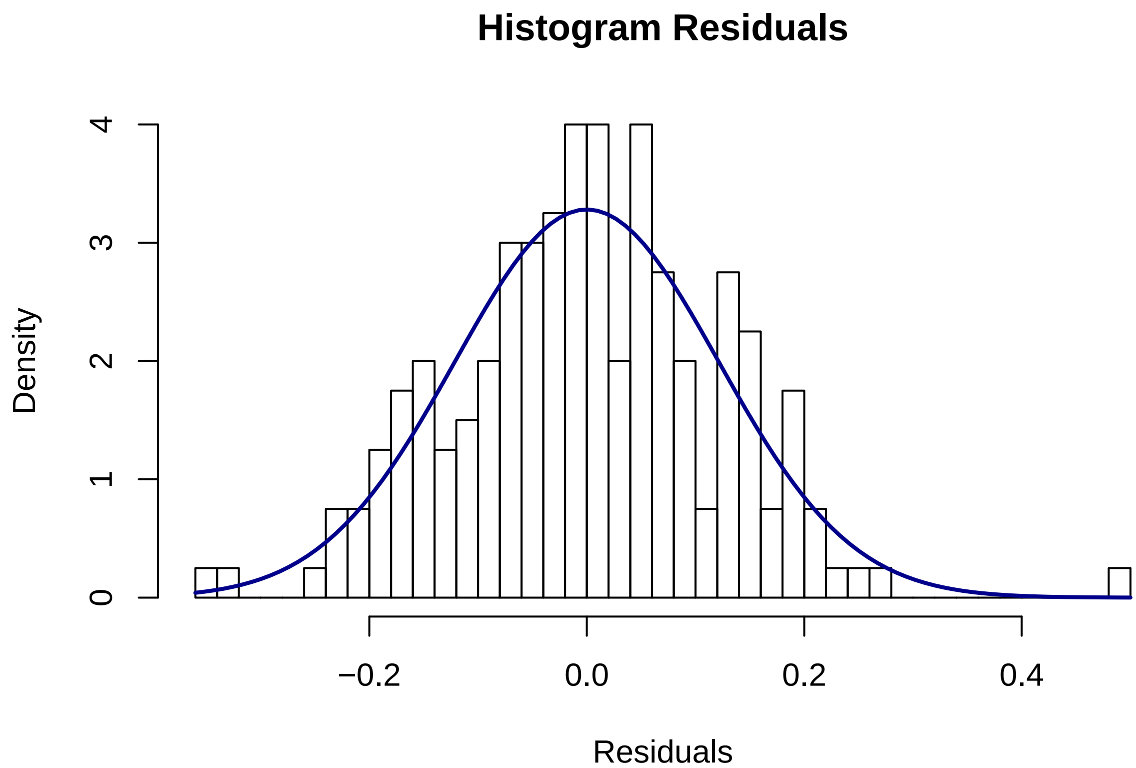
lambda of .45455

model assumptions for full interaction model

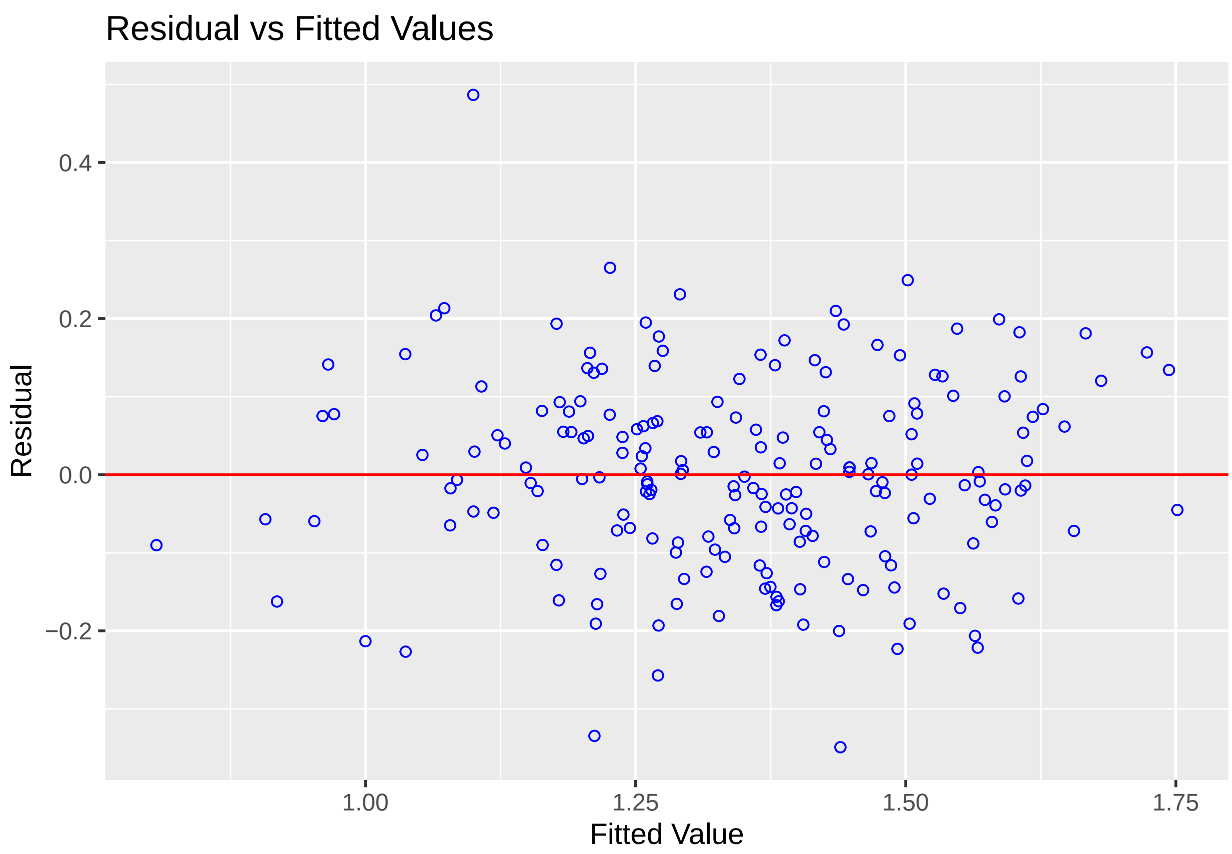
```
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.bc)
```




```
hist(fit.bc$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.bc$residuals), sd=sd(fit.bc$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.bc)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.bc)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.128714, Df = 1, p = 0.28805
```

```
#Normality of errors tests
ols_test_normality(fit.bc)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9903         0.1960
## Kolmogorov-Smirnov        0.0402         0.9023
## Cramer-von Mises         51.8664         0.0000
## Anderson-Darling          0.2579         0.7154
## -----------------------------------------------
```

```
ols_test_correlation(fit.bc)
```

```
## [1] 0.993773
```

Variance is now constant. All assumptions are met.

Performing model trimming to only th most influential variables.

```
fit.bc.full=lm(SWI^lambda~SWF*temperature*management,data=data.training)
fit.bc.null=lm(SWI^lambda~1,data=data.training)
```

Stepwise variable selection

Fitting the forward selection. backwards elimination, and both-ways selection model

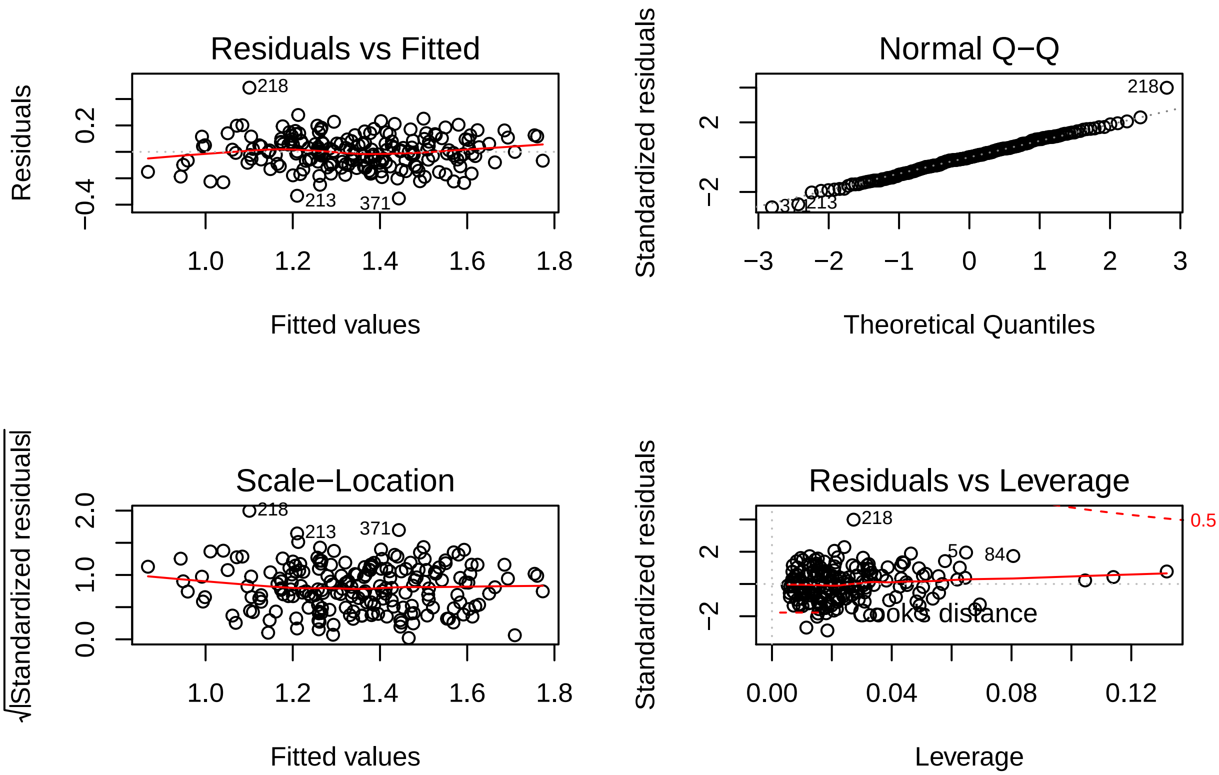
```
fit.bc.backwards=
    lm(formula = SWI^lambda ~ SWF + temperature + management + SWF:management,
    data = data.training)

fit.bc.forward=
  lm(formula = SWI^lambda ~ SWF + temperature + management + SWF:management,
    data = data.training)

fit.bc.both=
    lm(formula = SWI^lambda ~ SWF + temperature + management + SWF:management,
    data = data.training)
```

Box-cox+variable elimination model diagnostics

```
summary(fit.bc.backwards)
```

```
##
## Call:
## lm(formula = SWI^lambda ~ SWF + temperature + management + SWF:management,
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35281 -0.07840 -0.00021  0.07708  0.48590
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.503743   0.057804   8.715 1.26e-15 ***
## SWF             0.311480   0.035253   8.836 5.82e-16 ***
## temperature     0.015378   0.001642   9.368  < 2e-16 ***
## management      0.041077   0.011447   3.589 0.000421 ***
## SWF:management -0.012876   0.007191  -1.791 0.074918 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1235 on 195 degrees of freedom
## Multiple R-squared:  0.6769, Adjusted R-squared:  0.6702
## F-statistic: 102.1 on 4 and 195 DF,  p-value: < 2.2e-16
```

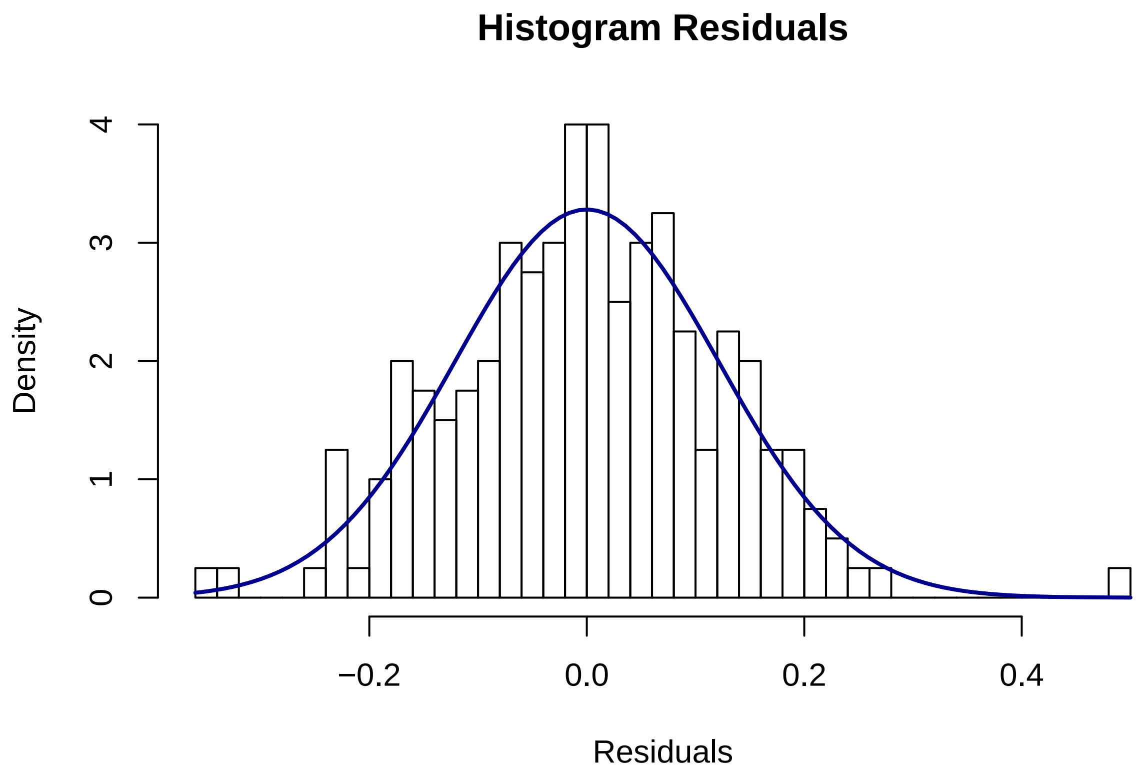
```
par(mfrow = c(2,2))
plot(fit.bc.backwards)
```




```r
#normality test
ols_test_normality(fit.bc.backwards)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9919         0.3289
## Kolmogorov-Smirnov        0.0375         0.9415
## Cramer-von Mises         51.7743         0.0000
## Anderson-Darling          0.1785         0.9176
## -----------------------------------------------
```

```r
par(mfrow = c(1,1))
hist(fit.bc.backwards$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.bc$residuals), sd=sd(fit.bc$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

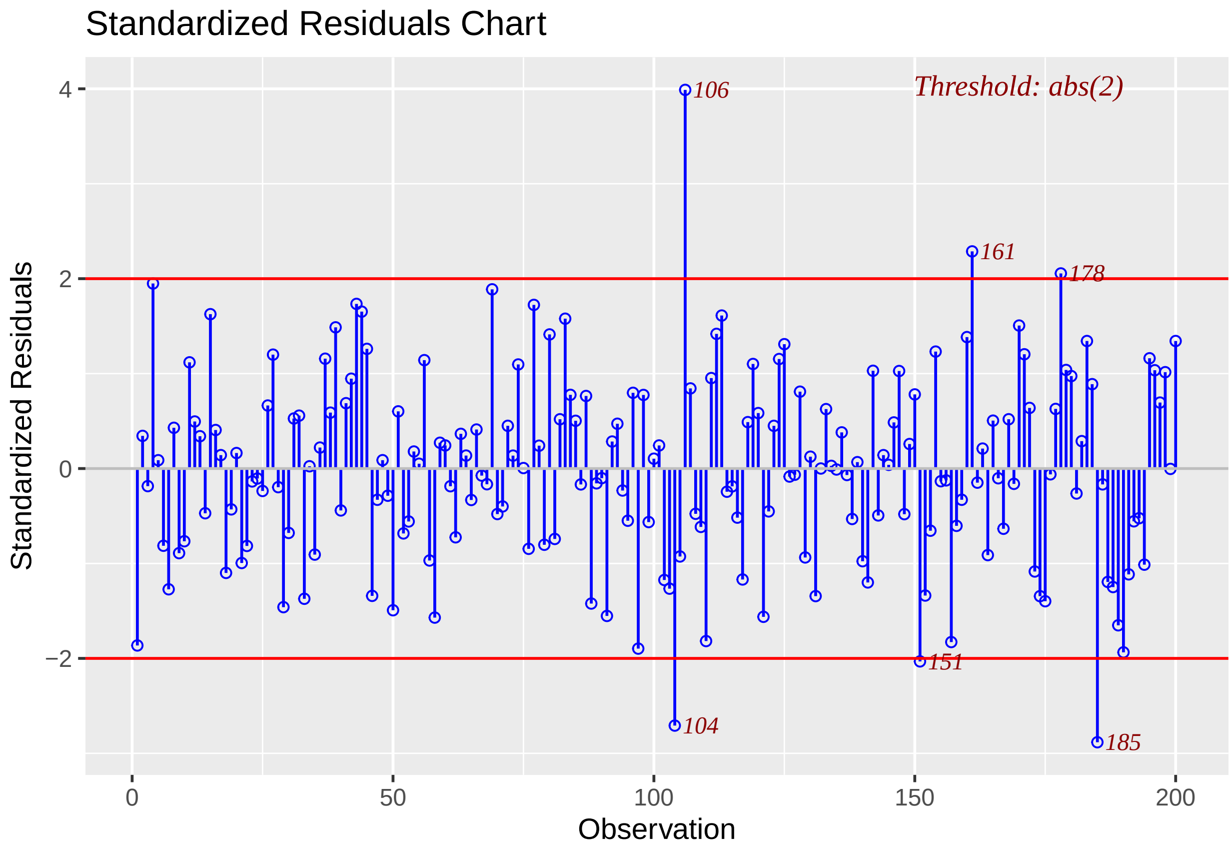



```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.bc.backwards)
```

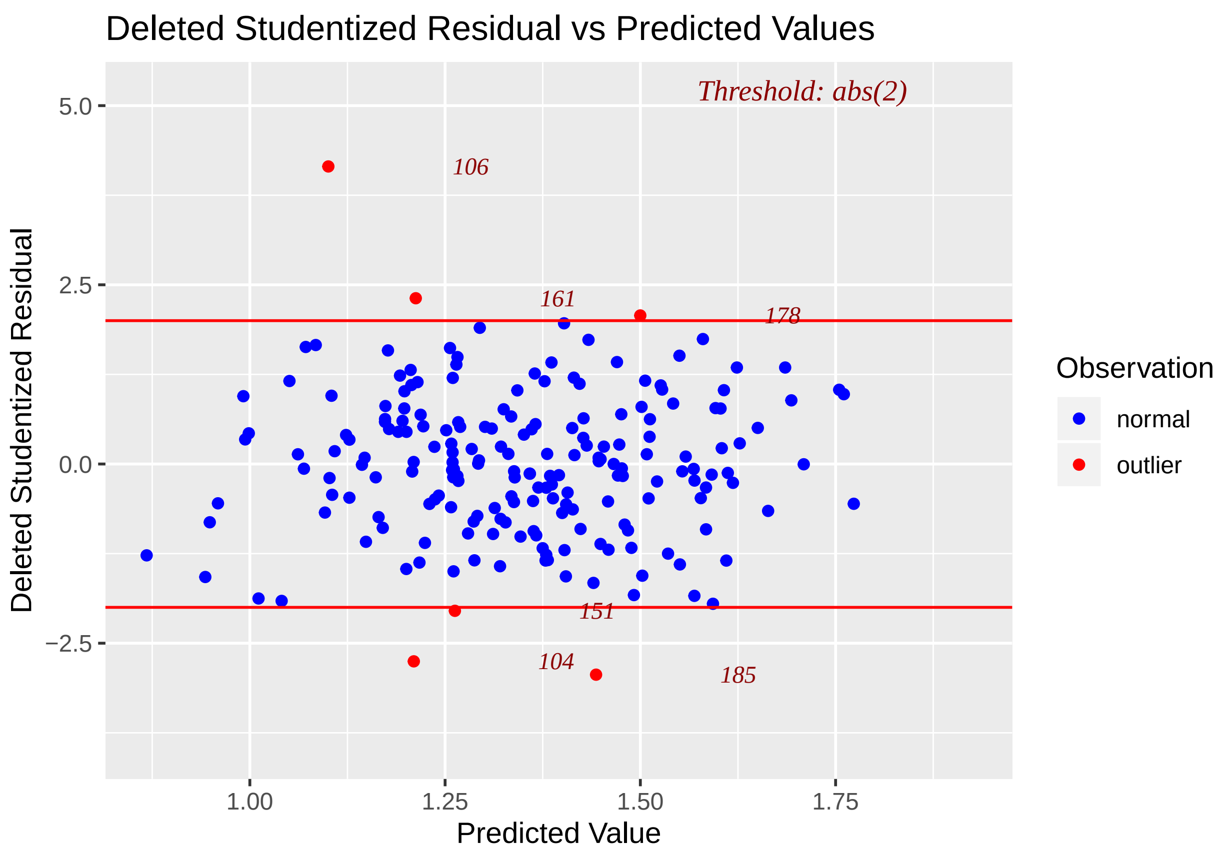
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.708554, Df = 1, p = 0.19117
```

```
fit.bc=fit.bc.backwards
```

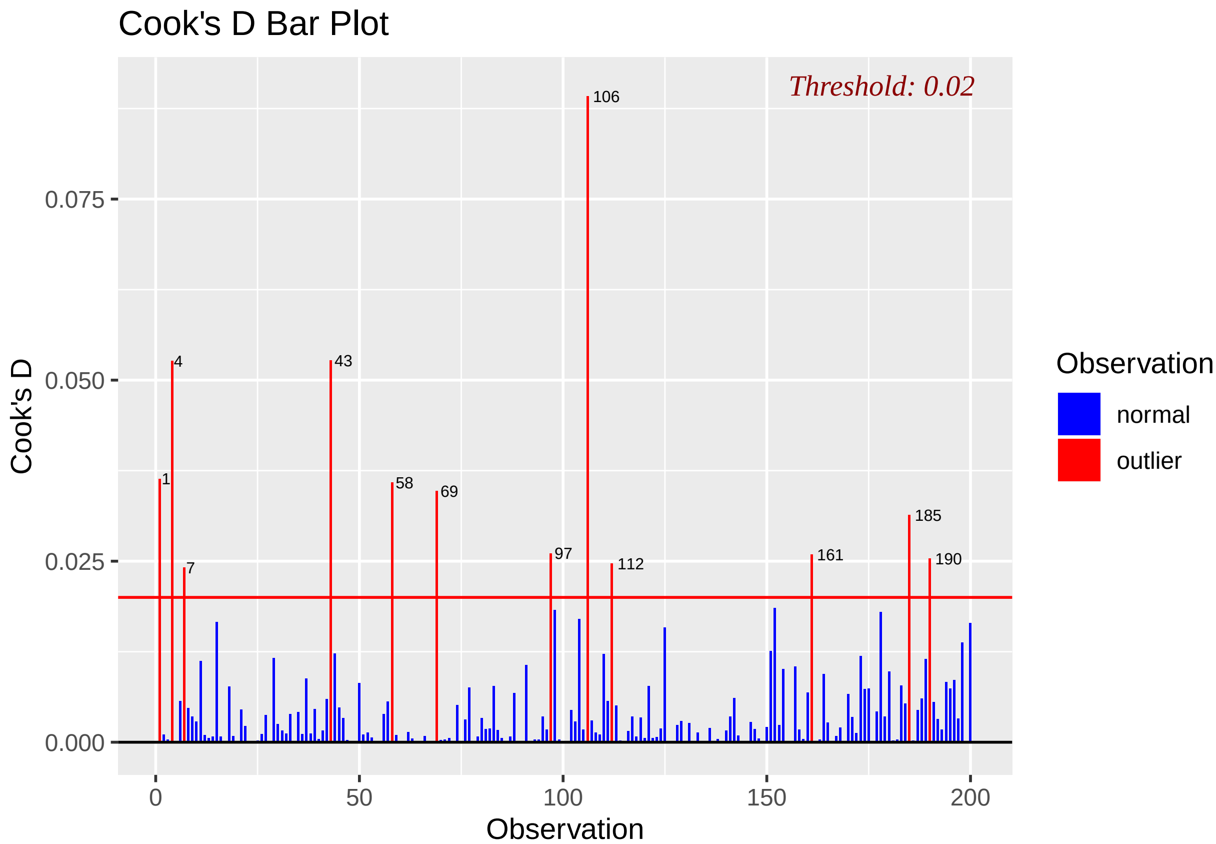
```
mod=fit.bc
#Standardized Residual plot
ols_plot_resid_stand(mod)
```



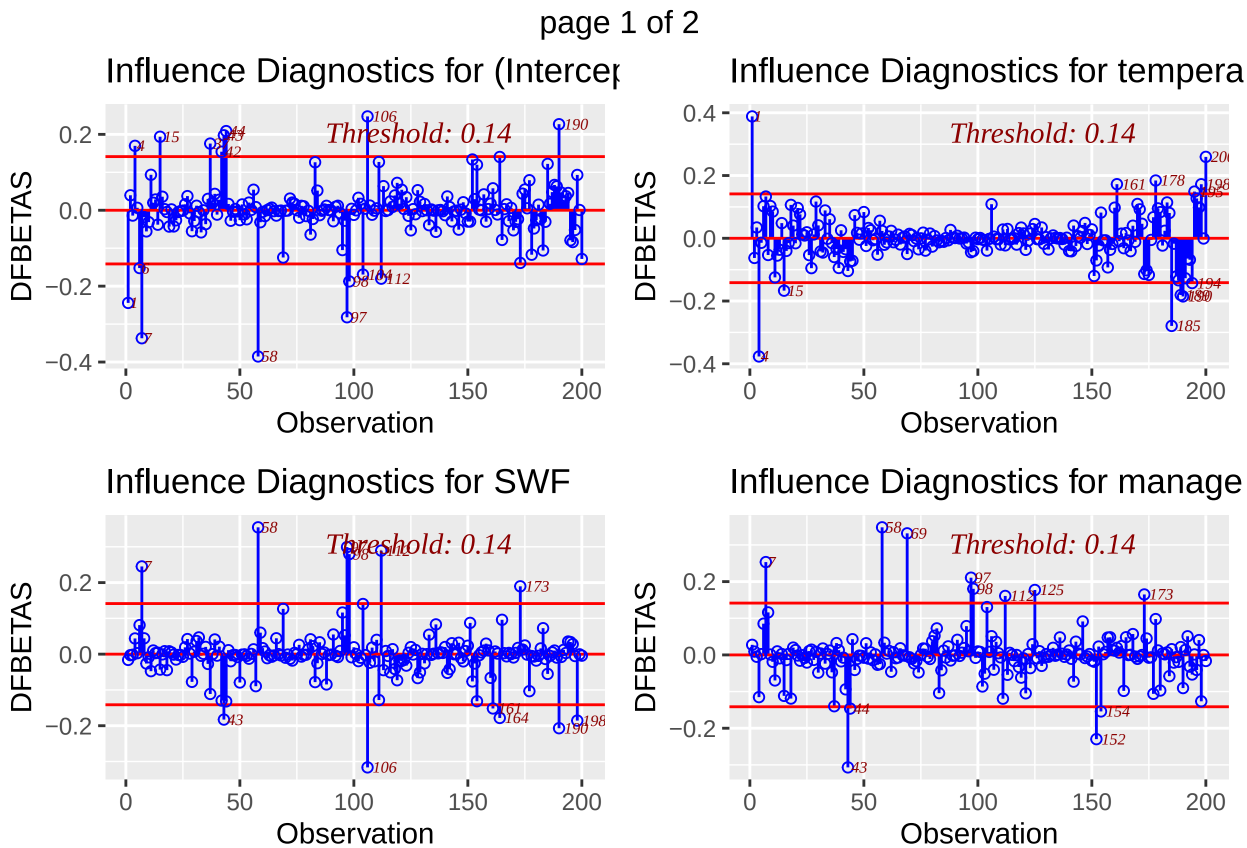

```r
#studentized residual plot
ols_plot_resid_stud_fit(mod)
```

