STAT611 Project

Irem Celen

7/19/2018

This document contains the descriptive statistics, assumption evaluation, and model selection with stepwise regression through AIC. For further results from different model selection models, please see the SAS outputs at the end of the document.

1. Yield Data

Necessary libraries

```
library(readxl)
library(dplyr)
library(MASS)
library(PerformanceAnalytics)
library(fmsb)
library(mctest)
library(car)
library(tidyr)
```

Read the data

```
yield <- read_excel("YieldLossData.xlsx", col_names = T)
```

Descriptive statistics

```
summary(yield)
```

```
##    YieldLoss           RPM            Temp1           Temp2      
##  Min.   : 32.37   Min.   :25.00   Min.   :108.7   Min.   :126.8  
##  1st Qu.: 54.40   1st Qu.:25.00   1st Qu.:116.1   1st Qu.:140.5  
##  Median : 70.48   Median :30.00   Median :120.3   Median :145.2  
##  Mean   : 69.96   Mean   :31.85   Mean   :119.7   Mean   :145.5  
##  3rd Qu.: 83.26   3rd Qu.:35.00   3rd Qu.:122.8   3rd Qu.:150.3  
##  Max.   :109.85   Max.   :40.00   Max.   :130.7   Max.   :168.5  
##       Flow           Conc            Line         Operator        
##  Min.   :200.0   Min.   :3.435   Min.   : 5.0   Length:100        
##  1st Qu.:200.0   1st Qu.:4.094   1st Qu.: 7.0   Class :character  
##  Median :250.0   Median :5.945   Median : 9.0   Mode  :character  
##  Mean   :249.5   Mean   :5.797   Mean   : 9.6                     
##  3rd Qu.:300.0   3rd Qu.:7.731   3rd Qu.:12.0                     
##  Max.   :300.0   Max.   :8.373   Max.   :15.0                     
##     Vacuum             Vendor         
##  Length:100         Length:100        
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
```

```
#Check the variables resembling categorical variables. If categorical, convert to factors
unique(yield$Flow) #three categories
```

```
## [1] 200 300 250
```

```
yield$Flow <- as.factor(yield$Flow)
unique(yield$RPM) #four categories
```

```
## [1] 25 30 35 40
```

```
yield$RPM <- as.factor(yield$RPM)
head(unique(yield$Conc)) #Continous
```

```
## [1] 3.929089 3.927433 3.714301 4.029295 3.967699 3.862349
```

```
unique(yield$Line) #five categories
```

```
## [1]  5  7  9 12 15
```

```
yield$Line <- as.factor(yield$Line)
yield$Operator <- as.factor(yield$Operator)
yield$Vendor <- as.factor(yield$Vendor)
yield$Vacuum <- as.factor(yield$Vacuum)
```