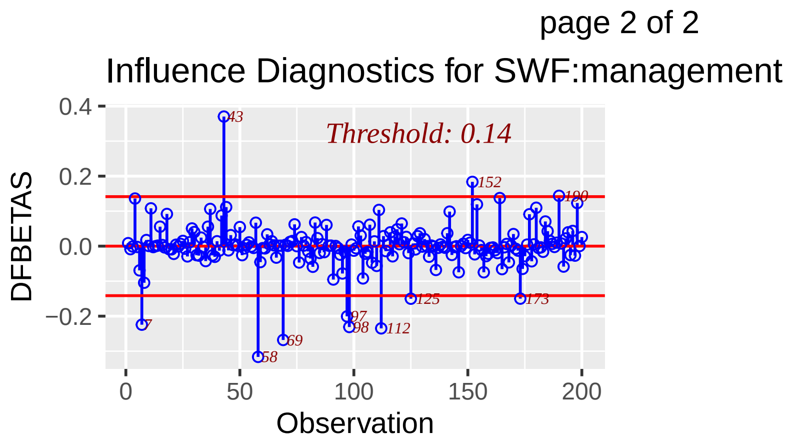



```
#cooks distance plot
ols_plot_cooksd_bar(mod)
```

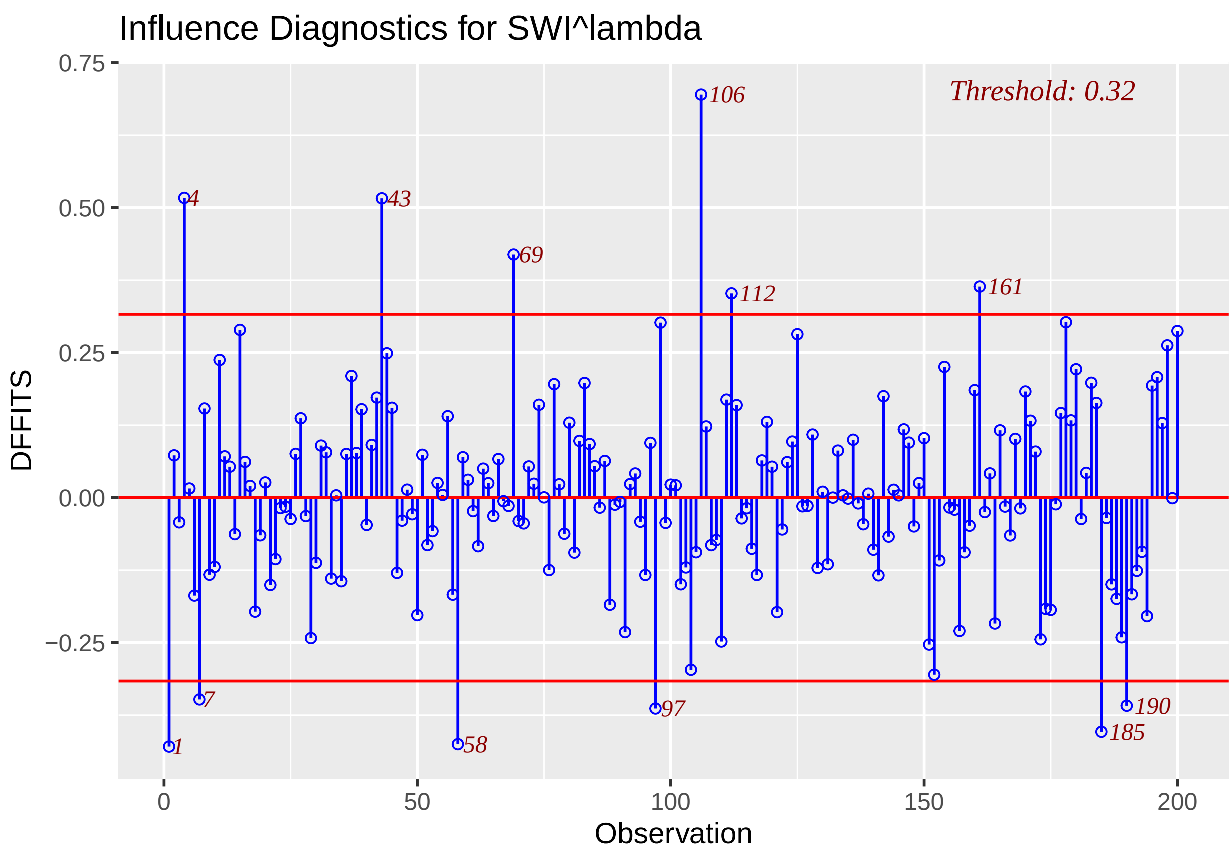



```r
#DFBetas for each variable
ols_plot_dfbetas(mod)
```

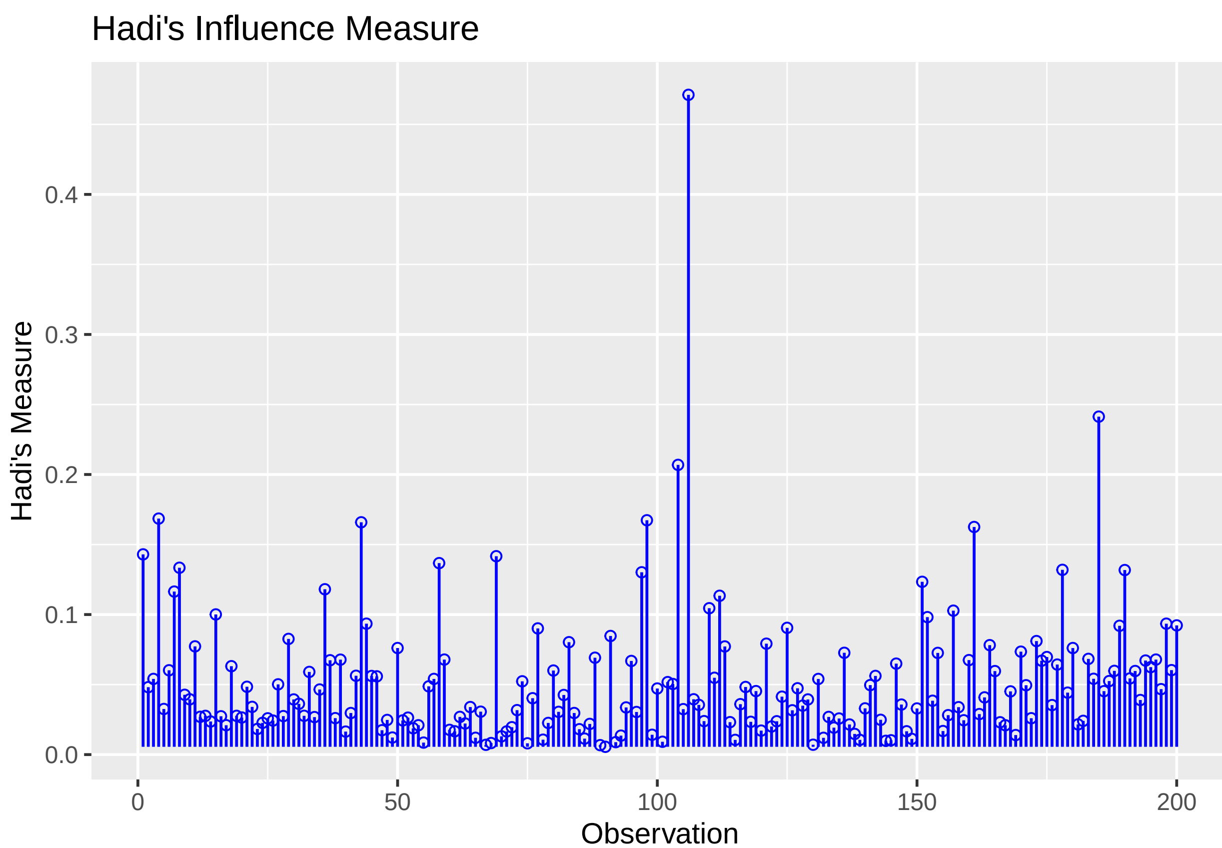

page 1 of 2
Influence Diagnostics for (Intercep
Threshold: 0.14
DFBETAS
Observation
Influence Diagnostics for tempera
Threshold: 0.14
DFBETAS
Observation
Influence Diagnostics for SWF
Threshold: 0.14
DFBETAS
Observation
Influence Diagnostics for manage
Threshold: 0.14
DFBETAS
Observation




```r
#Difference in fit chart for each sample
ols_plot_dffits(mod)
```




```r
#Plot for observation influence using hadi's distance
ols_plot_hadi(mod)
```

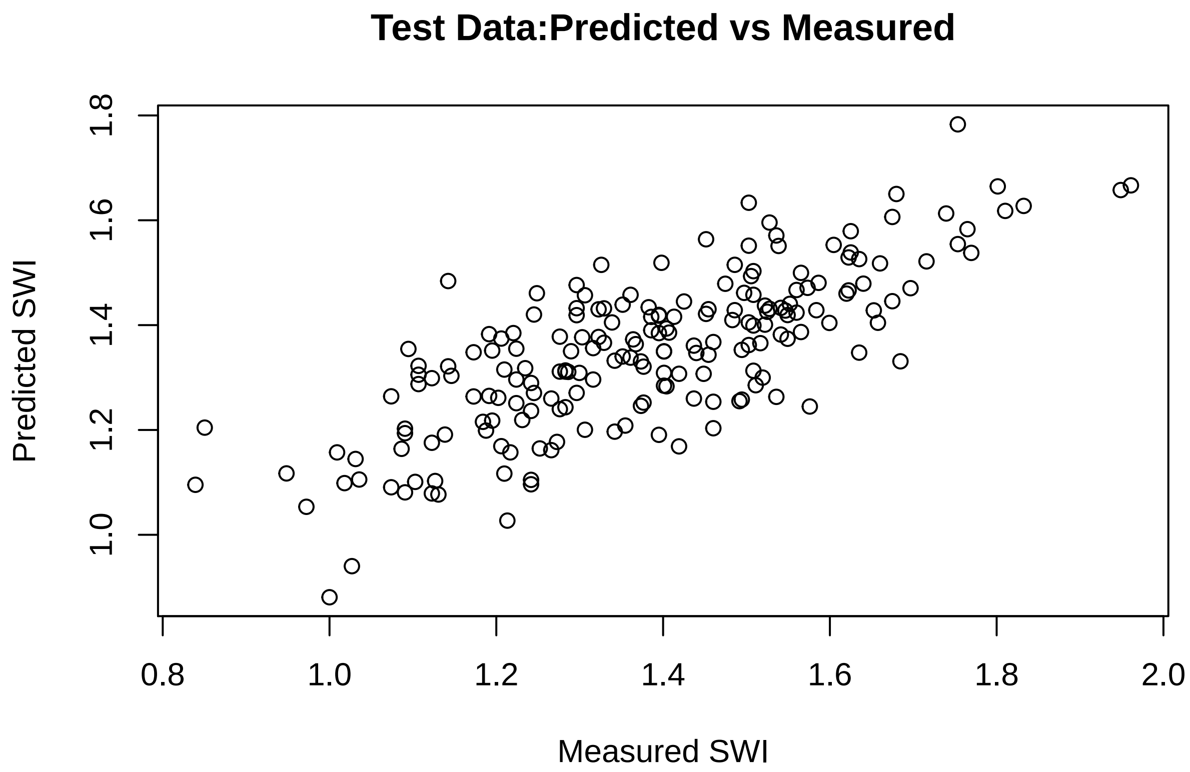


All assumptions still met, outliers still present.

```
results.bc=predict(fit.bc,data.test)
rmse(results.bc,data.test$SWI^lambda)
```

```
## [1] 0.1362177
```

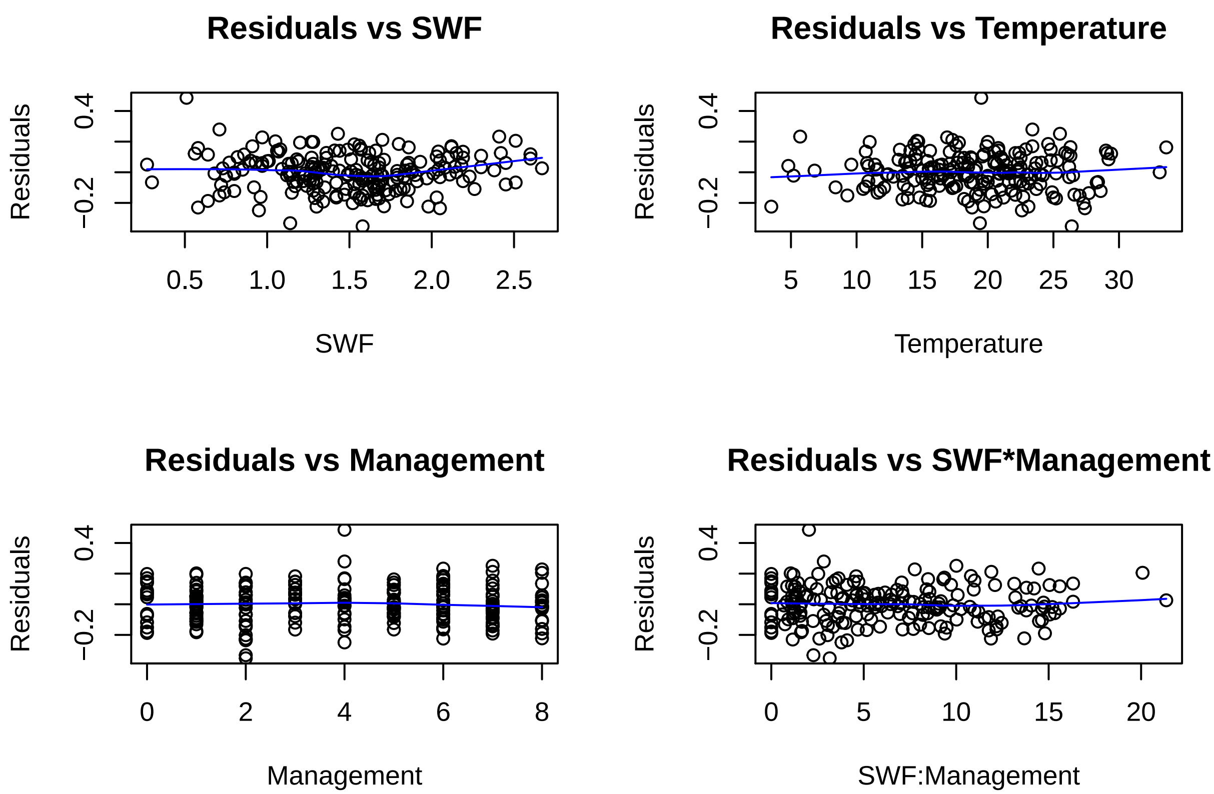
```
plot(data.test$SWI^lambda,results.bc,xlab="Measured SWI",ylab="Predicted SWI",main="Test Data:Predicted
```



Model sees drastic improvemt in RMSE after variable selection and box cox transfrom.

Checking variable linearity

```
par(mfrow=c(2,2))
#Checking linearity of SWF
plot(data.training$SWF,fit.bc$residuals,main="Residuals vs SWF",xlab="SWF",ylab="Residuals")
lines(lowess(fit.bc$residuals~data.training$SWF),col="blue")

#Checking linearity of Temperature
plot(data.training$temperature,fit.bc$residuals,main="Residuals vs Temperature",xlab="Temperature",ylab=
lines(lowess(fit.bc$residuals~data.training$temperature),col="blue")

#Checking linearity of Management
plot(data.training$management,fit.bc$residuals,main="Residuals vs Management",xlab="Management",ylab="Re
lines(lowess(fit.bc$residuals~data.training$management),col="blue")

plot(data.training$management*data.training$SWF,fit.bc$residuals,main="Residuals vs SWF*Management",xlab
lines(lowess(fit.bc$residuals~data.training$management*data.training$SWF),col="blue")
```



SWF Appears to be non-linear. Polynomial effects will be tested to see if they bring significance

Fitting polynomial model with interaction terms and SWF^2.

```
fit.poly=
  lm(formula = SWI ~ SWF * temperature * management * I(SWF^2), data = data.training)
summary(fit.poly)
```

```
##
## Call:
## lm(formula = SWI ~ SWF * temperature * management * I(SWF^2),
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93183 -0.22112 -0.00805  0.22828  1.21197
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           -3.9238033  4.4174404  -0.888    0.376
## SWF                                    9.1872558 10.6882098   0.860    0.391
## temperature                            0.1782457  0.2337267   0.763    0.447
## management                             0.4765277  0.7137731   0.668    0.505
## I(SWF^2)                              -5.8771375  7.9605384  -0.738    0.461
## SWF:temperature                       -0.3037308  0.5644875  -0.538    0.591
## SWF:management                        -0.8985644  1.7250861  -0.521    0.603
## temperature:management                 0.0017881  0.0417186   0.043    0.966
## SWF:I(SWF^2)                           1.2919949  1.8415574   0.702    0.484
```

```
## temperature:I(SWF^2)                0.2006896  0.4194884   0.478    0.633
## management:I(SWF^2)                 0.5541672  1.2807618   0.433    0.666
## SWF:temperature:management          0.0016545  0.0983606   0.017    0.987
## SWF:temperature:I(SWF^2)           -0.0411284  0.0969055  -0.424    0.672
## SWF:management:I(SWF^2)            -0.1076724  0.2942715  -0.366    0.715
## temperature:management:I(SWF^2)    -0.0010898  0.0719051  -0.015    0.988
## SWF:temperature:management:I(SWF^2)  0.0001218  0.0163892   0.007    0.994
##
## Residual standard error: 0.3812 on 184 degrees of freedom
## Multiple R-squared:  0.7111, Adjusted R-squared:  0.6876
## F-statistic:  30.2 on 15 and 184 DF,  p-value: < 2.2e-16
```

```
anova(fit.poly)
```

```
## Analysis of Variance Table
##
## Response: SWI
##                                      Df Sum Sq Mean Sq  F value    Pr(>F)
## SWF                                   1 43.515  43.515 299.5144 < 2.2e-16
## temperature                           1 13.431  13.431  92.4465 < 2.2e-16
## management                            1  4.765   4.765  32.7959 4.105e-08
## I(SWF^2)                              1  2.831   2.831  19.4888 1.722e-05
## SWF:temperature                       1  0.006   0.006   0.0427    0.8365
## SWF:management                        1  0.282   0.282   1.9376    0.1656
## temperature:management                1  0.183   0.183   1.2600    0.2631
## SWF:I(SWF^2)                          1  0.110   0.110   0.7600    0.3844
## temperature:I(SWF^2)                  1  0.001   0.001   0.0079    0.9295
## management:I(SWF^2)                   1  0.125   0.125   0.8626    0.3542
## SWF:temperature:management            1  0.000   0.000   0.0012    0.9726
## SWF:temperature:I(SWF^2)              1  0.115   0.115   0.7907    0.3750
## SWF:management:I(SWF^2)               1  0.444   0.444   3.0581    0.0820
## temperature:management:I(SWF^2)       1  0.001   0.001   0.0047    0.9457
## SWF:temperature:management:I(SWF^2)   1  0.000   0.000   0.0001    0.9941
## Residuals                           184 26.732   0.145
##
## SWF                                 ***
## temperature                         ***
## management                          ***
## I(SWF^2)                            ***
## SWF:temperature
## SWF:management
## temperature:management
## SWF:I(SWF^2)
## temperature:I(SWF^2)
## management:I(SWF^2)
## SWF:temperature:management
## SWF:temperature:I(SWF^2)
## SWF:management:I(SWF^2)             .
## temperature:management:I(SWF^2)
## SWF:temperature:management:I(SWF^2)
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Stepwise variable section for polynomial model both ways model = forward selection model

fitting each model and comparing them against each other.

```
fit.poly.backward=
  lm(formula = SWI ~ SWF + temperature + management + I(SWF^2) +
    SWF:temperature + SWF:management + SWF:I(SWF^2) + temperature:I(SWF^2) +
    management:I(SWF^2) + SWF:temperature:I(SWF^2) + SWF:management:I(SWF^2),
    data = data.training)
summary(fit.poly.backward)
```

```
##
## Call:
## lm(formula = SWI ~ SWF + temperature + management + I(SWF^2) +
##     SWF:temperature + SWF:management + SWF:I(SWF^2) + temperature:I(SWF^2) +
##     management:I(SWF^2) + SWF:temperature:I(SWF^2) + SWF:management:I(SWF^2),
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97111 -0.22625 -0.01605  0.22938  1.21856
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -4.10952    1.98707  -2.068  0.04000 *
## SWF                       9.32558    4.42890   2.106  0.03657 *
## temperature               0.18700    0.08345   2.241  0.02621 *
## management                0.50321    0.18992   2.650  0.00875 **
## I(SWF^2)                 -6.02560    3.07492  -1.960  0.05152 .
## SWF:temperature          -0.30731    0.18822  -1.633  0.10420
## SWF:management           -0.85401    0.41003  -2.083  0.03862 *
## SWF:I(SWF^2)              1.33376    0.66455   2.007  0.04618 *
## temperature:I(SWF^2)      0.20590    0.13117   1.570  0.11817
## management:I(SWF^2)       0.52431    0.27703   1.893  0.05995 .
## SWF:temperature:I(SWF^2) -0.04284    0.02835  -1.511  0.13238
## SWF:management:I(SWF^2)  -0.10325    0.05857  -1.763  0.07954 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3781 on 188 degrees of freedom
## Multiple R-squared:  0.7096, Adjusted R-squared:  0.6926
## F-statistic: 41.77 on 11 and 188 DF,  p-value: < 2.2e-16
```

```
fit.poly.forward=
 lm(formula = SWI ~ I(SWF^2) + temperature + management,
    data = data.training)
summary(fit.poly.forward)
```

```
##
## Call:
## lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01224 -0.25139 -0.01333  0.21775  1.37081
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.142086   0.110001   1.292    0.198
## I(SWF^2)    0.281397   0.018583  15.142  < 2e-16 ***
## temperature 0.047491   0.005039   9.425  < 2e-16 ***
## management  0.061961   0.010918   5.675 4.93e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3795 on 196 degrees of freedom
## Multiple R-squared:  0.6949, Adjusted R-squared:  0.6902
## F-statistic: 148.8 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
anova(fit.poly.backward,fit.poly.forward)
```

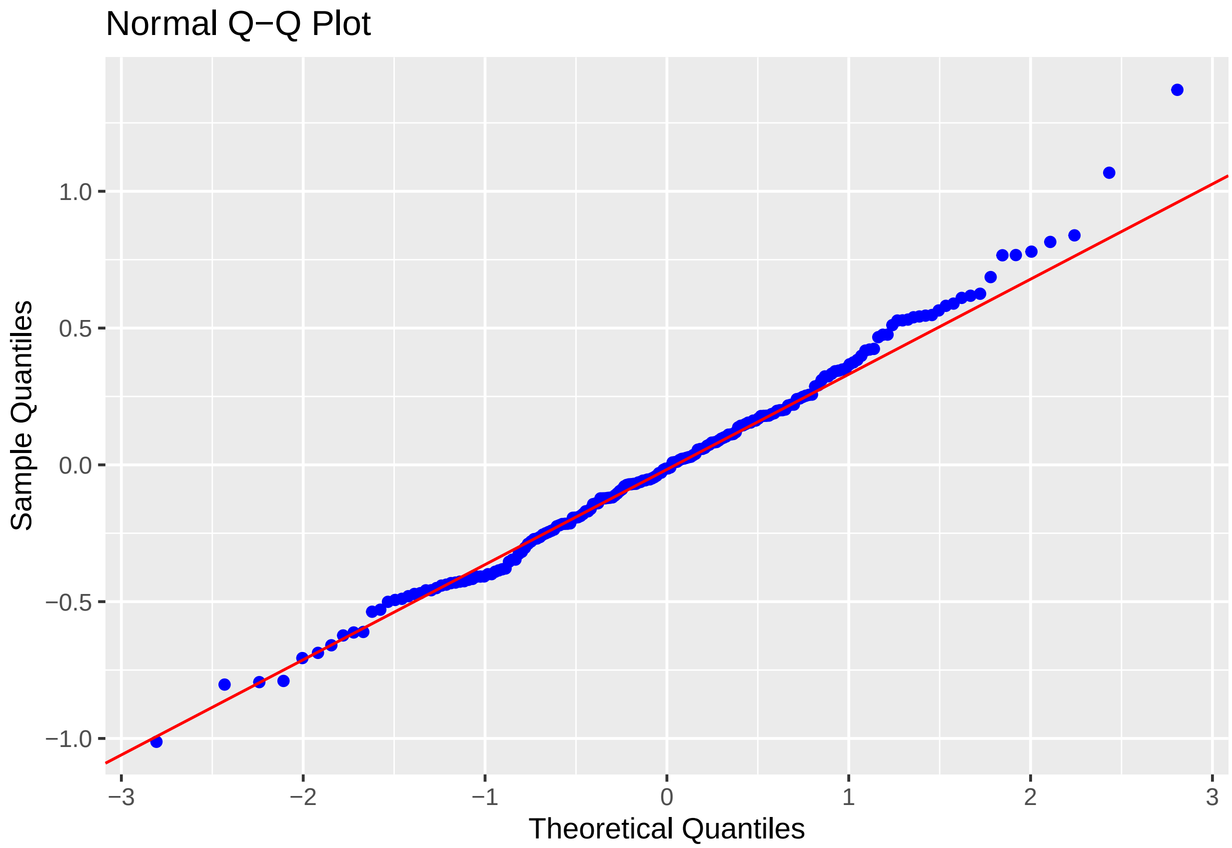
```
## Analysis of Variance Table
##
## Model 1: SWI ~ SWF + temperature + management + I(SWF^2) + SWF:temperature +
##     SWF:management + SWF:I(SWF^2) + temperature:I(SWF^2) + management:I(SWF^2) +
##     SWF:temperature:I(SWF^2) + SWF:management:I(SWF^2)
## Model 2: SWI ~ I(SWF^2) + temperature + management
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    188 26.871
## 2    196 28.235 -8   -1.3645 1.1934  0.305
```

```r
#no real difference found between the two models
fit.poly=fit.poly.forward
```

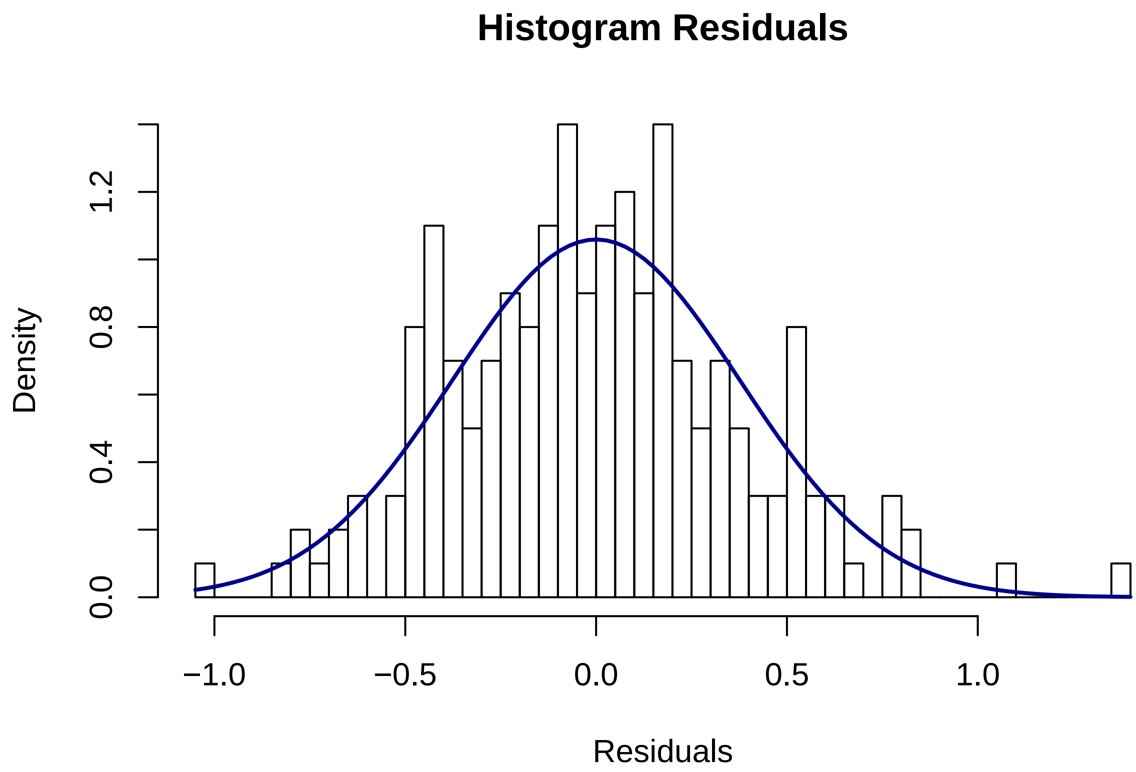
Anova model comparison shows no significant defference between the two models. Forward selection shall be used as it is substantially smaller than the backward elimination model. Interaction effects lose significane in the pressence of polynomial terms.

Checking model assumptions of the polynomial model.

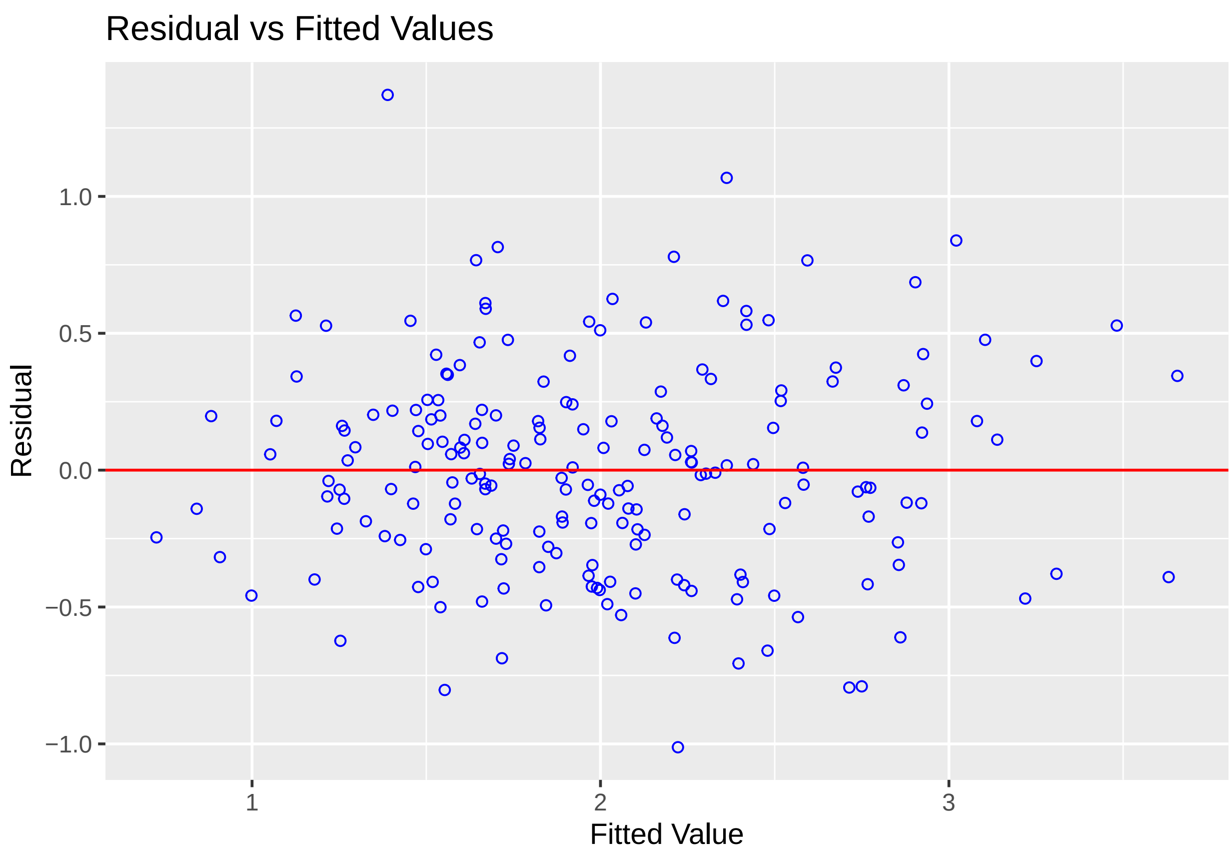
```r
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.poly)
```




```
hist(fit.poly$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.poly$residuals), sd=sd(fit.poly$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.poly)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.poly)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.096221, Df = 1, p = 0.078474
```

```
#Normality of errors tests
ols_test_normality(fit.poly)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9919          0.3364
## Kolmogorov-Smirnov        0.0406          0.8958
## Cramer-von Mises          30.616          0.0000
## Anderson-Darling          0.3461          0.4794
## -----------------------------------------------
```

```
ols_test_correlation(fit.poly)
```

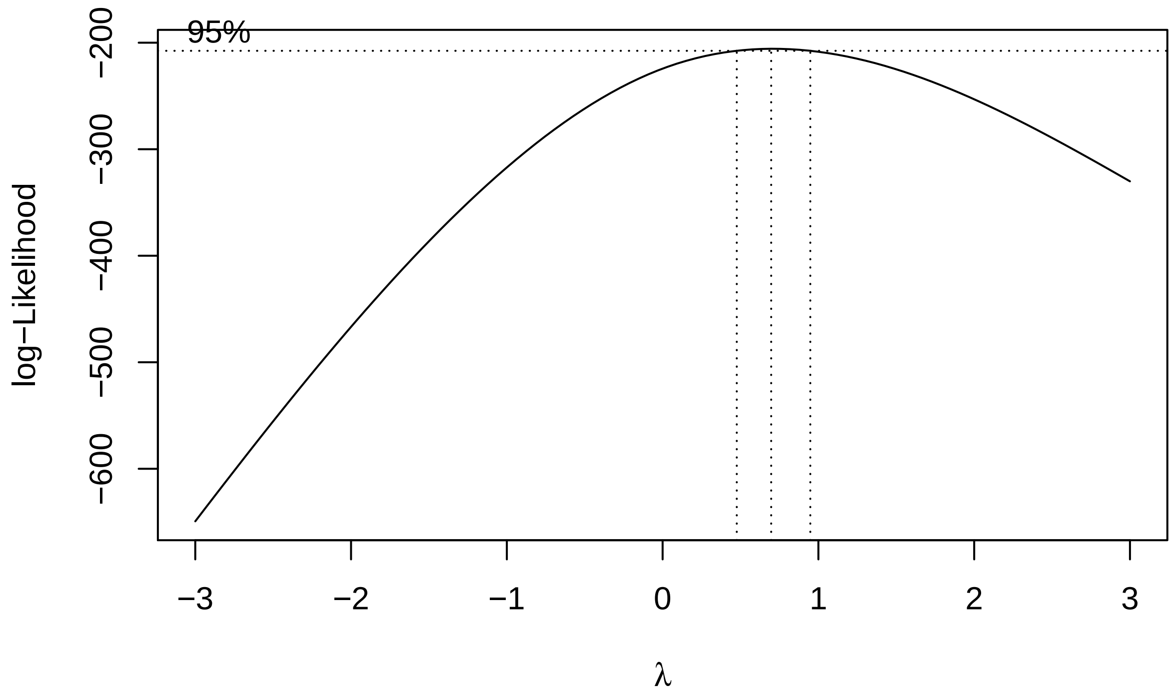
```
## [1] 0.9952231
```

Non-constant variance is an issue, will be addressed with a box cox transform. Other assumptions met.

Box-cox transform for polynomial model.

```
fullmodelpoly=
  lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.training)
```

```r
bcpoly=boxcox(fullmodelpoly,lambda = seq(-3,3))
```




```r
lambdapoly=bcpoly$x[which(bcpoly$y==max(bcpoly$y))]
lambdapoly
```

```
## [1] 0.6969697
```

```r
fit.bc.poly=
  lm(SWI^lambdapoly ~ I(SWF^2) + temperature + management, data = data.training)
summary(fit.bc.poly)
```

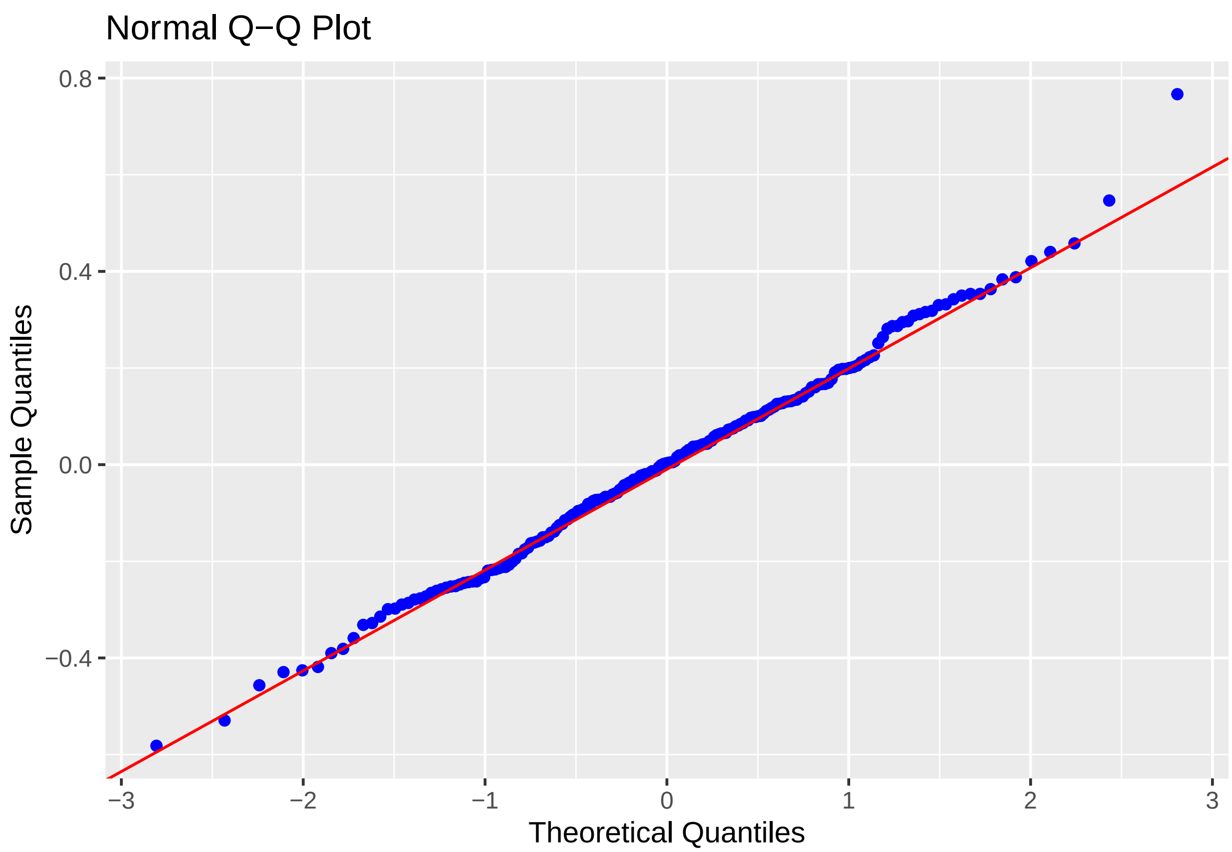
```
##
## Call:
## lm(formula = SWI^lambdapoly ~ I(SWF^2) + temperature + management,
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58175 -0.15022  0.00329  0.13104  0.76671
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.546052   0.062436   8.746 1.01e-15 ***
## I(SWF^2)    0.156010   0.010548  14.791  < 2e-16 ***
## temperature 0.027054   0.002860   9.460  < 2e-16 ***
## management  0.037048   0.006197   5.978 1.05e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2154 on 196 degrees of freedom
## Multiple R-squared:  0.6917, Adjusted R-squared:  0.687
## F-statistic: 146.6 on 3 and 196 DF,  p-value: < 2.2e-16
```

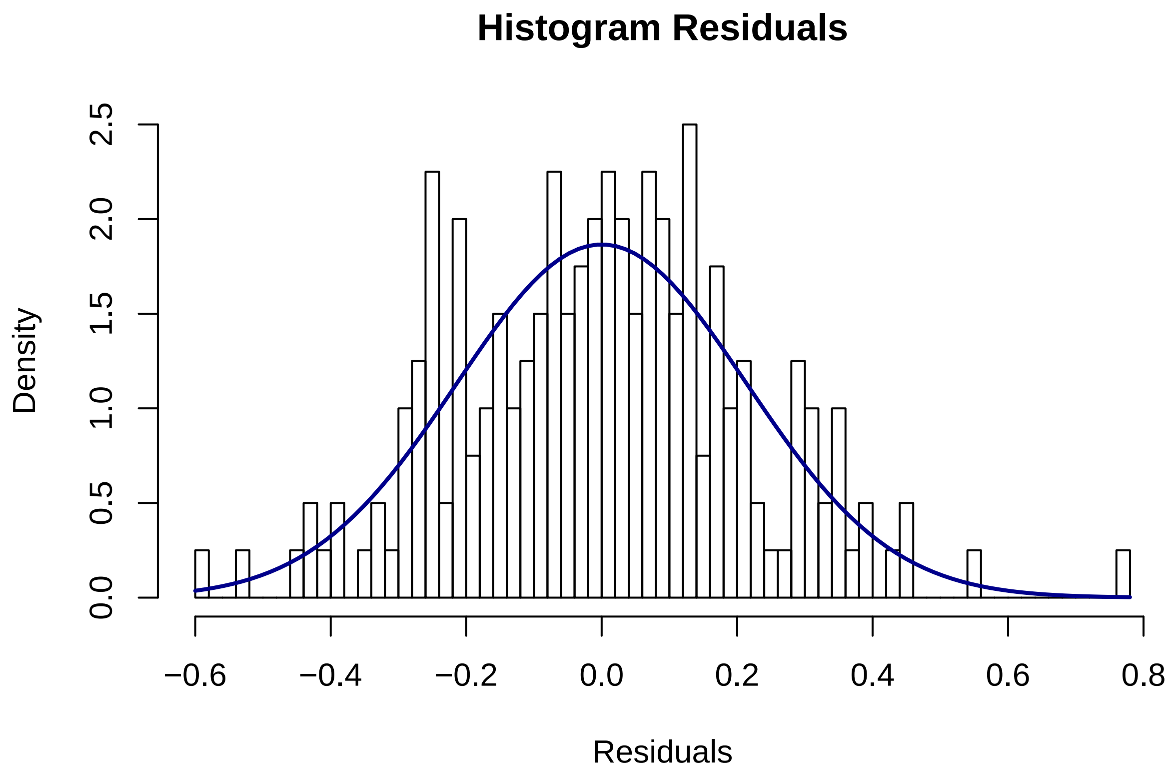
lambda of .69697

Model Diagnostics of box-cox transformed polynomial model

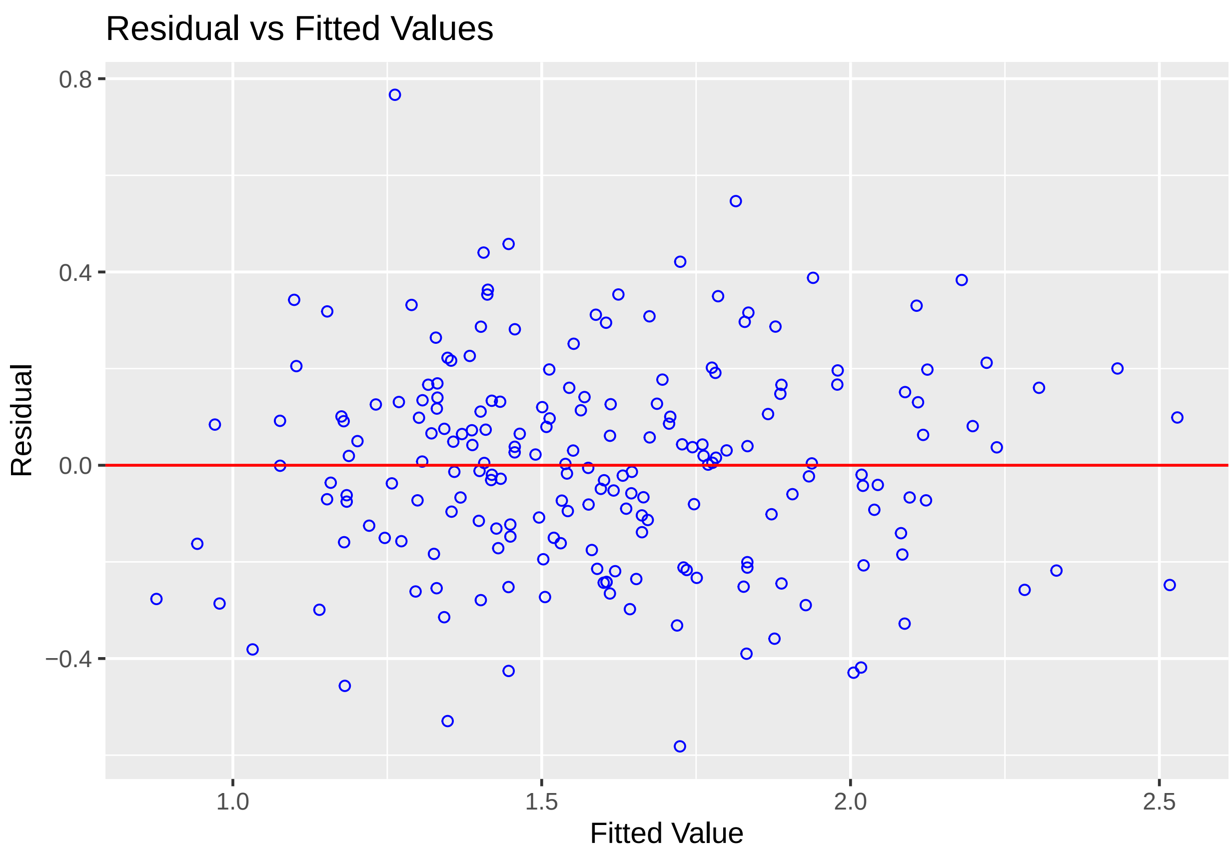
```
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.bc.poly)
```




```
hist(fit.bc.poly$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.bc.poly$residuals), sd=sd(fit.bc.poly$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.bc.poly)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.bc.poly)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.00738277, Df = 1, p = 0.93153
```

```
#Normality of errors tests
ols_test_normality(fit.bc.poly)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9952         0.7809
## Kolmogorov-Smirnov        0.0297         0.9946
## Cramer-von Mises         42.4756         0.0000
## Anderson-Darling          0.2129         0.8518
## -----------------------------------------------
```

```
ols_test_correlation(fit.bc.poly)
```

```
## [1] 0.9968848
```

All assumptions met.

Final model to be used for further analysis

```
fit.fin=fit.bc.poly
summary(fit.fin)
```

```
##
## Call:
## lm(formula = SWI^lambdapoly ~ I(SWF^2) + temperature + management,
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58175 -0.15022  0.00329  0.13104  0.76671
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.546052   0.062436   8.746 1.01e-15 ***
## I(SWF^2)    0.156010   0.010548  14.791  < 2e-16 ***
## temperature 0.027054   0.002860   9.460  < 2e-16 ***
## management  0.037048   0.006197   5.978 1.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2154 on 196 degrees of freedom
## Multiple R-squared:  0.6917, Adjusted R-squared:  0.687
## F-statistic: 146.6 on 3 and 196 DF,  p-value: < 2.2e-16
```

As an alternative model, an iteratively reweighted least squares model without the box cox transformation and only the main effects minus size and where SWF is taken to the 2nd power shall be considered.

```r
#Base Model
fit.alt.primary=
  lm(formula = SWI ~ I(SWF^2) + temperature + management,
     data = data.training)
summary(fit.alt.primary)
```

```
##
## Call:
## lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01224 -0.25139 -0.01333  0.21775  1.37081
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.142086   0.110001   1.292    0.198
## I(SWF^2)    0.281397   0.018583  15.142  < 2e-16 ***
## temperature 0.047491   0.005039   9.425  < 2e-16 ***
## management  0.061961   0.010918   5.675 4.93e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3795 on 196 degrees of freedom
## Multiple R-squared:  0.6949, Adjusted R-squared:  0.6902
## F-statistic: 148.8 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
#First itteration
resid=residuals(fit.alt.primary)
fit.alt.std=lm(abs(resid)~ I(SWF^2) + temperature + management, data = data.training)
summary(fit.alt.std)
```

```
##
## Call:
## lm(formula = abs(resid) ~ I(SWF^2) + temperature + management,
##     data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34608 -0.18305 -0.03833  0.12883  1.07880
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.088615   0.065975   1.343  0.18077
## I(SWF^2)    0.003972   0.011146   0.356  0.72193
## temperature 0.009990   0.003022   3.306  0.00113 **
## management  0.001890   0.006548   0.289  0.77314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2276 on 196 degrees of freedom
## Multiple R-squared:  0.0567, Adjusted R-squared:  0.04226
## F-statistic: 3.927 on 3 and 196 DF,  p-value: 0.009437
```

```
w=1/fit.alt.std$fitted^2
fit.alt1=lm(formula = SWI ~ I(SWF^2) + temperature + management,
           data = data.training, weights = w)
summary(fit.alt1)
```

```
##
## Call:
## lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.training,
##     weights = w)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7562 -0.9340 -0.0447  0.7948  4.7449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.142925   0.088352   1.618    0.107
## I(SWF^2)    0.288042   0.018296  15.744  < 2e-16 ***
## temperature 0.046763   0.004367  10.709  < 2e-16 ***
## management  0.061181   0.010080   6.070 6.52e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.271 on 196 degrees of freedom
## Multiple R-squared:  0.7268, Adjusted R-squared:  0.7226
## F-statistic: 173.8 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
#second iteration
resid1=residuals(fit.alt1)
fit.alt1.std=lm(abs(resid1)~ I(SWF^2) + temperature + management, data = data.training)
summary(fit.alt1.std)
```

```
##
```

```
## Call:
## lm(formula = abs(resid1) ~ I(SWF^2) + temperature + management,
##     data = data.training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3455 -0.1795 -0.0414  0.1433  1.0900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.086517   0.065965   1.312 0.191200
## I(SWF^2)    0.002407   0.011144   0.216 0.829198
## temperature 0.010142   0.003022   3.357 0.000948 ***
## management  0.002665   0.006547   0.407 0.684379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2276 on 196 degrees of freedom
## Multiple R-squared:  0.05765,    Adjusted R-squared:  0.04323
## F-statistic: 3.997 on 3 and 196 DF,  p-value: 0.008612
```

```
w1=1/fit.alt1.std$fitted^2
fit.alt2=lm(formula = SWI ~ I(SWF^2) + temperature + management,
           data = data.training, weights = w1)
summary(fit.alt2)
```

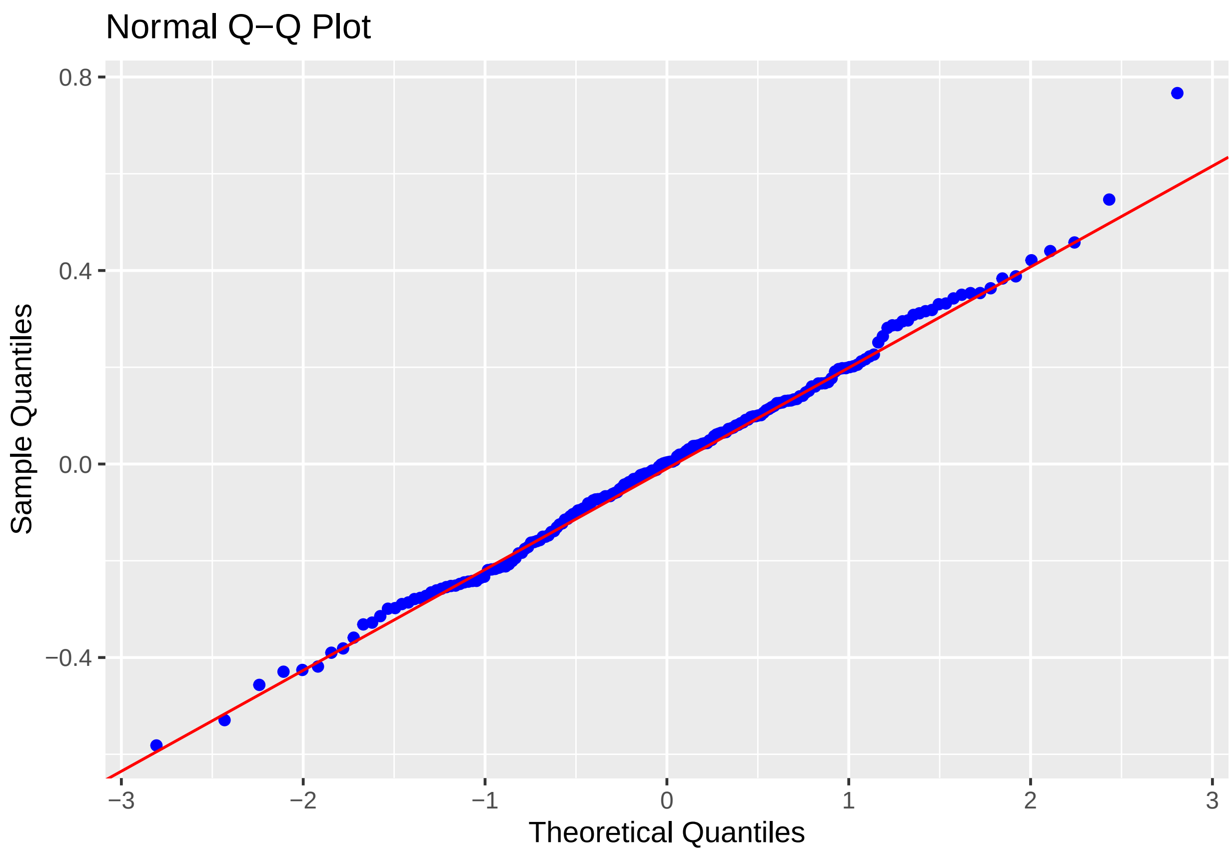
```
##
## Call:
## lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.training,
##     weights = w1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7578 -0.9322 -0.0453  0.8008  4.6982
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.142132   0.087878   1.617    0.107
## I(SWF^2)    0.289410   0.018172  15.926  < 2e-16 ***
## temperature 0.046648   0.004352  10.718  < 2e-16 ***
## management  0.061068   0.010063   6.069 6.55e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.271 on 196 degrees of freedom
## Multiple R-squared:  0.7299, Adjusted R-squared:  0.7258
## F-statistic: 176.6 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
#nothing really changed setting alternative model to the second iteration.
fit.alt=fit.alt2
```

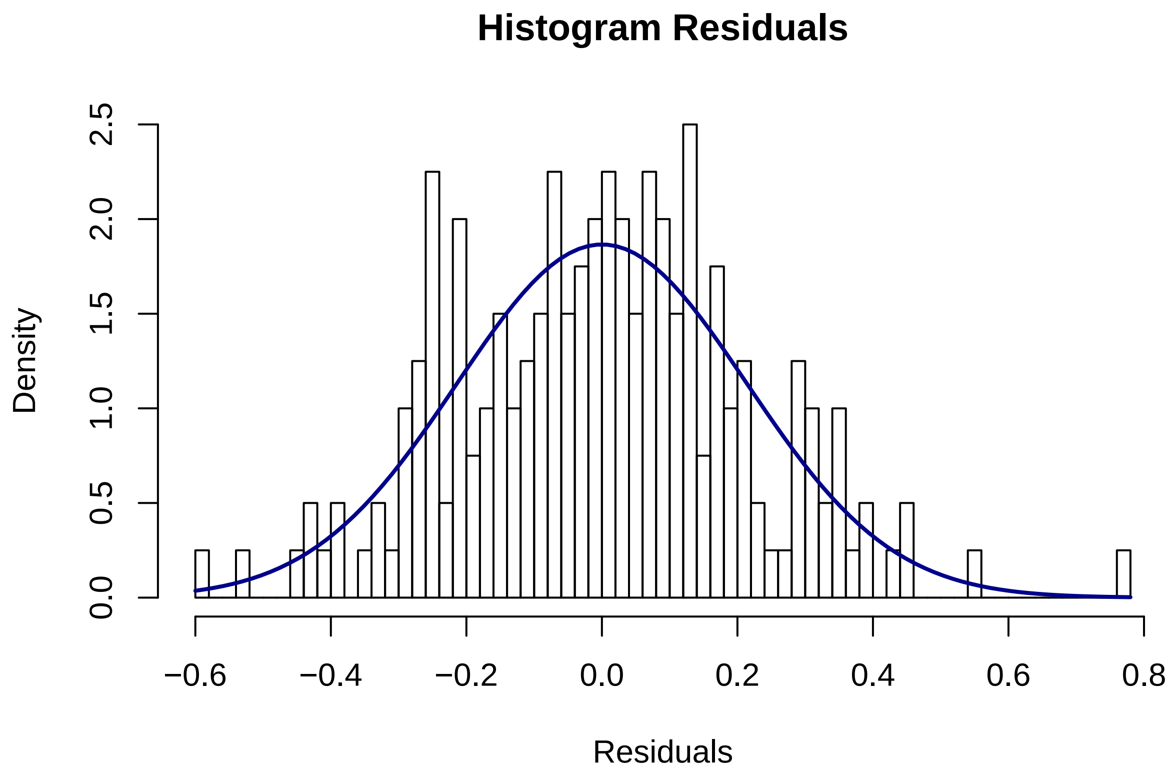
2 Ttterations of rewieghting seems appropriate

Model Diagnostics for final model.

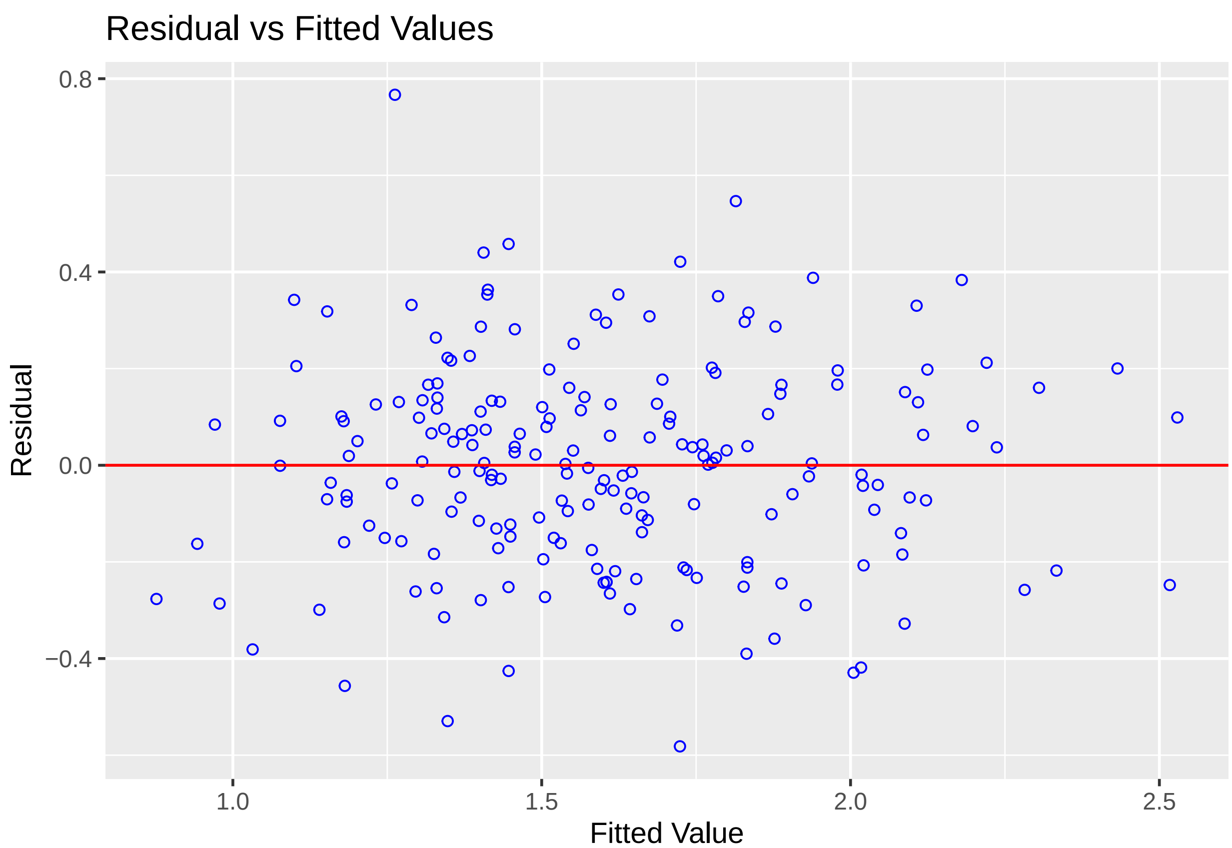
```r
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.fin)
```




```r
hist(fit.fin$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.fin$residuals), sd=sd(fit.fin$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```r
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.fin)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.fin)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.00738277, Df = 1, p = 0.93153
```

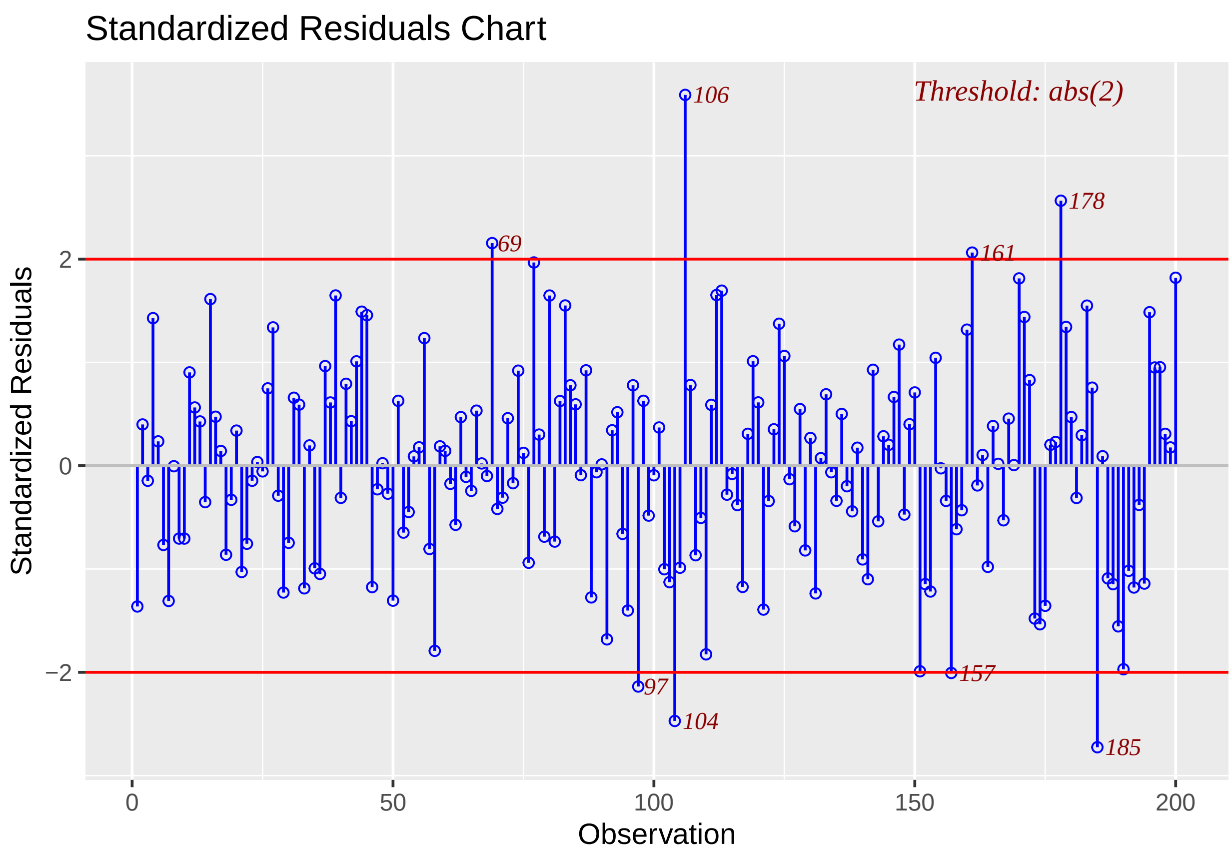
```
#Normality of errors tests
ols_test_normality(fit.fin)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9952         0.7809
## Kolmogorov-Smirnov        0.0297         0.9946
## Cramer-von Mises         42.4756         0.0000
## Anderson-Darling          0.2129         0.8518
## -----------------------------------------------
```

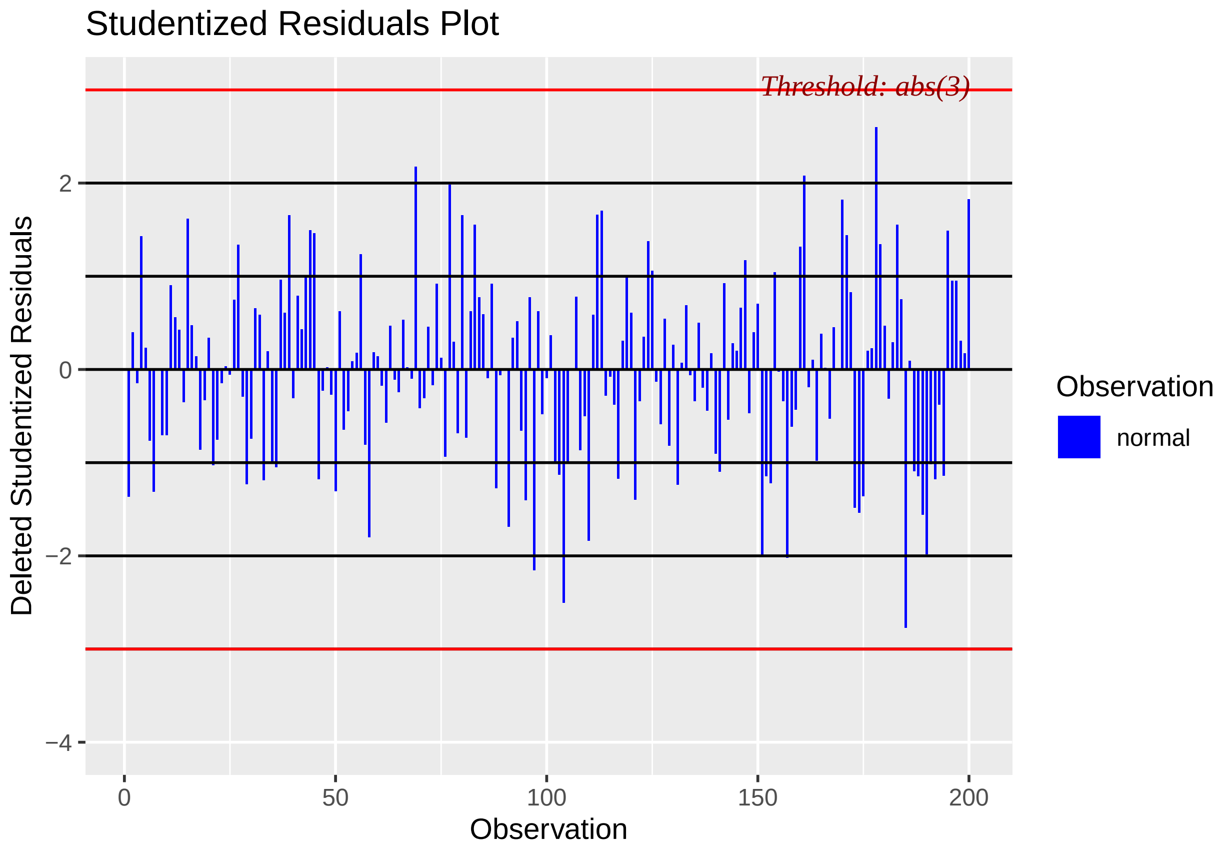
```
ols_test_correlation(fit.fin)
```

```
## [1] 0.9968848
```

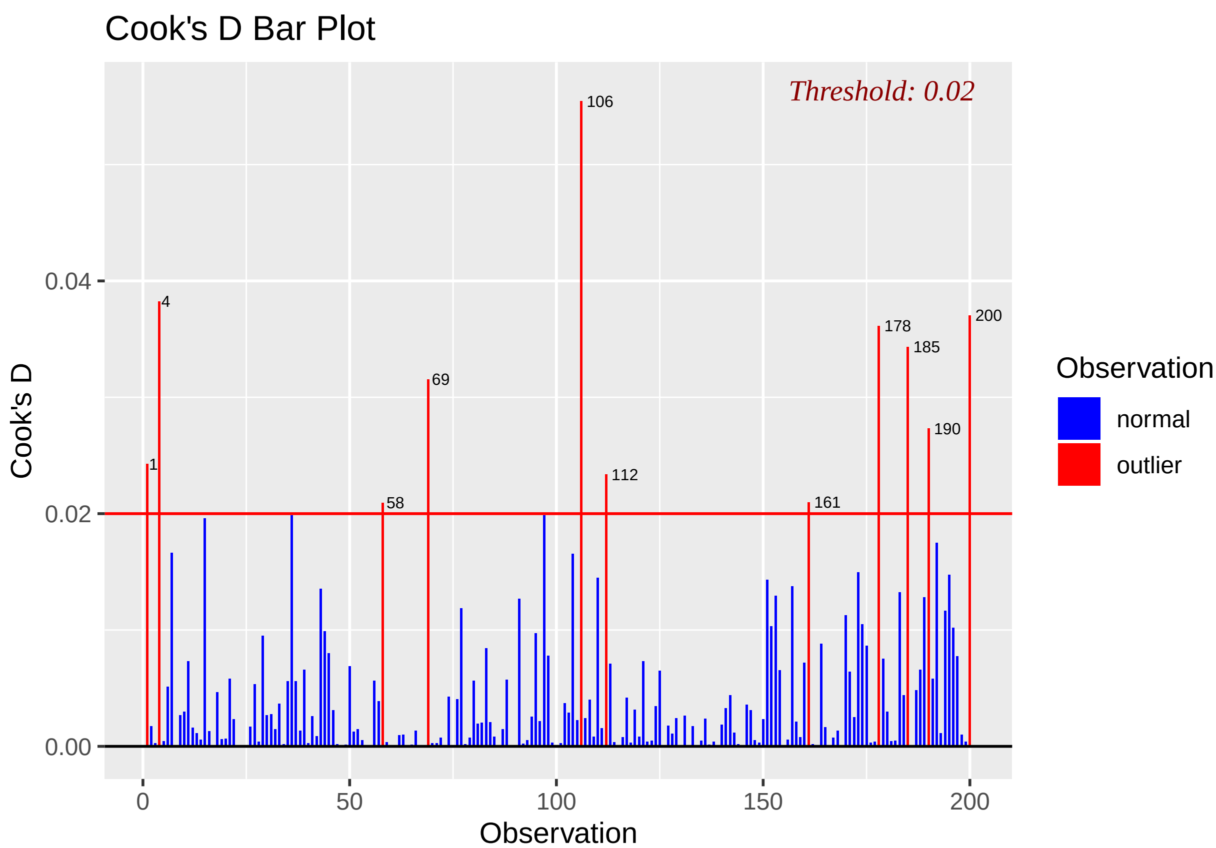
```
mod=fit.fin
#Standardized Residual plot
ols_plot_resid_stand(mod)
```



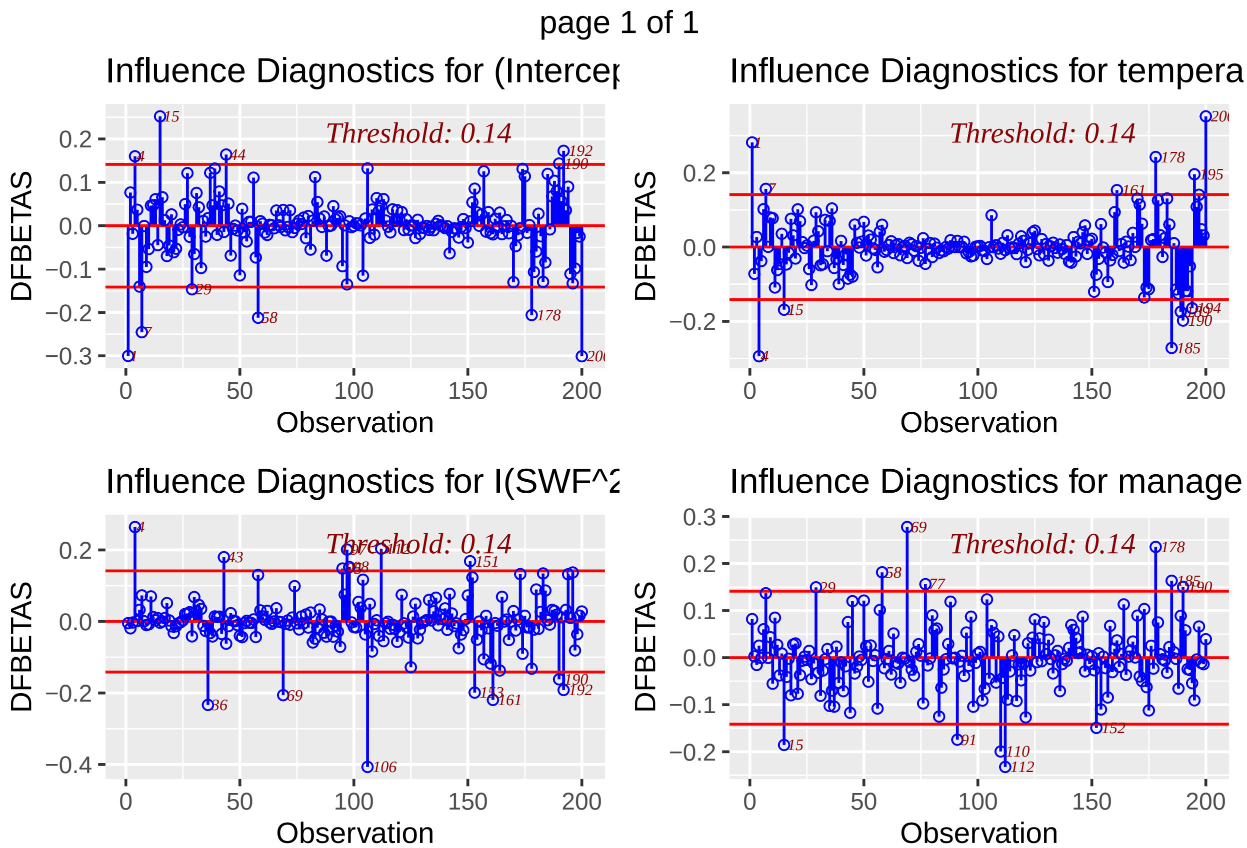

```r
#studentized residual plot
ols_plot_resid_stud(mod)
```



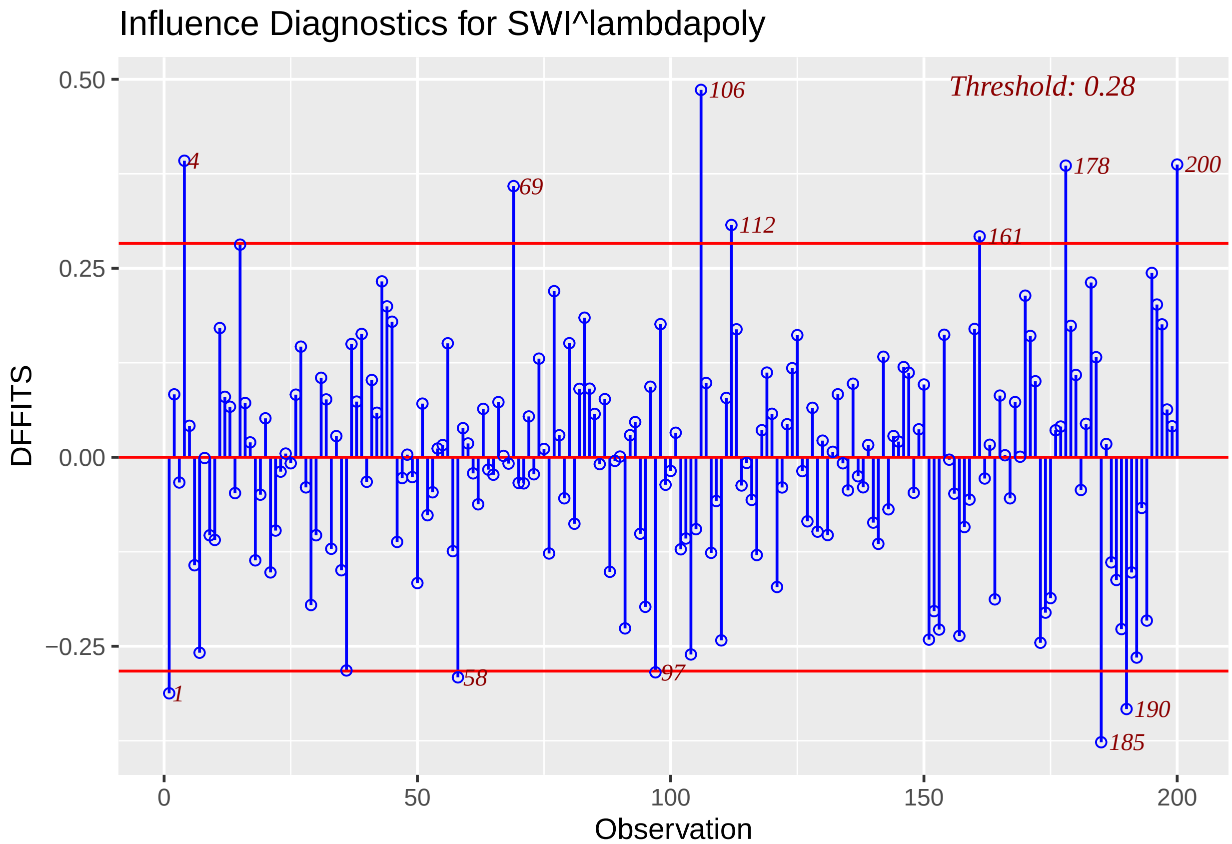

```
#cooks distance plot
ols_plot_cooksd_bar(mod)
```



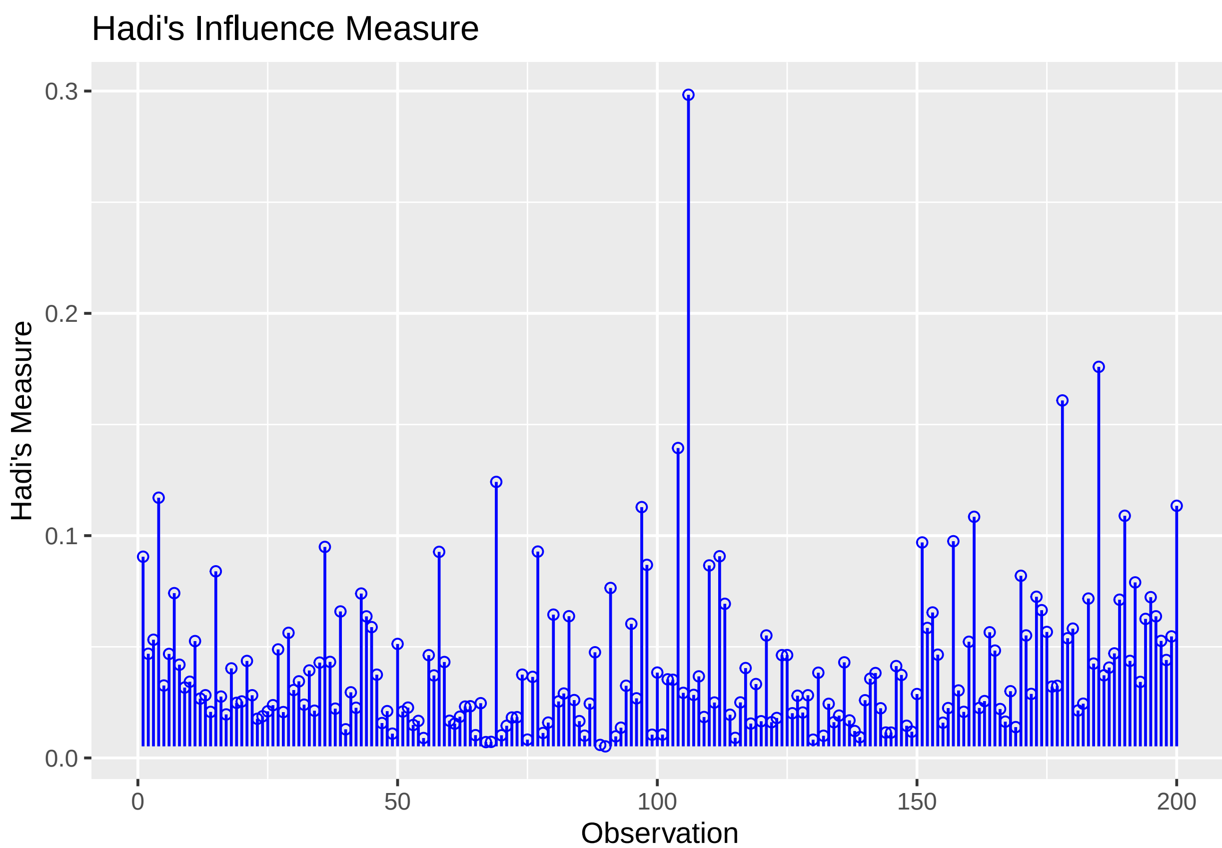

```
#DFBetas for each variable
ols_plot_dfbetas(mod)
```




```r
#Difference in fit chart for each sample
ols_plot_dffits(mod)
```

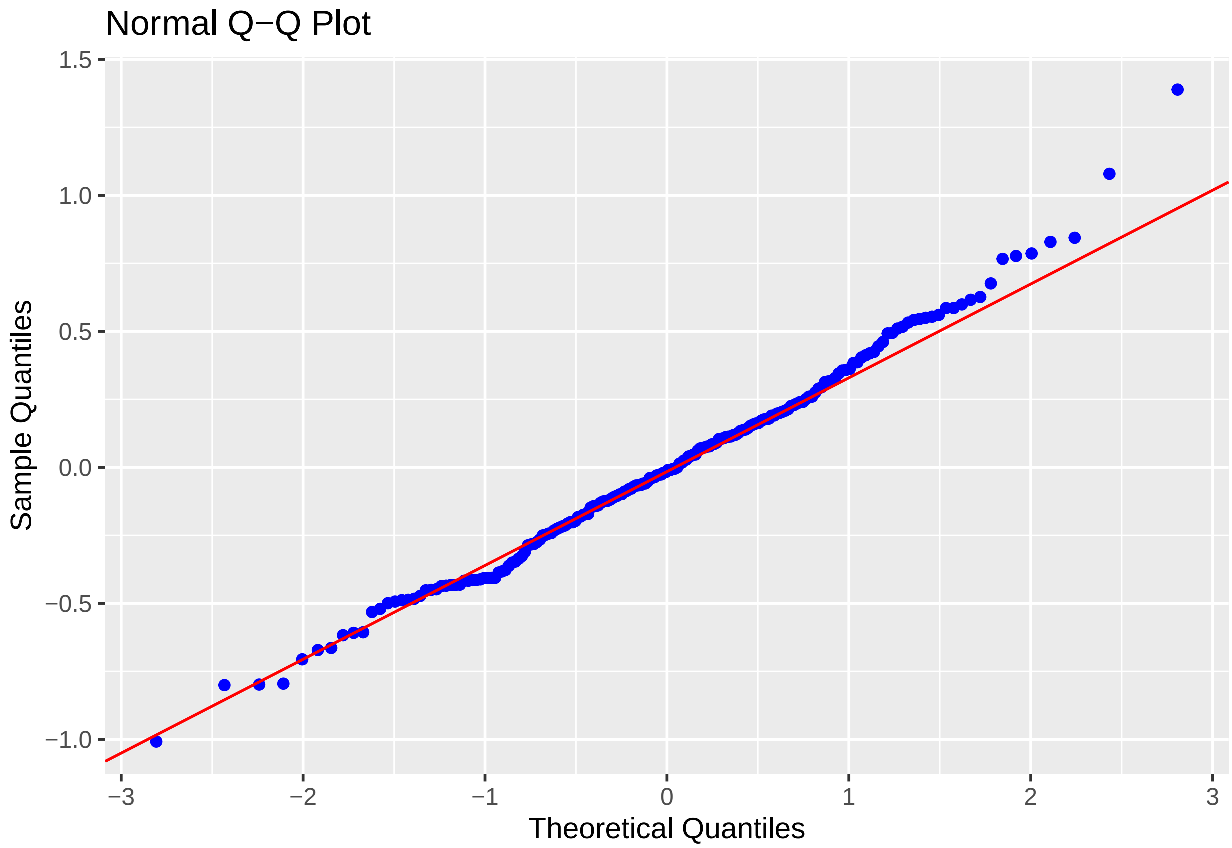



```r
#Plot for observation influence using hadi's distance
ols_plot_hadi(mod)
```

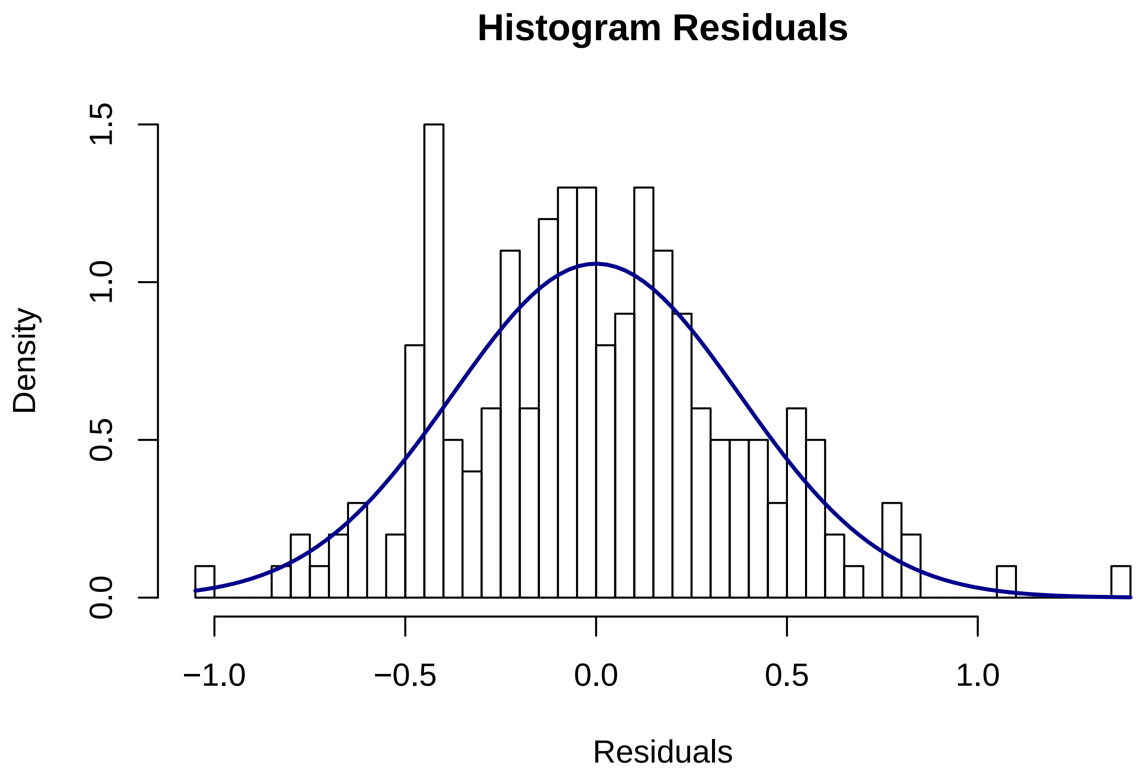


Alternative model diagnostics

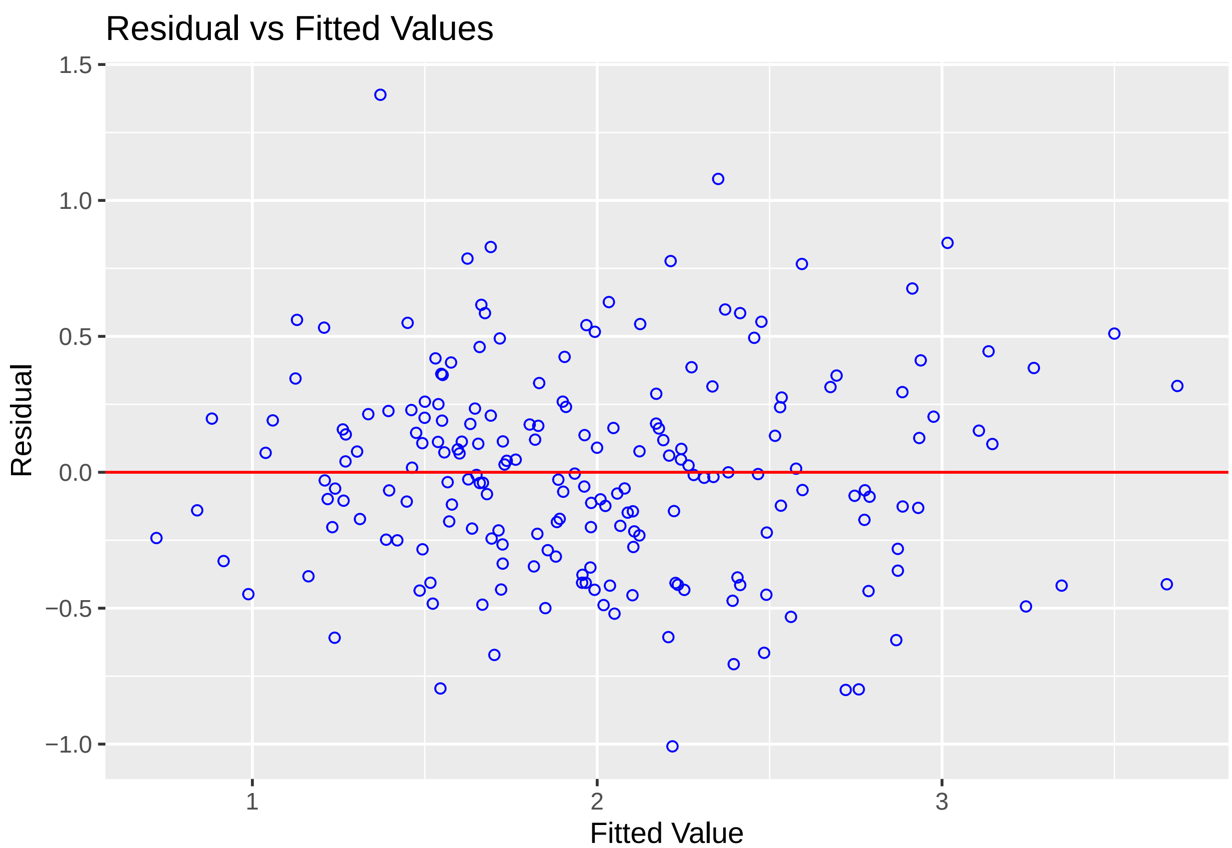
```
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(fit.alt)
```




```
hist(fit.alt$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(fit.alt$residuals), sd=sd(fit.alt$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(fit.alt)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(fit.alt)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.5727483, Df = 1, p = 0.44917
```

```
#Normality of errors tests
ols_test_normality(fit.alt)
```

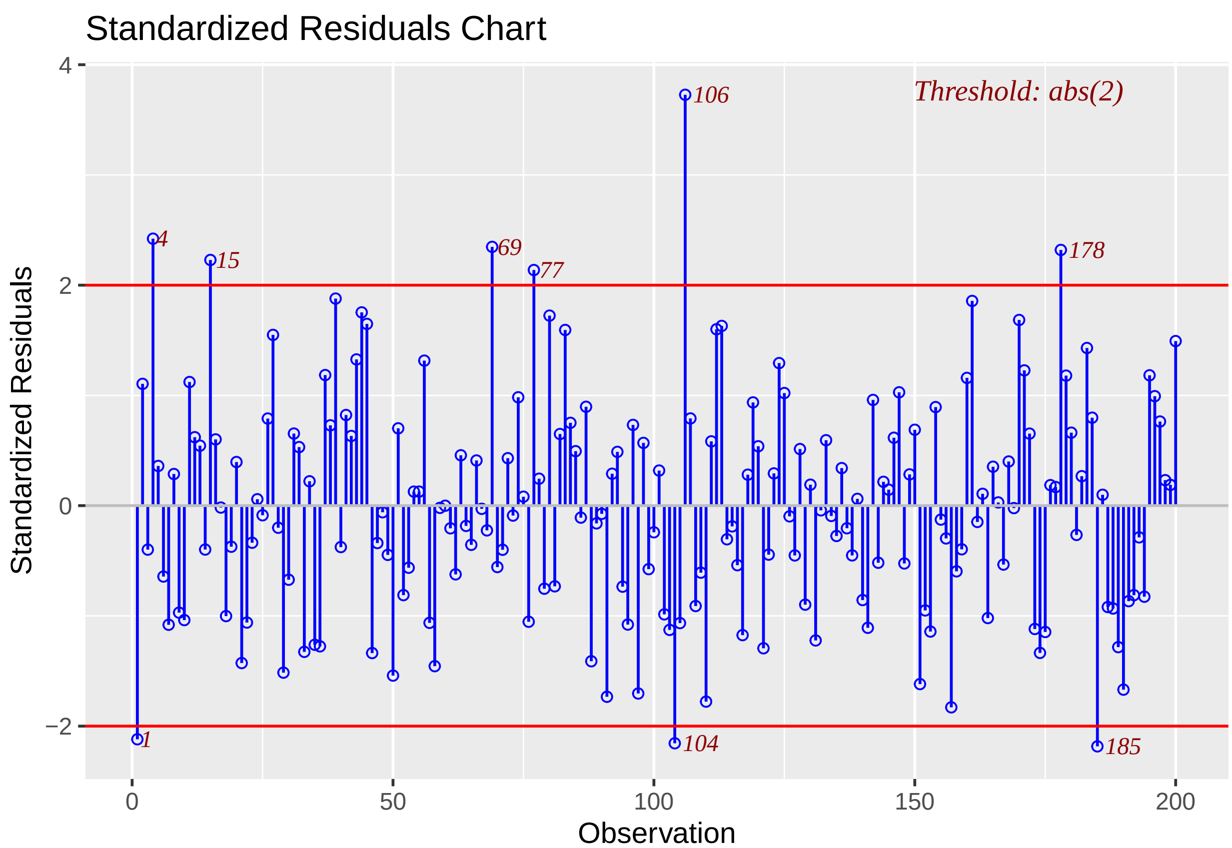
```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9912         0.2689
## Kolmogorov-Smirnov        0.0365         0.9530
## Cramer-von Mises         30.6541         0.0000
## Anderson-Darling          0.3552         0.4570
## -----------------------------------------------
```

```
ols_test_correlation(fit.alt)
```

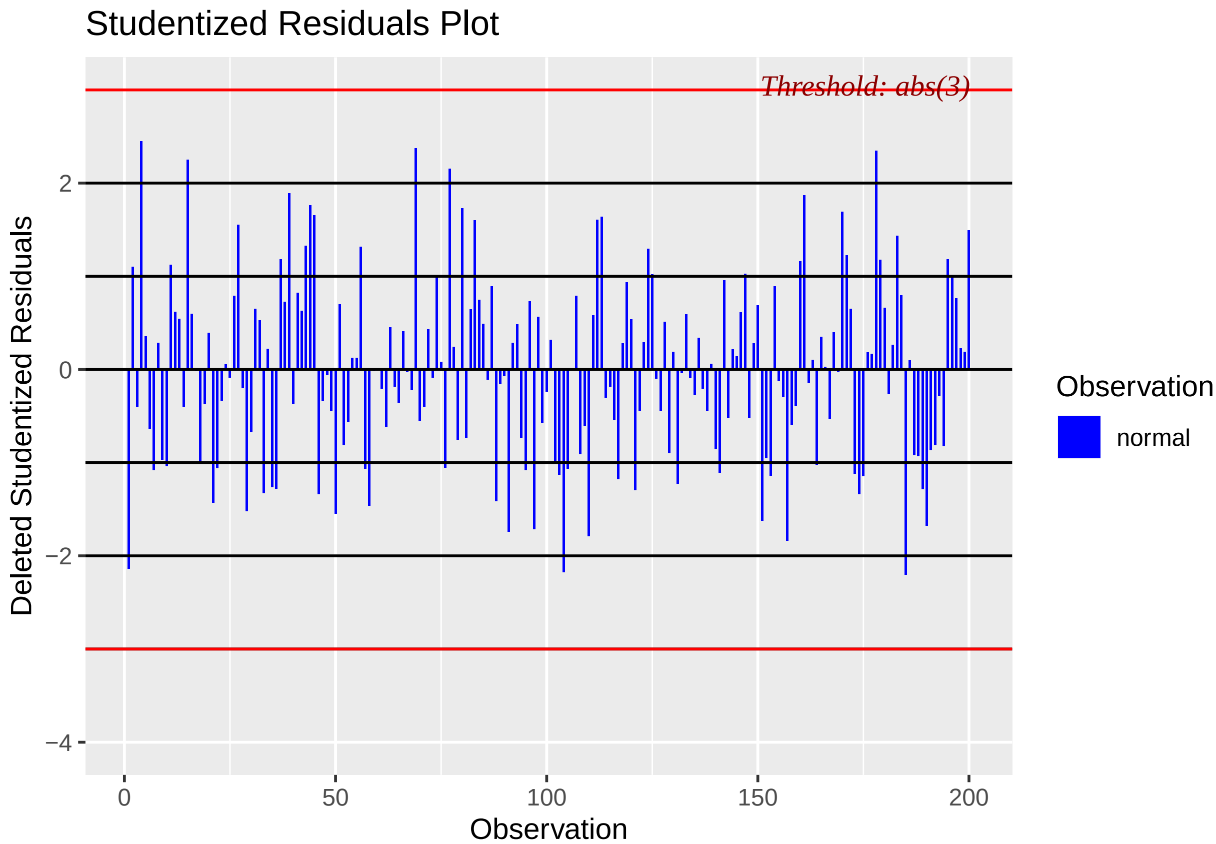
```
## Warning in cor(h, out): the standard deviation is zero
```

```
## [1] NA
```

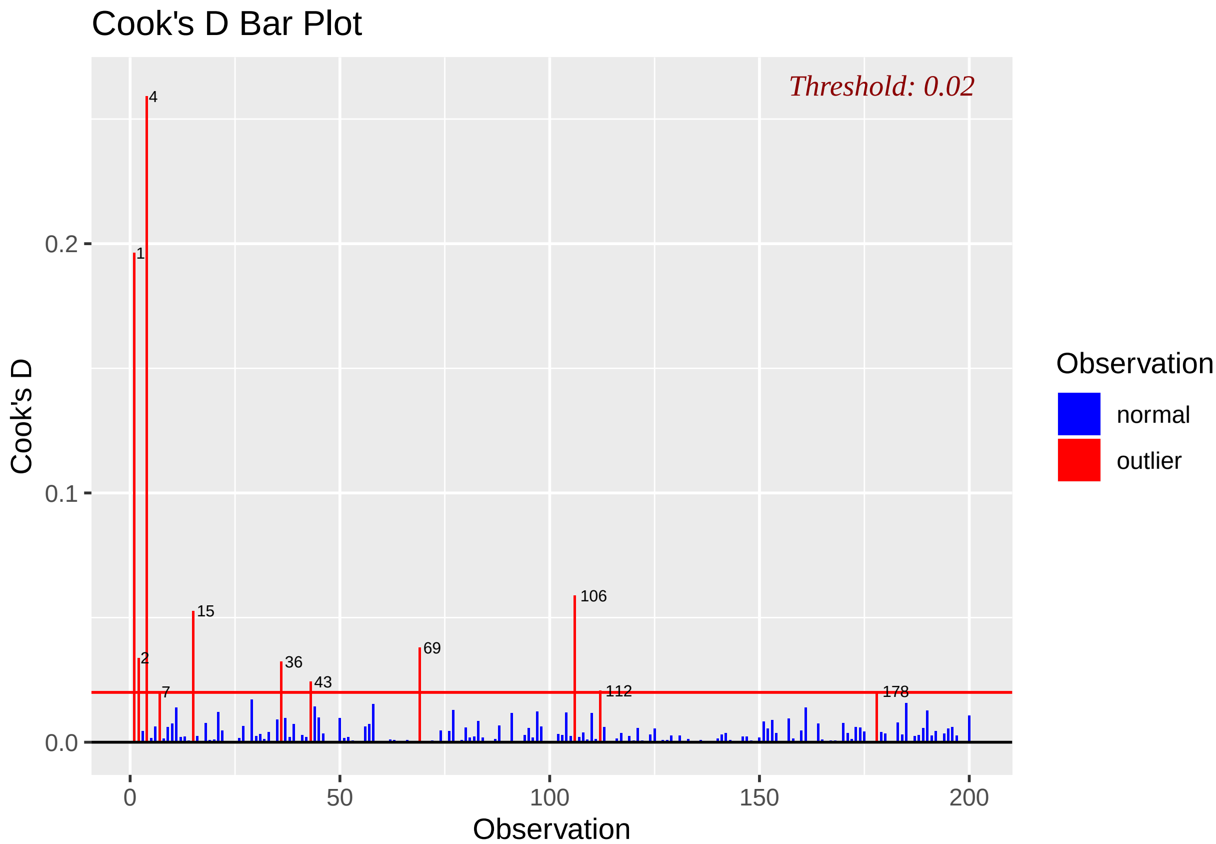
```
mod=fit.alt
#Standardized Residual plot
ols_plot_resid_stand(mod)
```



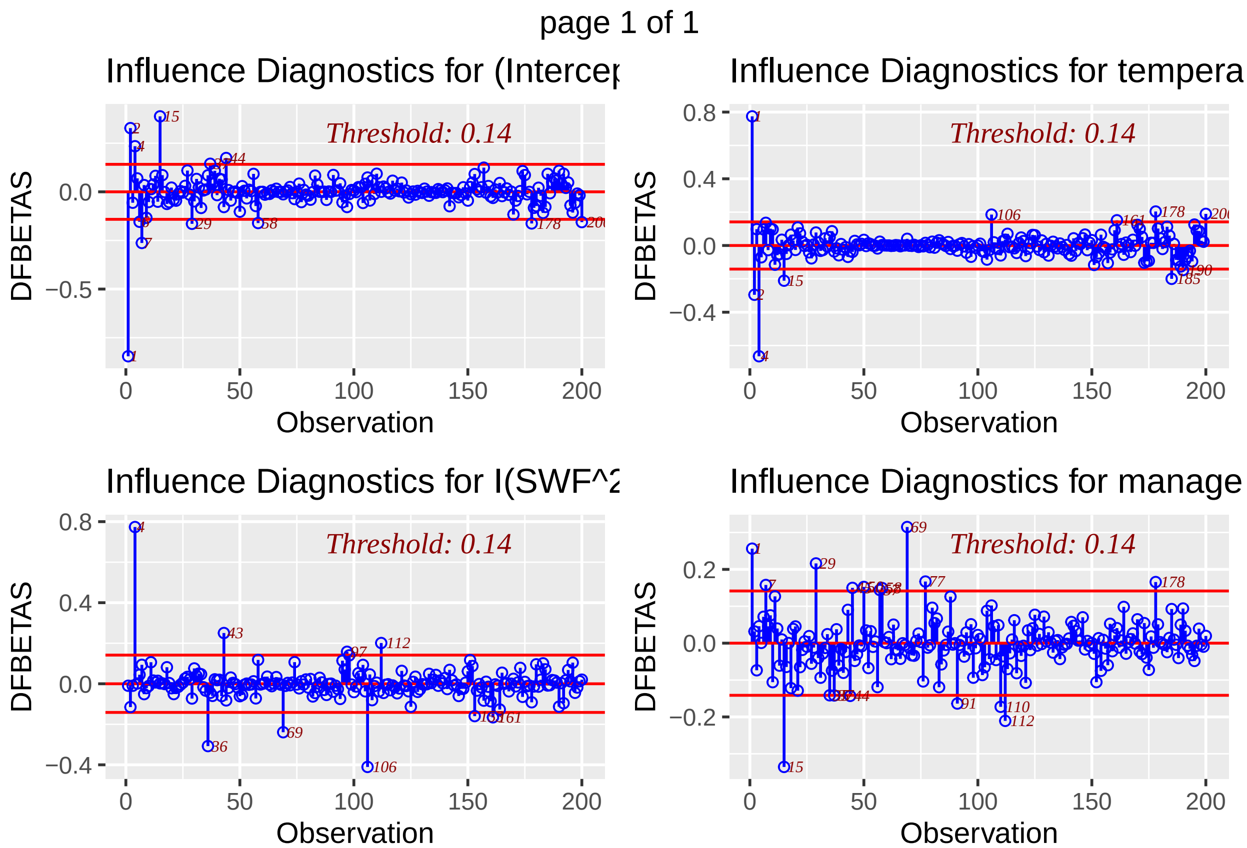

```r
#studentized residual plot
ols_plot_resid_stud(mod)
```



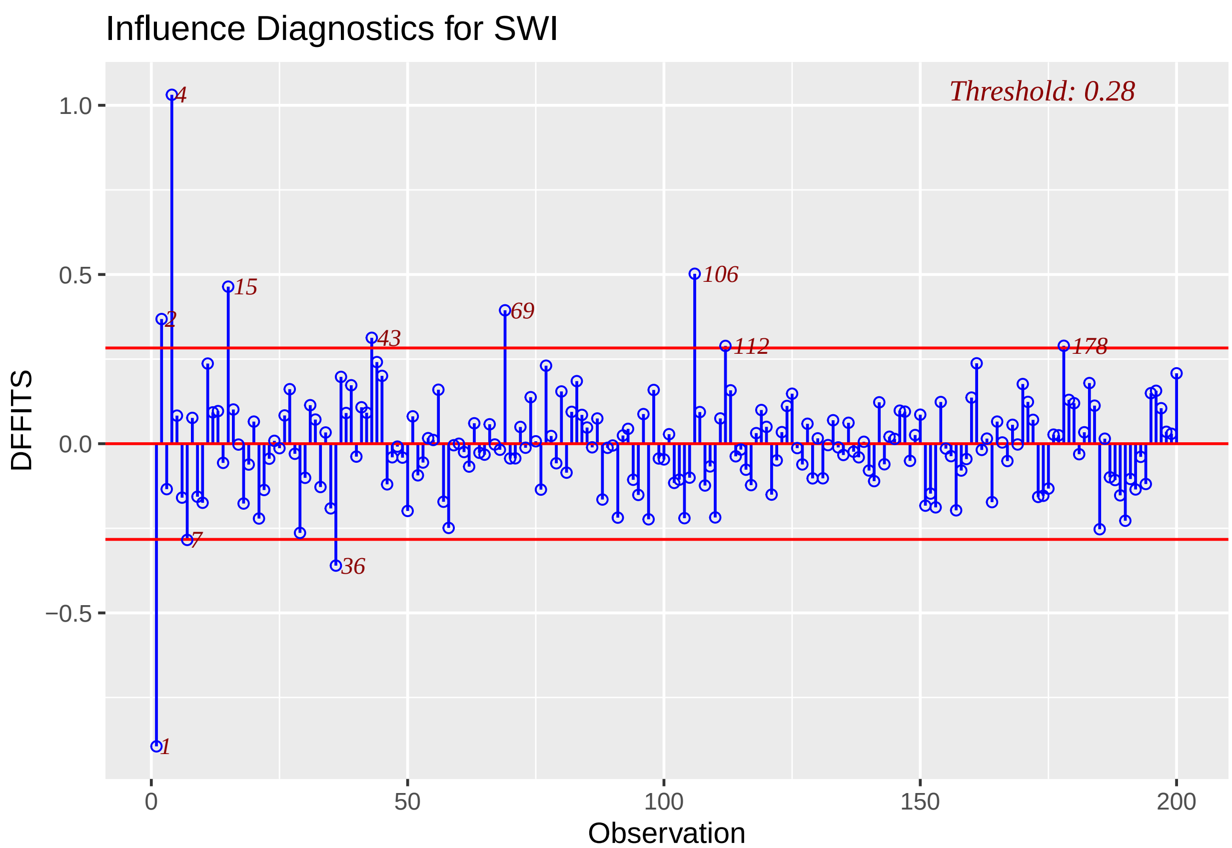

```
#cooks distance plot
ols_plot_cooksd_bar(mod)
```



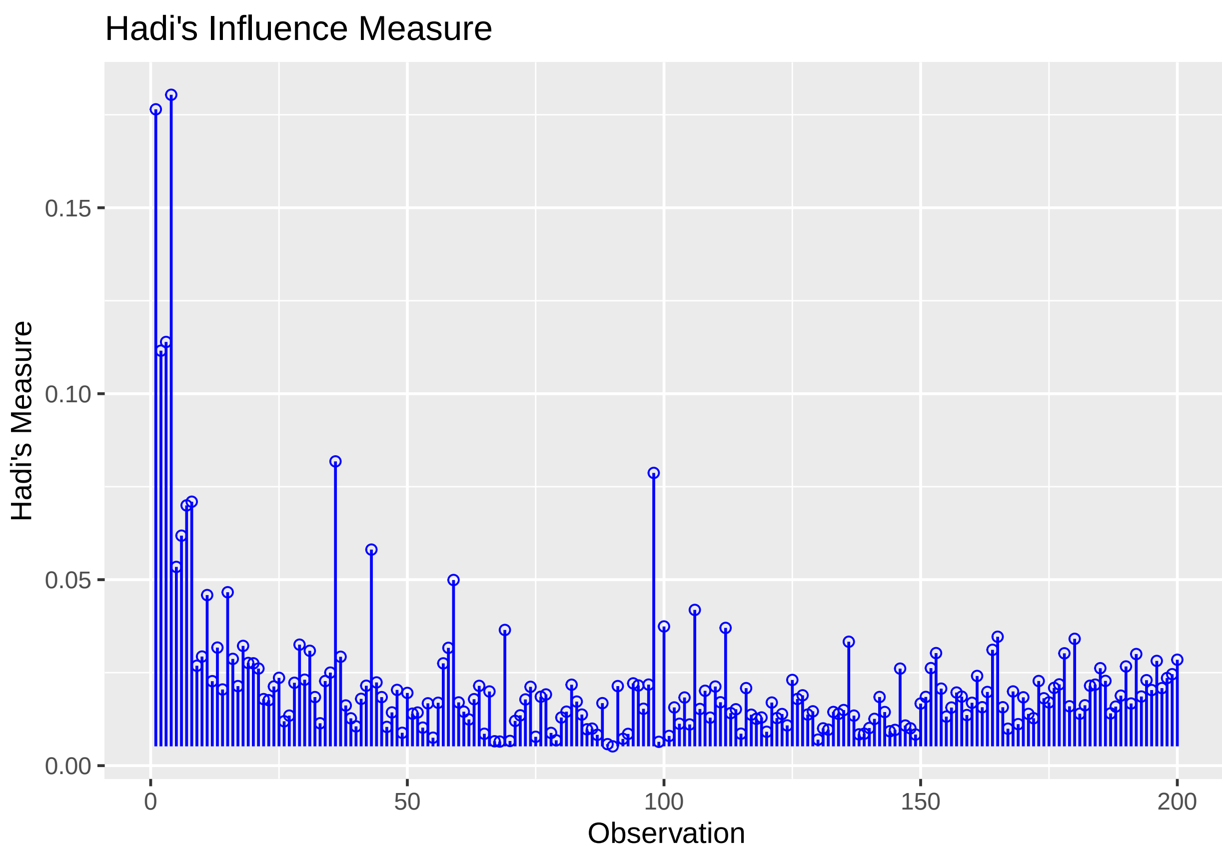

```
#DFBetas for each variable
ols_plot_dfbetas(mod)
```




```r
#Difference in fit chart for each sample
ols_plot_dffits(mod)
```




```r
#Plot for observation influence using hadi's distance
ols_plot_hadi(mod)
```



All assumptions met for both models. Outliers are still influencing the models but remain unredacted as per assignment instructions.

RMSE of models against the test data.

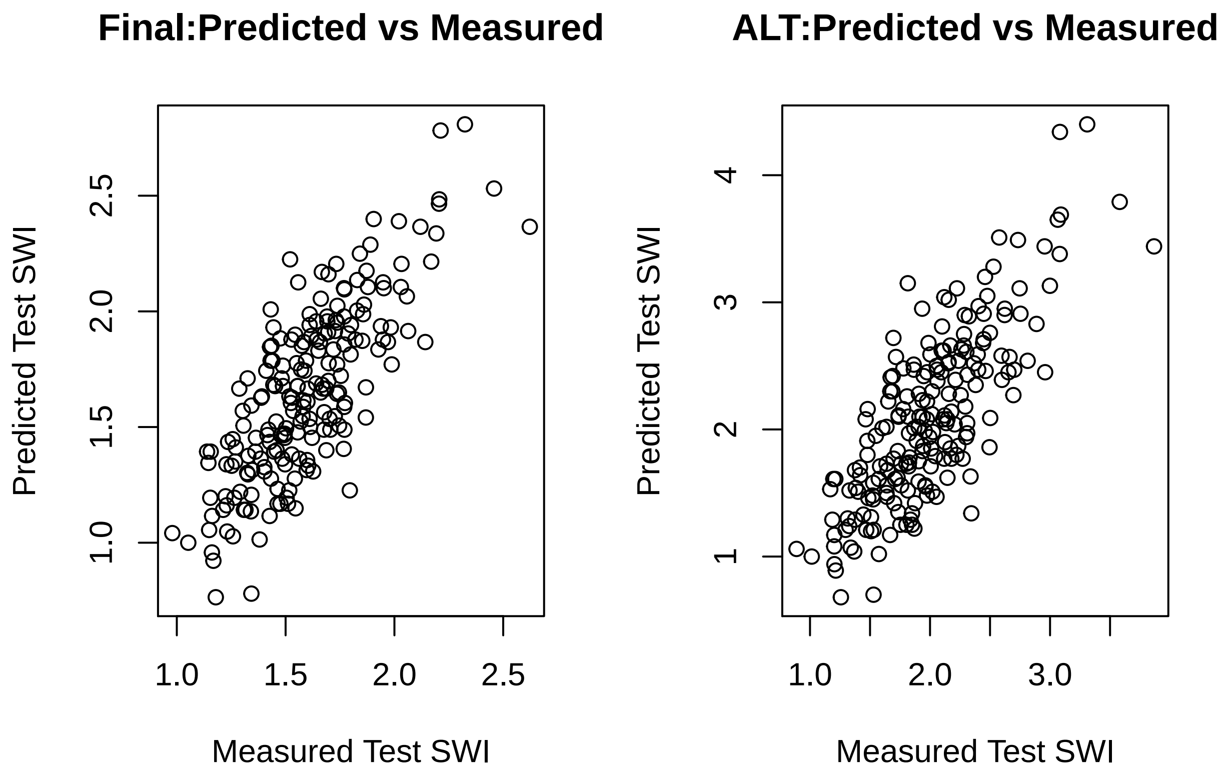
```
#RMSE of final and alternative models when test data is applied
results.fin=predict(fit.fin,data.test)
results.alt=predict(fit.alt,data.test)
rmse(results.fin,data.test$SWI^lambdapoly)
```

```
## [1] 0.2426738
```

```
rmse(results.alt,data.test$SWI)
```

```
## [1] 0.434435
```

```
#Plots of final and alternative models predicting test data results
par(mfrow=(c(1,2)))
plot(results.fin,data.test$SWI^lambdapoly,
     xlab="Measured Test SWI",ylab="Predicted Test SWI",main="Final:Predicted vs Measured")
plot(results.alt,data.test$SWI,
     xlab="Measured Test SWI",ylab="Predicted Test SWI",main="ALT:Predicted vs Measured")
```



Final model with box-cox transform is out performing the WLS model.

Fitting the test data to the models and evaluating the results.

```
fit.fin.test=lm(formula = SWI^lambdapoly ~ I(SWF^2) + temperature + management,
           data = data.test)
fit.alt.test=lm(formula = SWI ~ I(SWF^2) + temperature + management,
           data = data.test, weights = w1)
summary(fit.fin.test)
```

```
##
## Call:
## lm(formula = SWI^lambdapoly ~ I(SWF^2) + temperature + management,
##     data = data.test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67351 -0.17479 -0.00278  0.17206  0.66883
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.495702   0.080482   6.159 4.07e-09 ***
## I(SWF^2)    0.170526   0.011368  15.000  < 2e-16 ***
## temperature 0.030994   0.003425   9.048  < 2e-16 ***
## management  0.036173   0.006678   5.417 1.77e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2368 on 196 degrees of freedom
## Multiple R-squared:  0.6104, Adjusted R-squared:  0.6045
## F-statistic: 102.4 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
summary(fit.alt.test)
```

```
##
## Call:
## lm(formula = SWI ~ I(SWF^2) + temperature + management, data = data.test,
##     weights = w1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8662 -1.0027 -0.1475  0.9436  4.1657
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.05709    0.14166  -0.403    0.687
## I(SWF^2)     0.30982    0.02146  14.440  < 2e-16 ***
## temperature  0.05863    0.00599   9.788  < 2e-16 ***
## management   0.07078    0.01192   5.940 1.28e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.514 on 196 degrees of freedom
## Multiple R-squared:  0.6165, Adjusted R-squared:  0.6106
## F-statistic:   105 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
#comparing to base model to examine potential overfitting
fit.basic.test=lm(formula = SWI ~ SWF + temperature + management + size,
           data = data.test)


#investigating if the generalization problem resides
#within the choice of training/test data.
summary(fit.basic)
```

```
##
## Call:
## lm(formula = SWI ~ . - duration, data = data.training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02005 -0.27531 -0.02608  0.24343  1.53643
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.357059   0.147639  -2.418   0.0165 *
## SWF          0.834942   0.059745  13.975  < 2e-16 ***
## temperature  0.048518   0.005275   9.198  < 2e-16 ***
## size        -0.001927   0.001576  -1.222   0.2231
## management   0.063350   0.011409   5.553 9.13e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3961 on 195 degrees of freedom
## Multiple R-squared:  0.6694, Adjusted R-squared:  0.6626
## F-statistic:  98.7 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
summary(fit.basic.test)
```

```
##
## Call:
## lm(formula = SWI ~ SWF + temperature + management + size, data = data.test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23253 -0.28300 -0.00478  0.28897  1.29020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.638053   0.179246  -3.560 0.000466 ***
## SWF          0.919424   0.063734  14.426  < 2e-16 ***
## temperature  0.054235   0.006331   8.567  3.2e-15 ***
## management   0.068697   0.012296   5.587  7.7e-08 ***
## size         0.001090   0.001561   0.698 0.485735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4356 on 195 degrees of freedom
## Multiple R-squared:  0.5974, Adjusted R-squared:  0.5892
## F-statistic: 72.35 on 4 and 195 DF,  p-value: < 2.2e-16
```

```r
#generalization problem appears to be a result of the test data set
#as perfromance is worse even without any transforms or reweighting
```

Merging the training and test data together and assigning class labels. (this is only used for making one of the following plots easier to code)

```r
data.training$class=rep(2,200)
data.test$class=rep(3,200)
data.withclass=rbind(data.training,data.test)
```

two sided t-tests to compare distributions of each variable across training and test data

```r
t.test(data.training$SWI,data.test$SWI,"two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  data.training$SWI and data.test$SWI
## t = -1.4506, df = 398, p-value = 0.1477
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.23258612  0.03508612
## sample estimates:
## mean of x mean of y
##   1.99375   2.09250
```

```r
t.test(data.training$SWF,data.test$SWF,"two.sided")
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  data.training$SWF and data.test$SWF
## t = 0.011292, df = 397.96, p-value = 0.991
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09520335  0.09630335
## sample estimates:
## mean of x mean of y
##   1.49750   1.49695
```

```
t.test(data.training$temperature,data.test$temperature,"two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  data.training$temperature and data.test$temperature
## t = 0.0077209, df = 395.14, p-value = 0.9938
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.014527  1.022527
## sample estimates:
## mean of x mean of y
##   18.9495   18.9455
```

```
t.test(data.training$management,data.test$management,"two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  data.training$management and data.test$management
## t = 0, df = 398, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4943628  0.4943628
## sample estimates:
## mean of x mean of y
##     4.115     4.115
```

```
t.test(data.training$size,data.test$size,"two.sided")
```

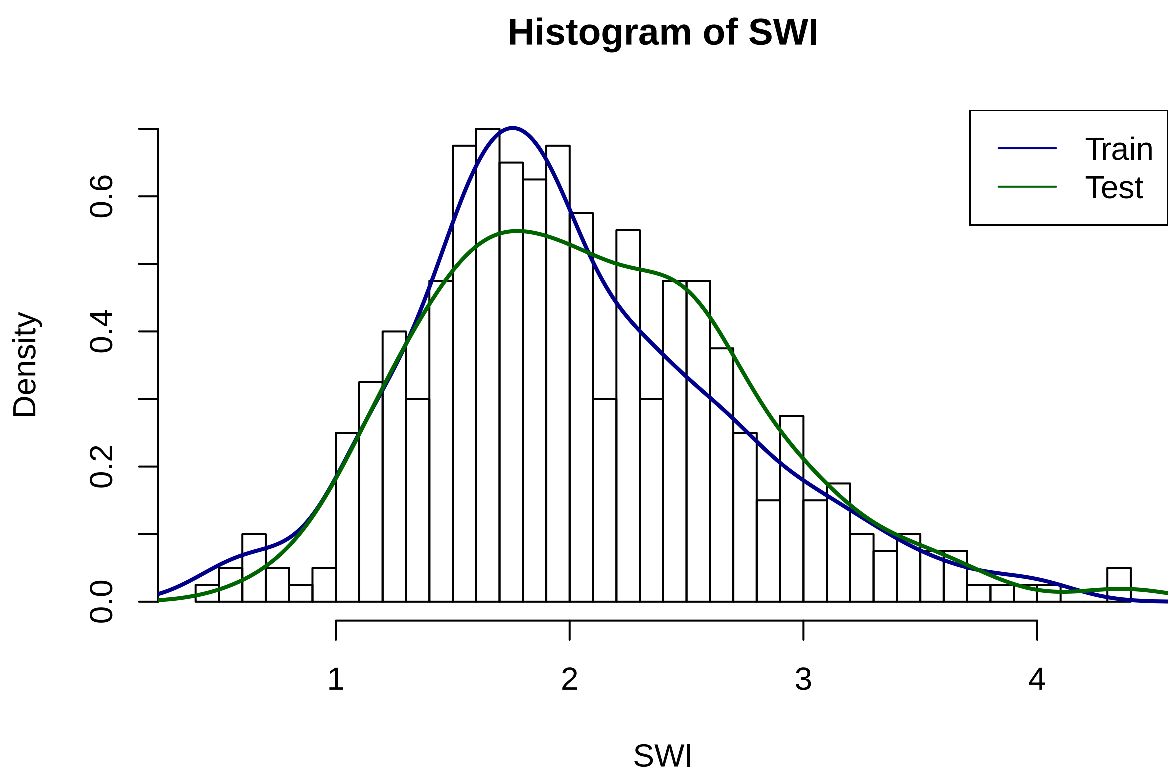
```
##
##  Welch Two Sample t-test
##
## data:  data.training$size and data.test$size
## t = 0.49369, df = 393.78, p-value = 0.6218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.791402  4.663402
## sample estimates:
## mean of x mean of y
##   41.3155   40.3795
```

```
#SWI shows a loose relation between the two data sets
#relative to the tests performed on other variables.

#This implies that either the training or test data set
#may have an unusually high number of outliers.
```

P-value of t-test for SWI is only .14 compared to .9+ for each other variable excluding size.

Plot of Histogram of all SWI values and the density plots of the training and test data

```
hist(data.withclass$SWI,breaks=50,prob=T,main="Histogram of SWI",xlab="SWI")
lines(density(data.training$SWI), col="darkblue", lwd=2, yaxt="n")
lines(density(data.test$SWI), col="darkgreen", lwd=2, yaxt="n")
legend("topright",legend=c("Train","Test"),col=c("darkblue","darkgreen"),lty=c(1,1))
```



This plot shows quite the substantial difference in the two distributions leading to poor generalization with model building.

Final and alternative model 95% regression coeficients interval

```
confint(fit.fin)
```

```
##                  2.5 %     97.5 %
## (Intercept) 0.42291995 0.66918447
## I(SWF^2)    0.13520836 0.17681183
## temperature 0.02141378 0.03269399
## management  0.02482665 0.04926958
```

```
confint(fit.fin.test)
```

```
##                  2.5 %     97.5 %
## (Intercept) 0.33697952 0.65442507
## I(SWF^2)    0.14810604 0.19294625
## temperature 0.02423844 0.03774877
## management  0.02300331 0.04934291
```

```r
confint(fit.alt)
```

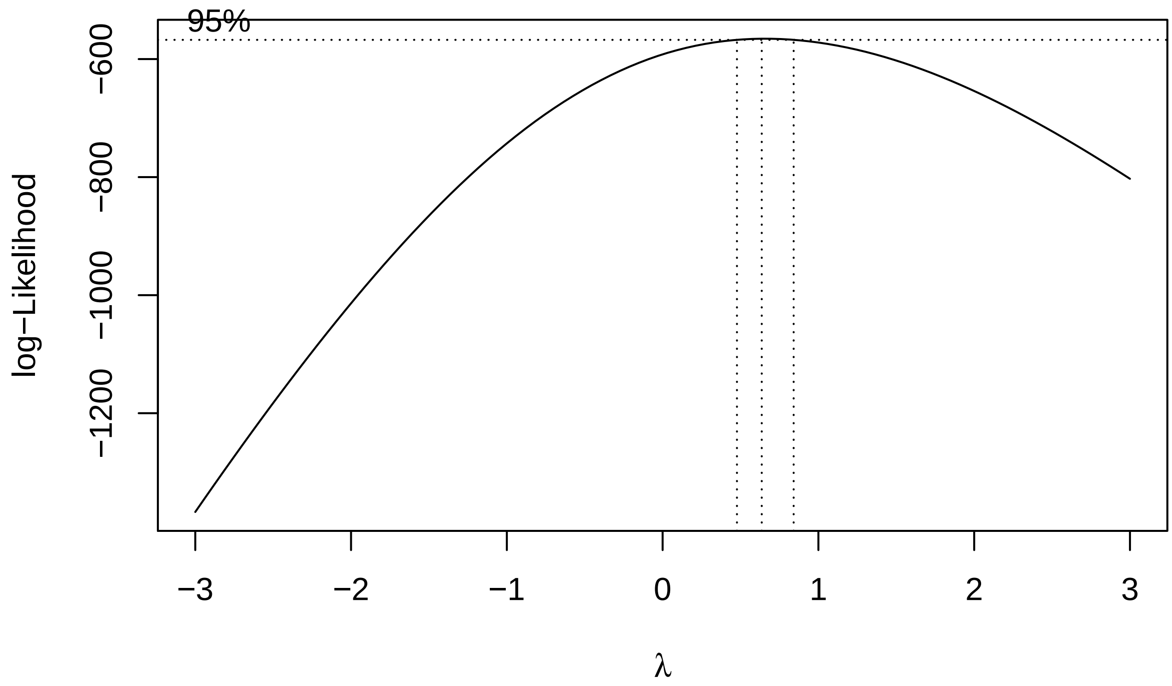
```
##                   2.5 %     97.5 %
## (Intercept) -0.03117542 0.31543985
## I(SWF^2)     0.25357277 0.32524800
## temperature  0.03806482 0.05523197
## management   0.04122228 0.08091336
```

```r
confint(fit.alt.test)
```

```
##                   2.5 %     97.5 %
## (Intercept) -0.33646650 0.22229391
## I(SWF^2)     0.26750405 0.35213102
## temperature  0.04682043 0.07044803
## management   0.04728440 0.09428369
```

Fitting ultimate model to full data set.

```r
fullmodel.ult=lm(SWI~I(SWF^2)+temperature+management,data=data)
bc.ult=boxcox(fullmodel.ult,lambda = seq(-3,3))
```




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

```
## [1] 0.4545455
```

```r
fit.bc.ult=lm(SWI^lambdault~I(SWF^2)+temperature+management,data=data)
summary(fit.bc.ult)
```

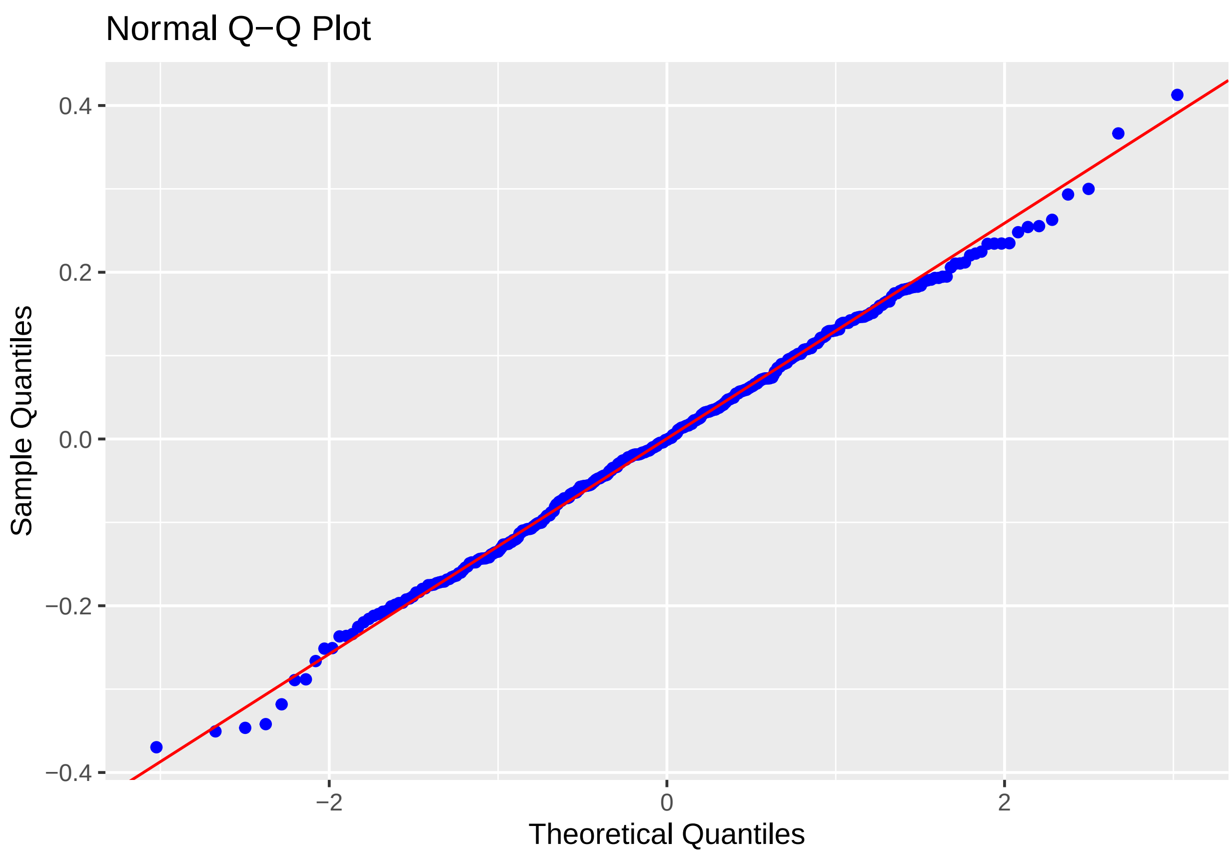
```
##
## Call:
## lm(formula = SWI^lambdault ~ I(SWF^2) + temperature + management,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36971 -0.08664 -0.00097  0.08763  0.41273
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.760397   0.027906  27.249  < 2e-16 ***
## I(SWF^2)    0.088873   0.004299  20.673  < 2e-16 ***
## temperature 0.015846   0.001232  12.867  < 2e-16 ***
## management  0.020289   0.002552   7.949 1.98e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1273 on 396 degrees of freedom
## Multiple R-squared:  0.6393, Adjusted R-squared:  0.6366
## F-statistic:   234 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
confint(fit.bc.ult)
```

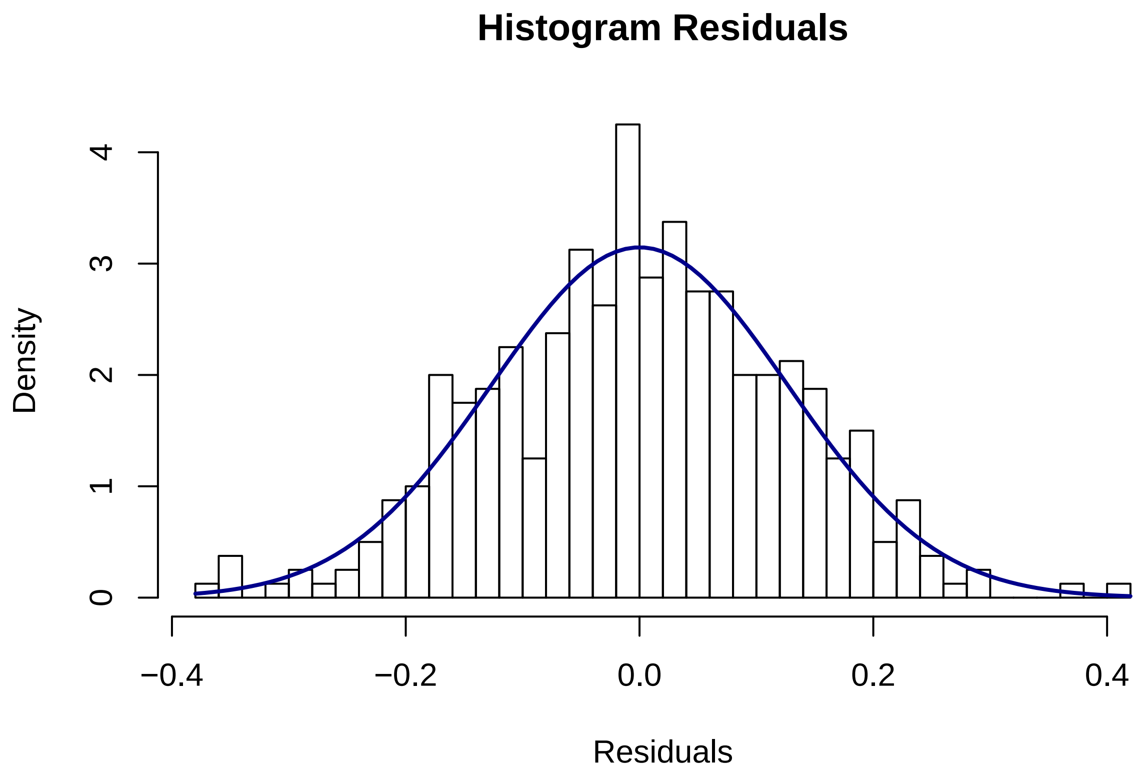
```
##                  2.5 %     97.5 %
## (Intercept) 0.70553500 0.81525939
## I(SWF^2)    0.08042145 0.09732497
## temperature 0.01342487 0.01826734
## management  0.01527115 0.02530666
```

Model diagnostics for ultimate model

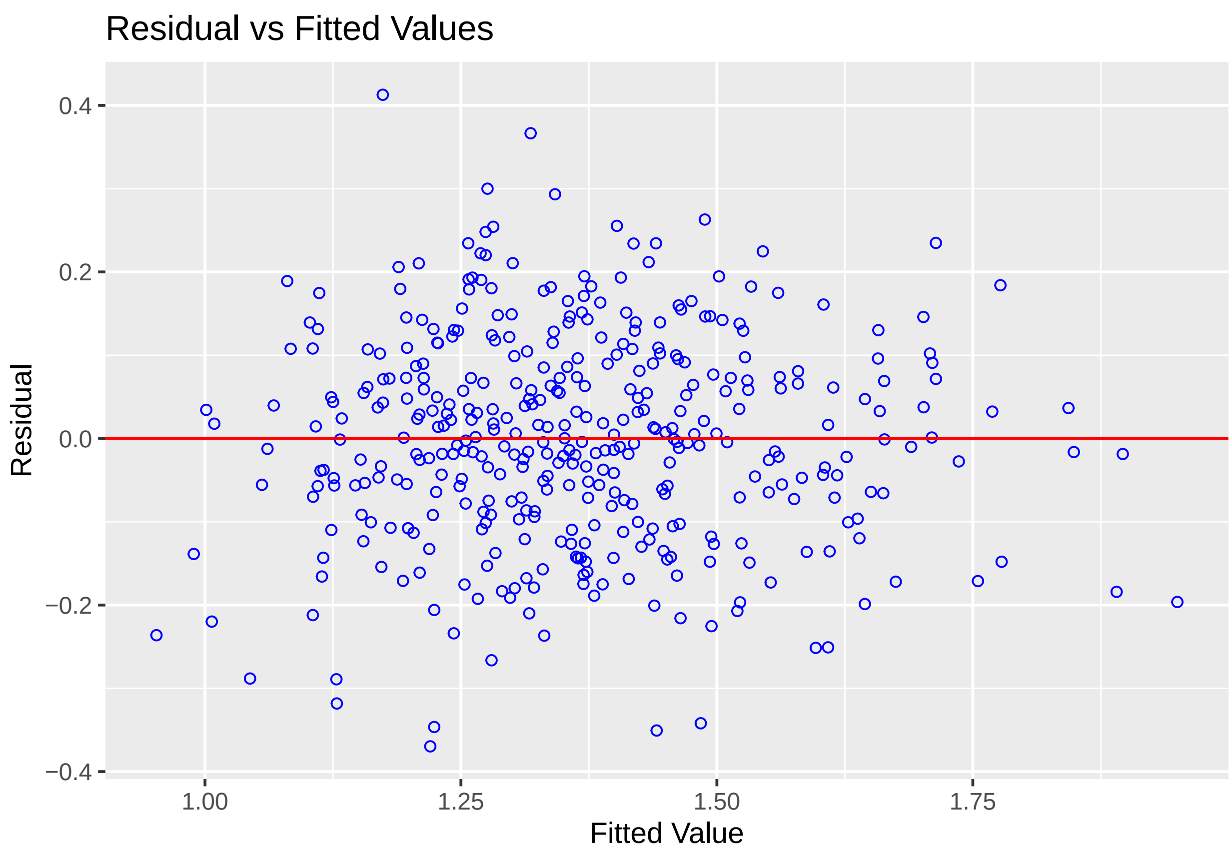
```
mod=fit.bc.ult
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(mod)
```




```
hist(mod$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(mod$residuals), sd=sd(mod$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(mod)
```




```r
#Breusch-Pagan Test for homoscedasticity
ncvTest(mod)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.258875, Df = 1, p = 0.26186
```

```r
#Normality of errors tests
ols_test_normality(mod)
```

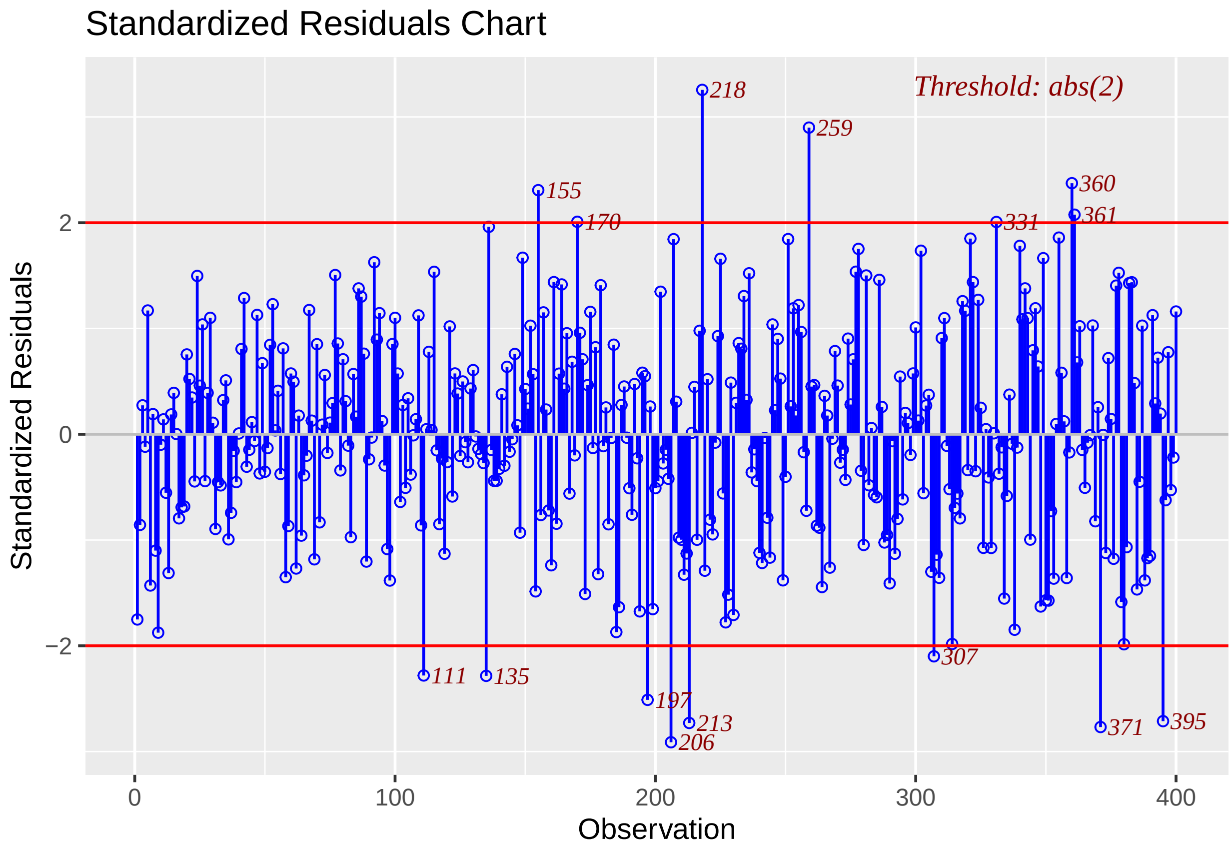
```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9976         0.8368
## Kolmogorov-Smirnov        0.0239         0.9767
## Cramer-von Mises        102.138          0.0000
## Anderson-Darling          0.2144         0.8492
## -----------------------------------------------
```

```r
ols_test_correlation(mod)
```

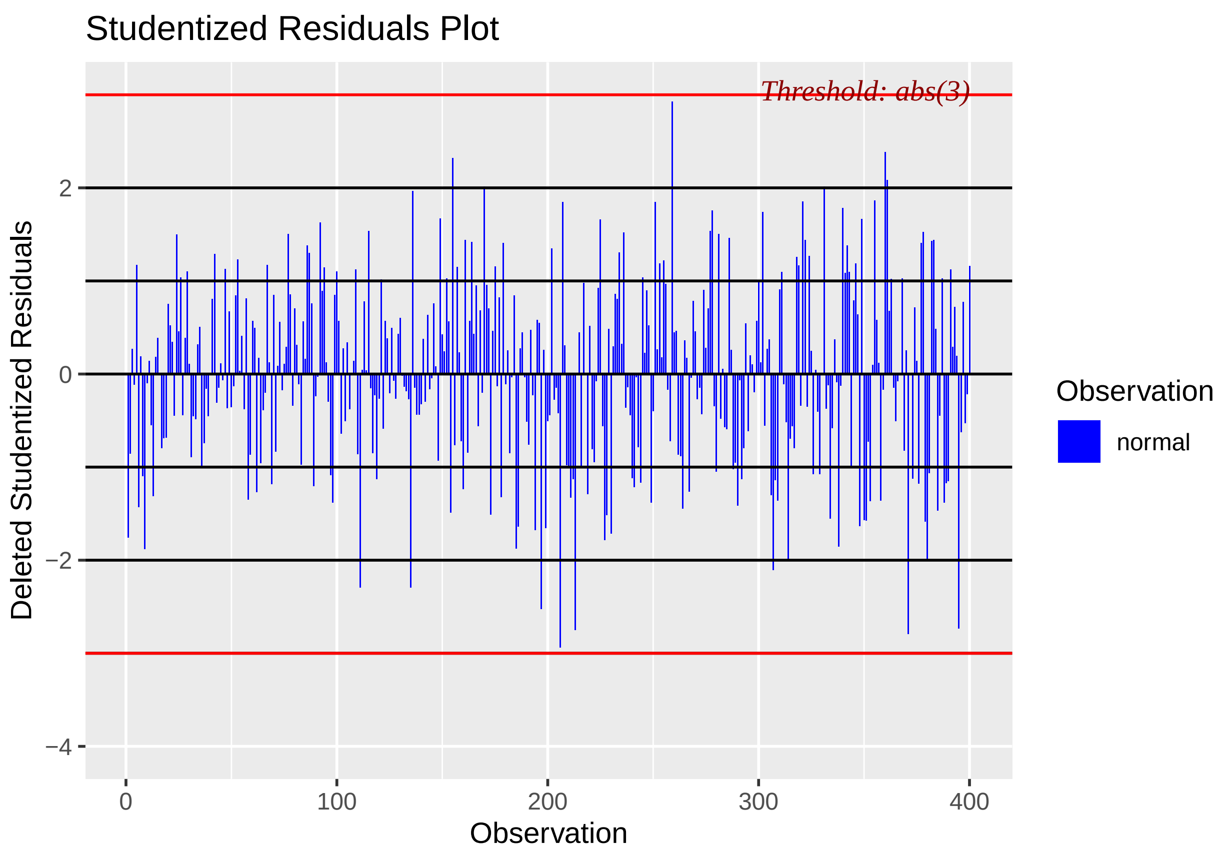
```
## [1] 0.9987477
```

More ultimate model diagnostics

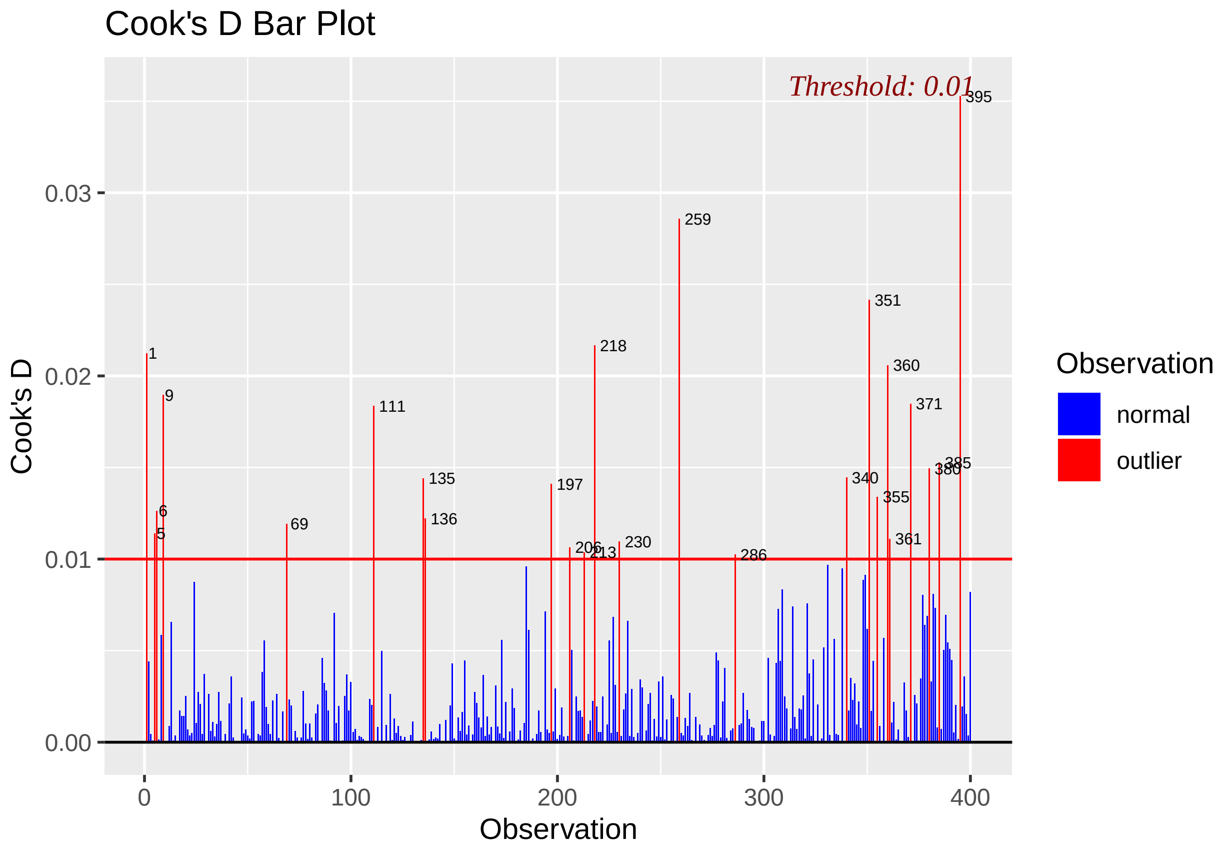
```r
#Standardized Residual plot
ols_plot_resid_stand(mod)
```



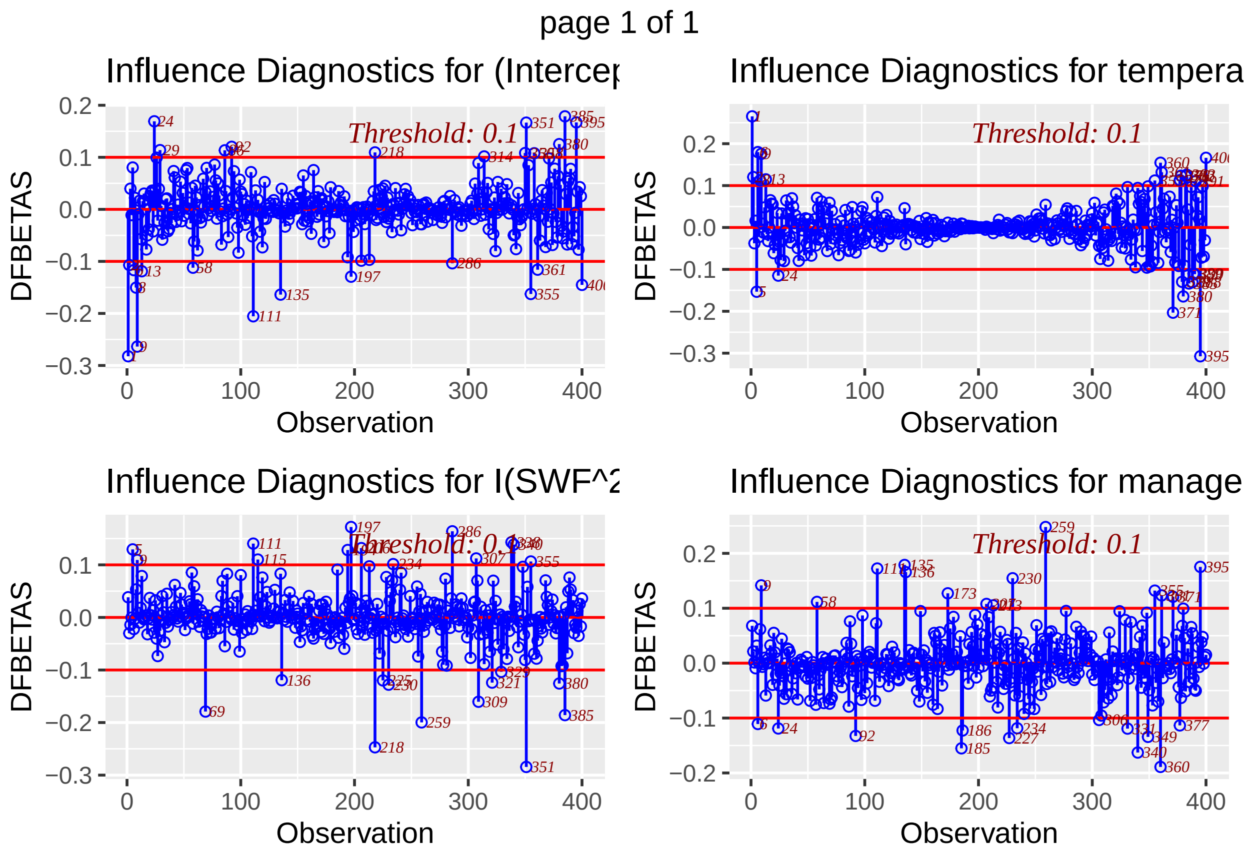

```
#studentized residual plot
ols_plot_resid_stud(mod)
```



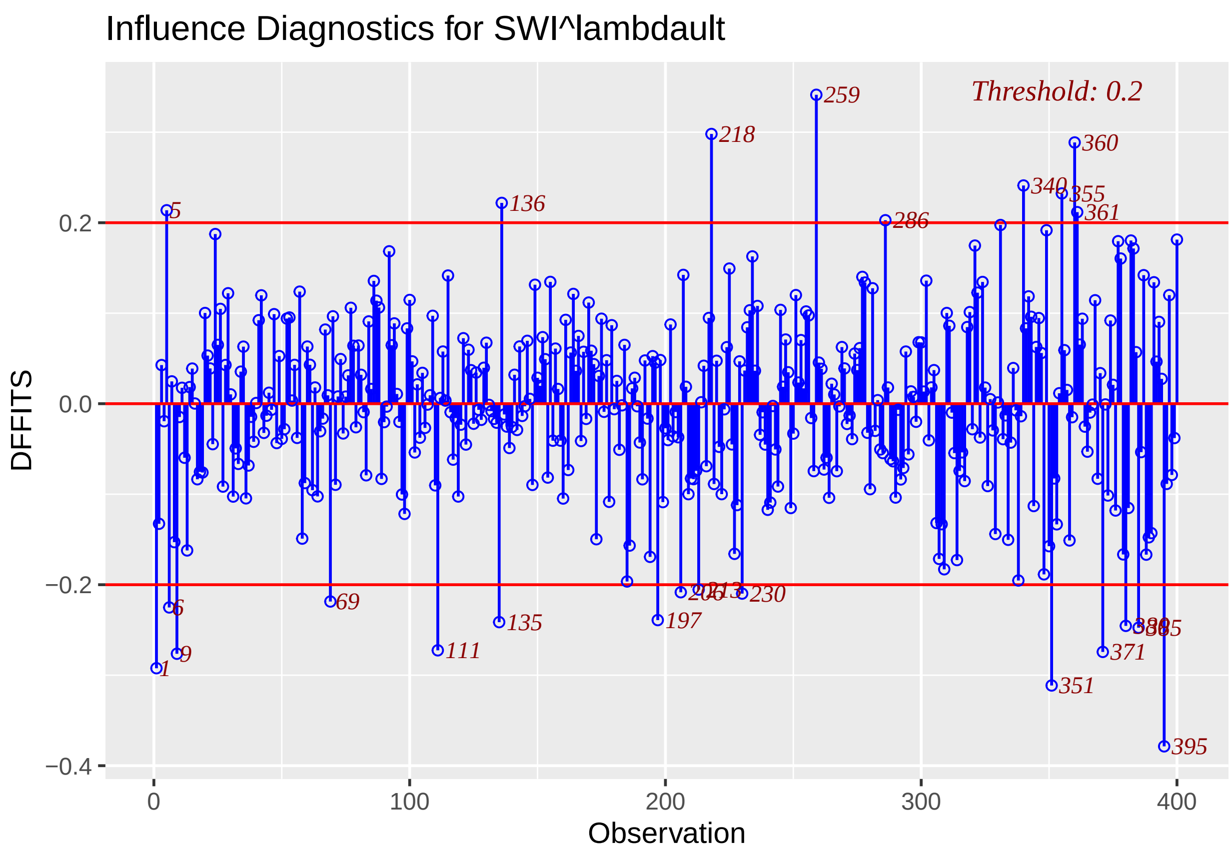

```
#cooks distance plot
ols_plot_cooksd_bar(mod)
```



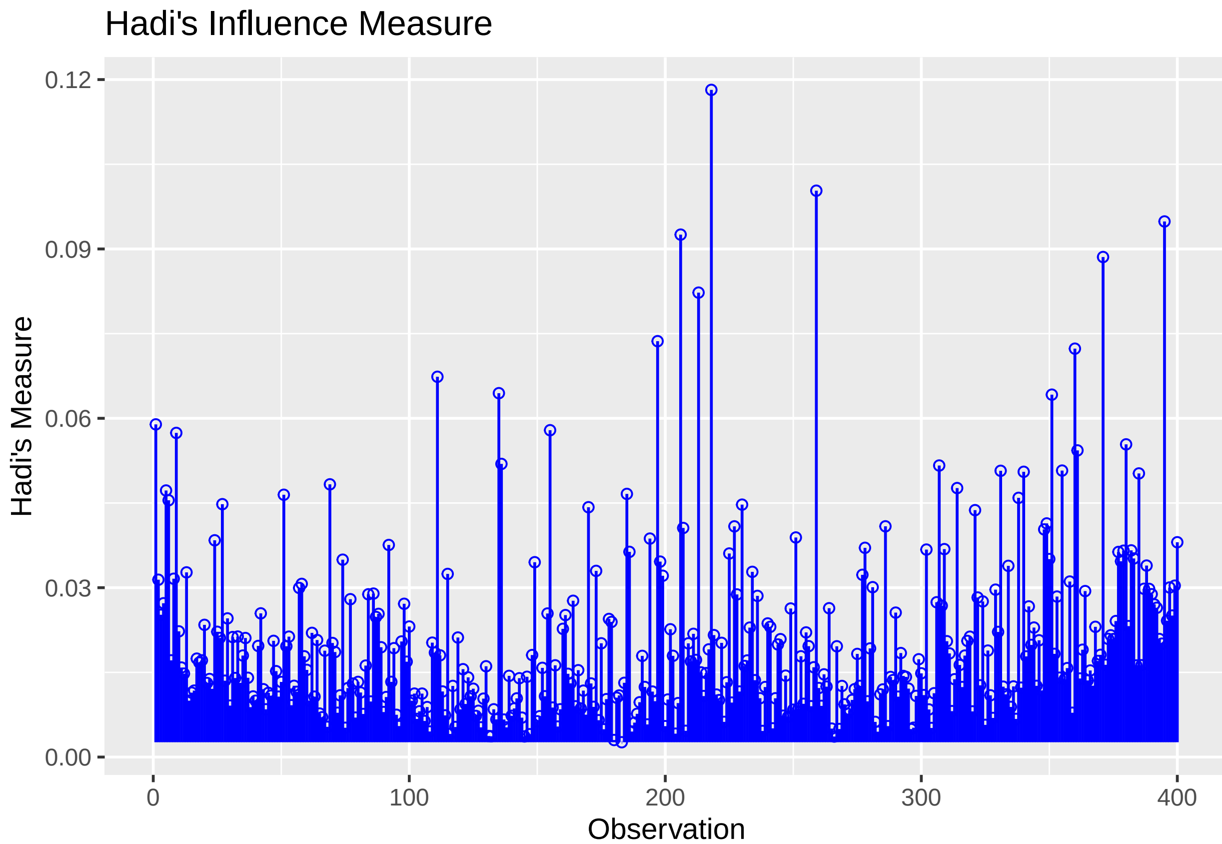

```r
#DFBetas for each variable
ols_plot_dfbetas(mod)
```




```r
#Difference in fit chart for each sample
ols_plot_dffits(mod)
```




```r
#Plot for observation influence using hadi's distance
ols_plot_hadi(mod)
```

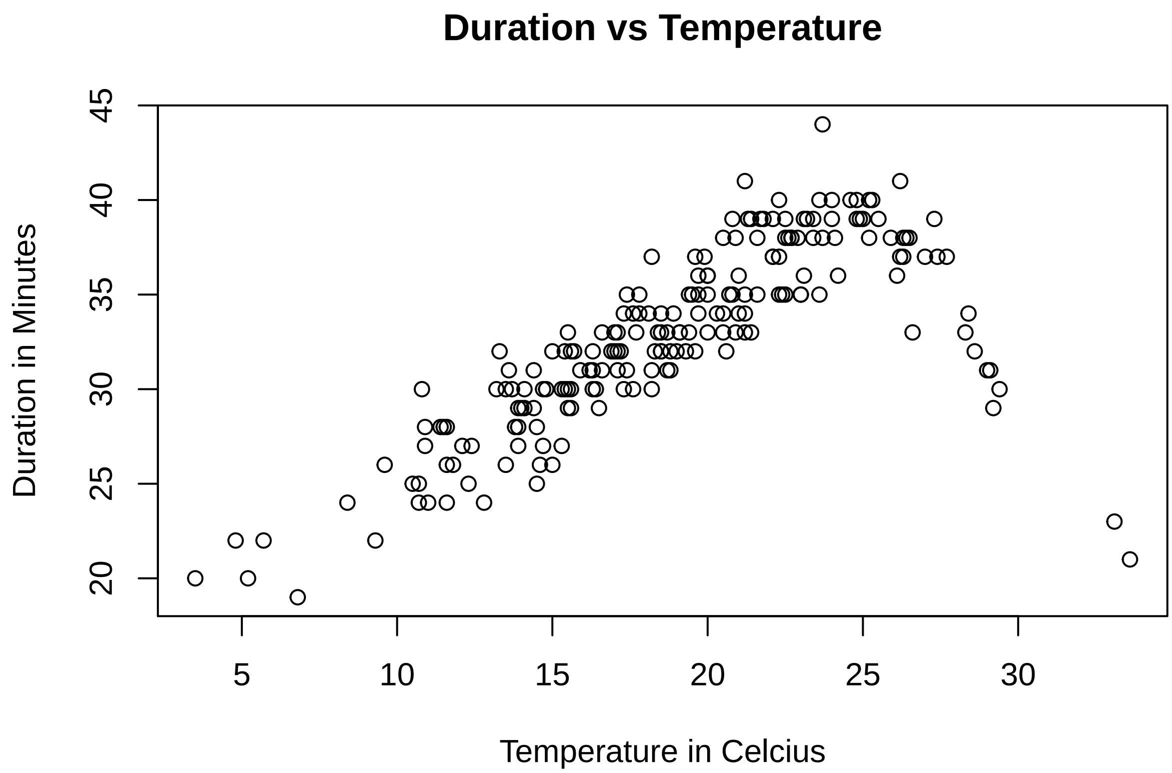


Model assumptions are met, the presence of outliers proves to be impactful to the model.

Task 6 Non-parametric Regression for Duration~Temperature

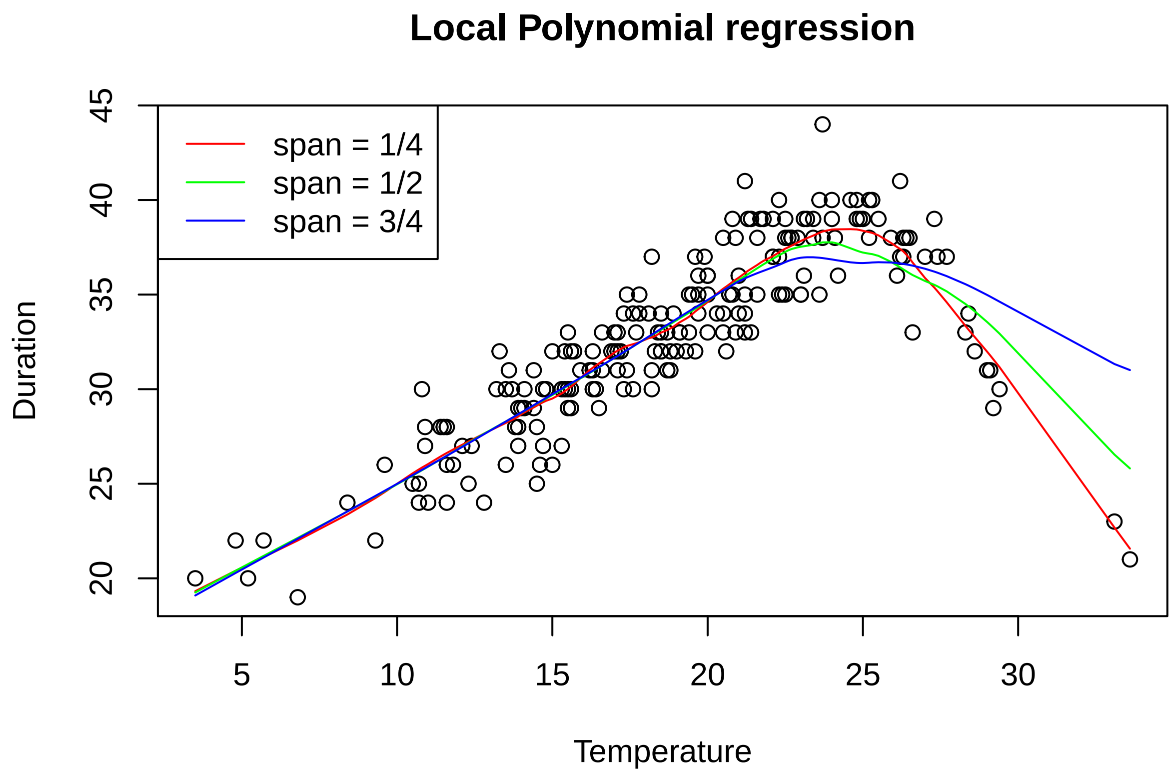
Data exploration

```
plot(data.training$temperature,data.training$duration,
     main="Duration vs Temperature",
     xlab="Temperature in Celcius",ylab="Duration in Minutes")
```



local linear model degree=1

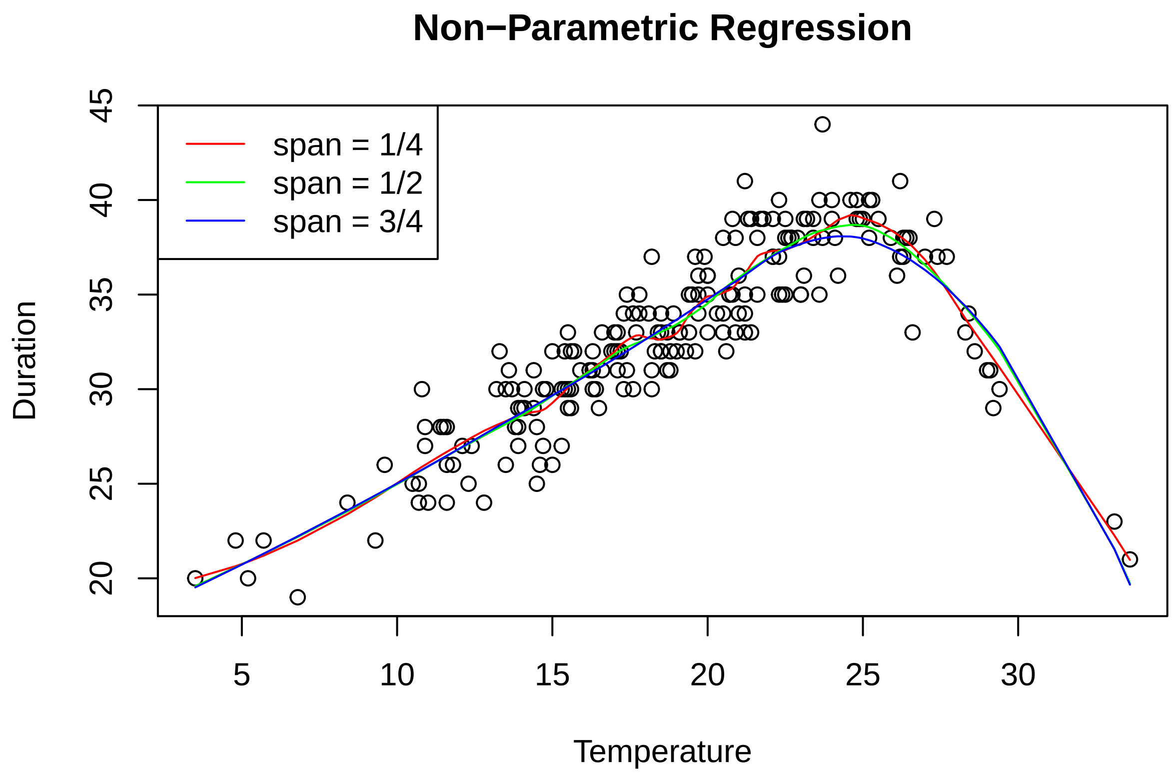
```
plot(data.training$temperature, data.training$duration,
      main = "Local Polynomial regression",xlab = "Temperature",ylab="Duration")
s <- c(1/4, 2/4, 3/4)
colors <- c("red", "green", "blue")
for (i in 1:length(s)) {
  lines(data.training$temperature,
  predict(loess(data.training$duration ~ data.training$temperature,
  span = s[i], degree = 1), data = mcycle), col = colors[i])}
legend("topleft", c("span = 1/4", "span = 1/2", "span = 3/4"), lty = 1, col = colors)
```



Fit could be better, struggles at the tail for temp>25.

local polynomial model degree=2

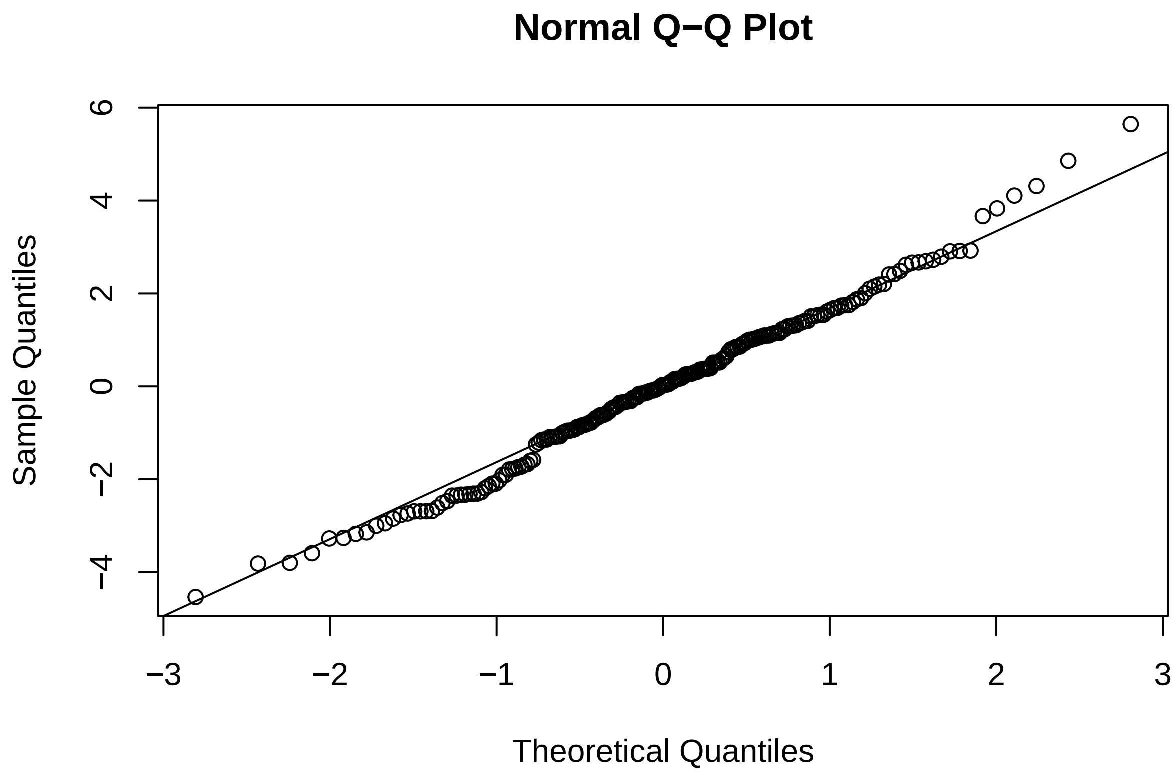
```
plot(data.training$temperature, data.training$duration,
     main = "Non-Parametric Regression",xlab = "Temperature",ylab="Duration")
s <- c(1/4, 2/4, 3/4)
colors <- c("red", "green", "blue")
for (i in 1:length(s)){
  lines(data.training$temperature,
        predict(loess(data.training$duration ~ data.training$temperature,
                      span = s[i], degree = 2), data = mcycle), col = colors[i])}
legend("topleft", c("span = 1/4", "span = 1/2", "span = 3/4"), lty = 1, col = colors)
```



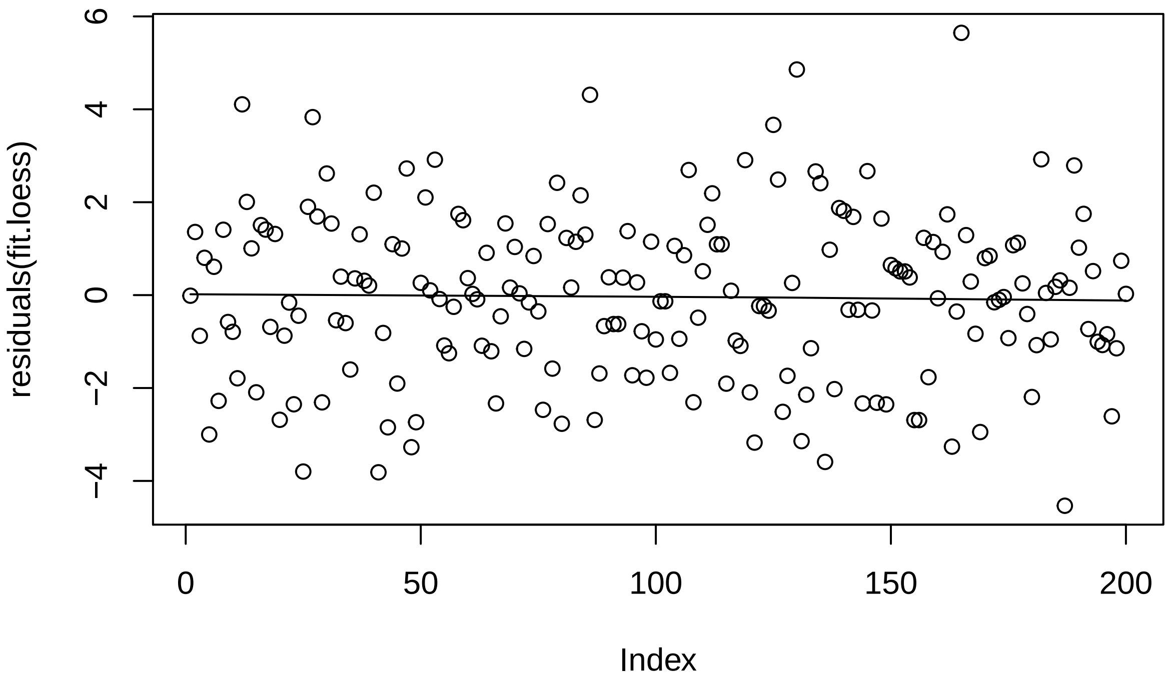
2nd degree fit proves to be quite sufficient.

Model Diagnosticsfor non-parametric model

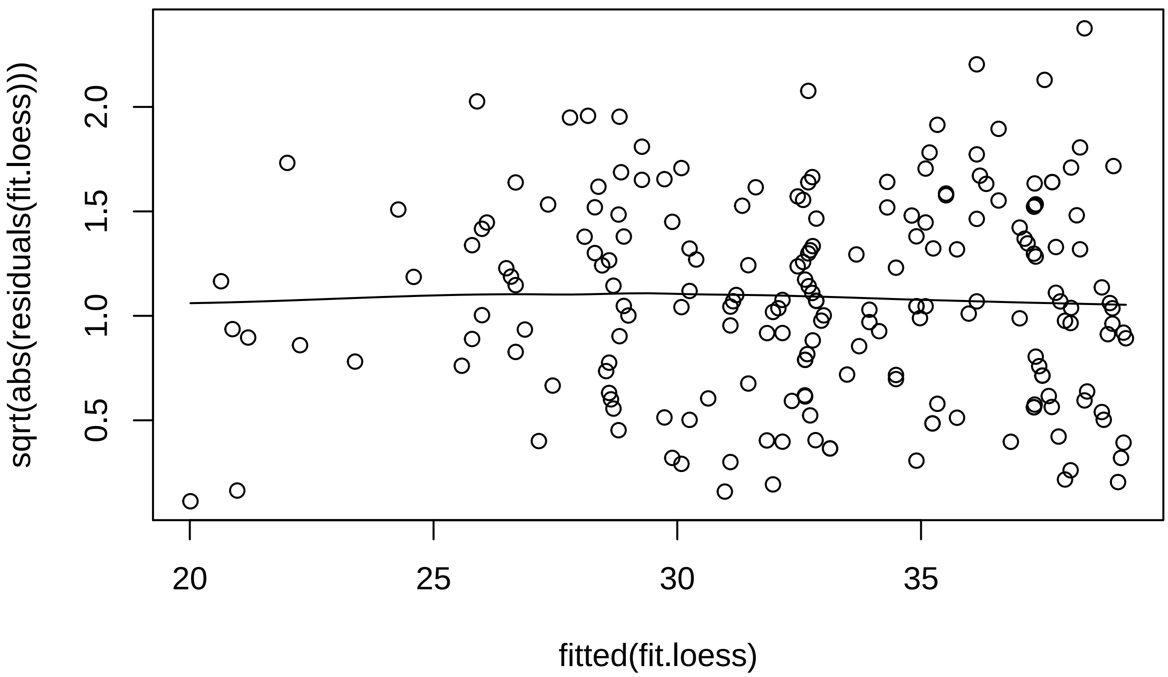
```
fit.loess <- loess(duration ~ temperature, span = 1/4, degree = 2,data=data.training)
qqnorm(residuals(fit.loess))
qqline(residuals(fit.loess))
```




```r
scatter.smooth(residuals(fit.loess), span = 1, degree = 1)
```




```
scatter.smooth(fitted(fit.loess), sqrt(abs(residuals(fit.loess))), span = 1, degree = 1)
```




```
shapiro.test(residuals(fit.loess))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.loess)
## W = 0.99311, p-value = 0.4752
```

Heavy tails in QQ plot show potential deviation from normality, Shapiro-Wilks test fails to reject normaility. Varaince appears to be constant. Linearity appears to hold as well.

Confidence interval for several temperature measurement

```
t <- c(seq(6, 33, by = 3))
t.pred <- predict(fit.loess, t, se = TRUE)
t.upper <- t.pred$fit + qnorm(0.975) * t.pred$se.fit
t.lower <- t.pred$fit - qnorm(0.975) * t.pred$se.fit
data.frame("pred" = t.pred$fit, "lower" = t.lower, "upper" = t.upper)
```

```
##       pred    lower    upper
## 1  21.40291 20.12482 22.68100
## 2  23.97338 22.89848 25.04828
## 3  27.06465 26.16349 27.96581
## 4  29.27451 28.45160 30.09742
## 5  32.74872 31.92041 33.57704
## 6  35.73759 34.92458 36.55059
## 7  38.70893 37.84195 39.57590
## 8  36.84233 35.96330 37.72136
```

```
## 9  29.74875 28.55294 30.94455
## 10 22.51513 20.26060 24.76966
```

Fitting a polynomial model to predict Duration~Temperature. Using Forward selection to reduce model to most influential predictors.

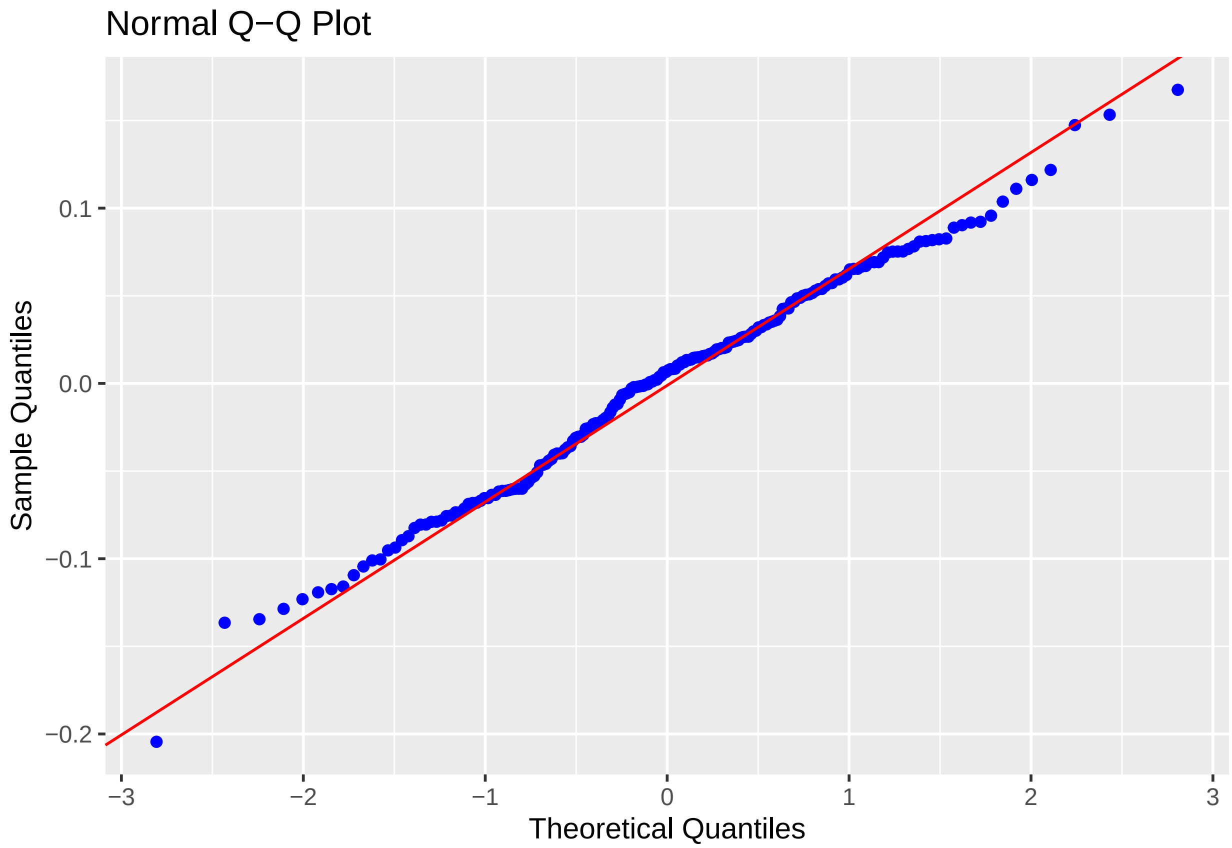
Fitting final polynomial model of log(duration) ~ I(temperature^3) + I(temperature^4). (Normality test failed so a log transform was taken to adress this.)

```
fit.dur.poly3=lm(formula = log(duration) ~ I(temperature^3) + I(temperature^4),
    data = data.training)
summary(fit.dur.poly3)
```

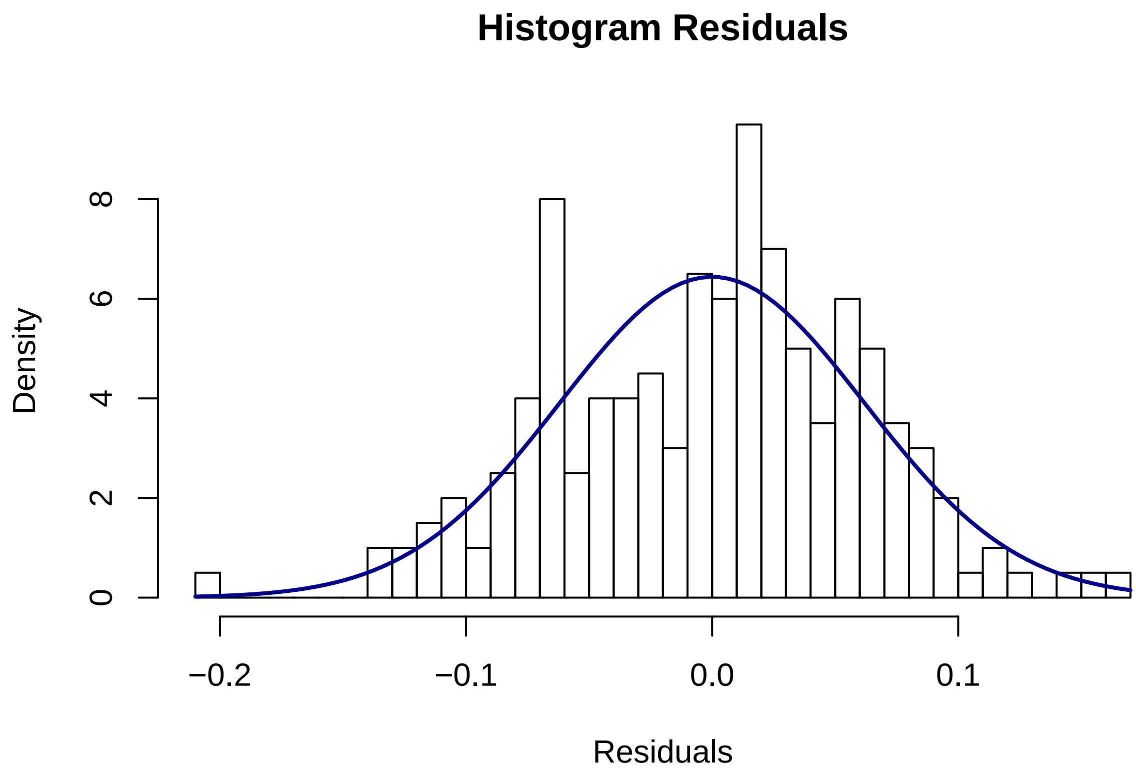
```
##
## Call:
## lm(formula = log(duration) ~ I(temperature^3) + I(temperature^4),
##     data = data.training)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.204459 -0.045988  0.007033  0.043664  0.167501
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.113e+00  1.189e-02  261.95   <2e-16 ***
## I(temperature^3)  1.425e-04  4.671e-06   30.52   <2e-16 ***
## I(temperature^4) -4.355e-06  1.534e-07  -28.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06226 on 197 degrees of freedom
## Multiple R-squared:  0.8463, Adjusted R-squared:  0.8448
## F-statistic: 542.5 on 2 and 197 DF,  p-value: < 2.2e-16
```

Model Diagnostics

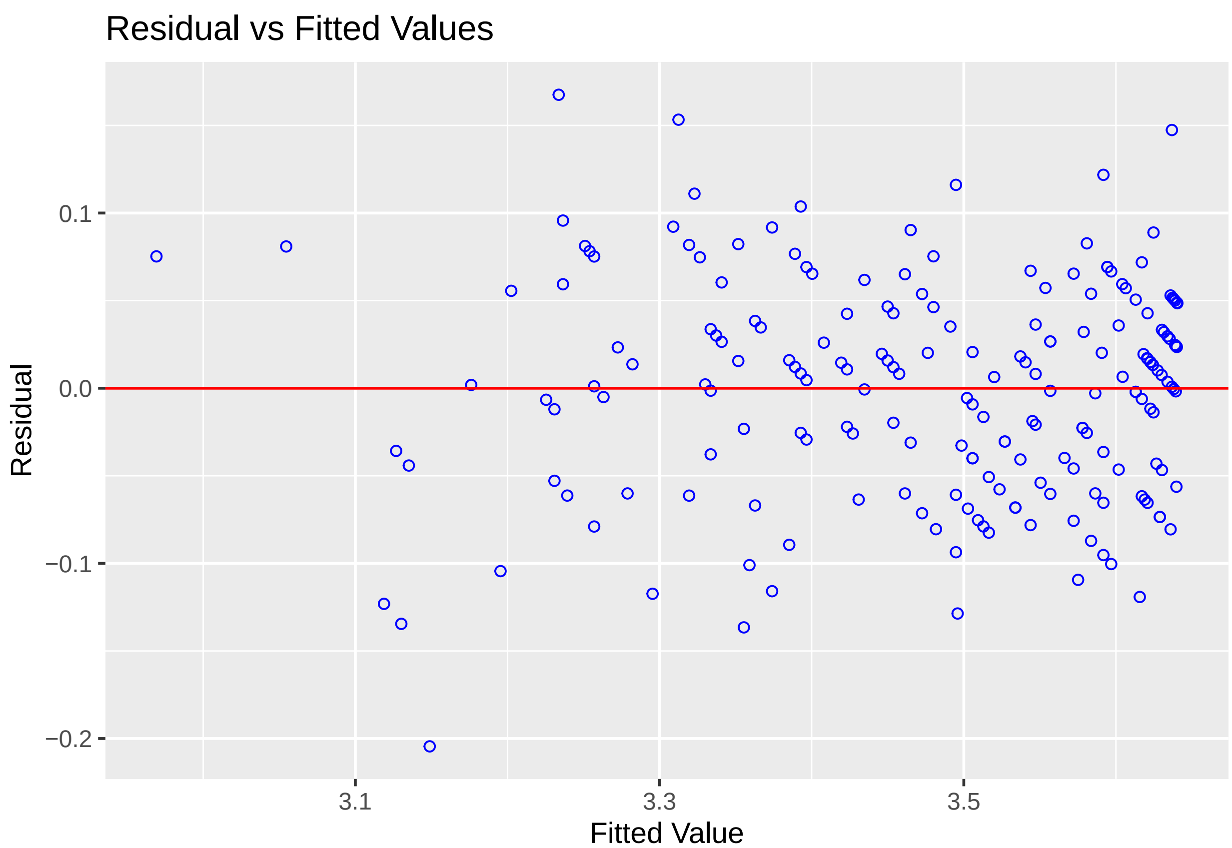
```
mod=fit.dur.poly3
#QQ-Plot and Histogram to visualize normality of errors
ols_plot_resid_qq(mod)
```




```r
hist(mod$residuals,breaks = 50,
     xlab="Residuals", main="Histogram Residuals",
     probability = T)
curve(dnorm(x, mean=mean(mod$residuals), sd=sd(mod$residuals)),
      col="darkblue", lwd=2, add=TRUE, yaxt="n")
```




```
##Visualiztion of Residuals vs. Fitted Values to evaluate homoscedasticity
ols_plot_resid_fit(mod)
```




```
#Breusch-Pagan Test for homoscedasticity
ncvTest(mod)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 16.04692, Df = 1, p = 6.1792e-05
```

```
#Normality of errors tests
ols_test_normality(mod)
```

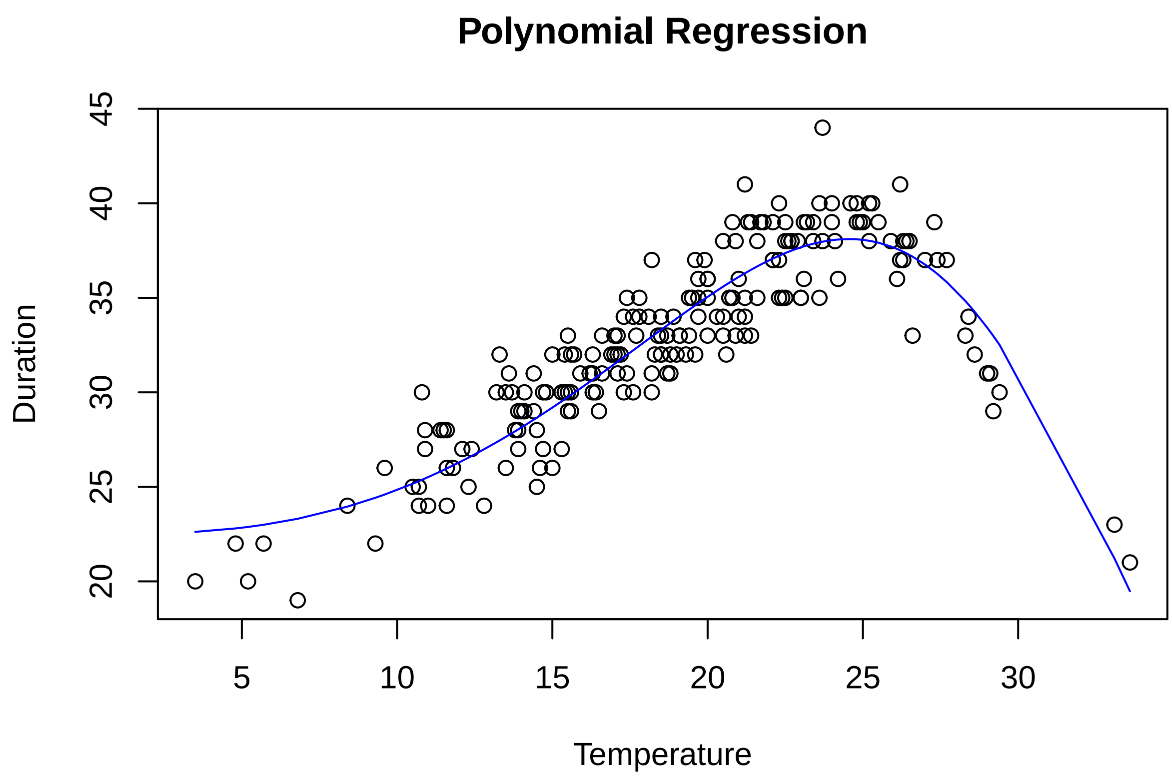
```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present
## for the Kolmogorov-Smirnov test

## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9926         0.4057
## Kolmogorov-Smirnov        0.0616         0.4347
## Cramer-von Mises         58.6631         0.0000
## Anderson-Darling          0.582          0.1283
## -----------------------------------------------
```

```
ols_test_correlation(mod)
```

```
## [1] 0.9959428
```

Plot of Quadratic Regression model along with regression line fit.

```r
fit.dur.poly.coef=coefficients(fit.dur.poly3)
temp=data.training$temperature
y=fit.dur.poly.coef[1]+fit.dur.poly.coef[2]*temp^3+fit.dur.poly.coef[3]*temp^4
plot(data.training$temperature, data.training$duration,
     main = "Polynomial Regression",xlab = "Temperature",ylab="Duration")
lines(x=data.training$temperature,y=exp(y),col="blue")
```



Comparing models with RMSE(fitted,actual)

```r
#Loess model RMSE
rmse(fit.loess$fitted,data.training$duration)
```

```
## [1] 1.780385
```

```r
#polynomial model RMSE
rmse(exp(fitted.values(fit.dur.poly3)),data.test$duration)
```

```
## [1] 6.440638
```

Non-parametric model proves to be superior than the polynomial model.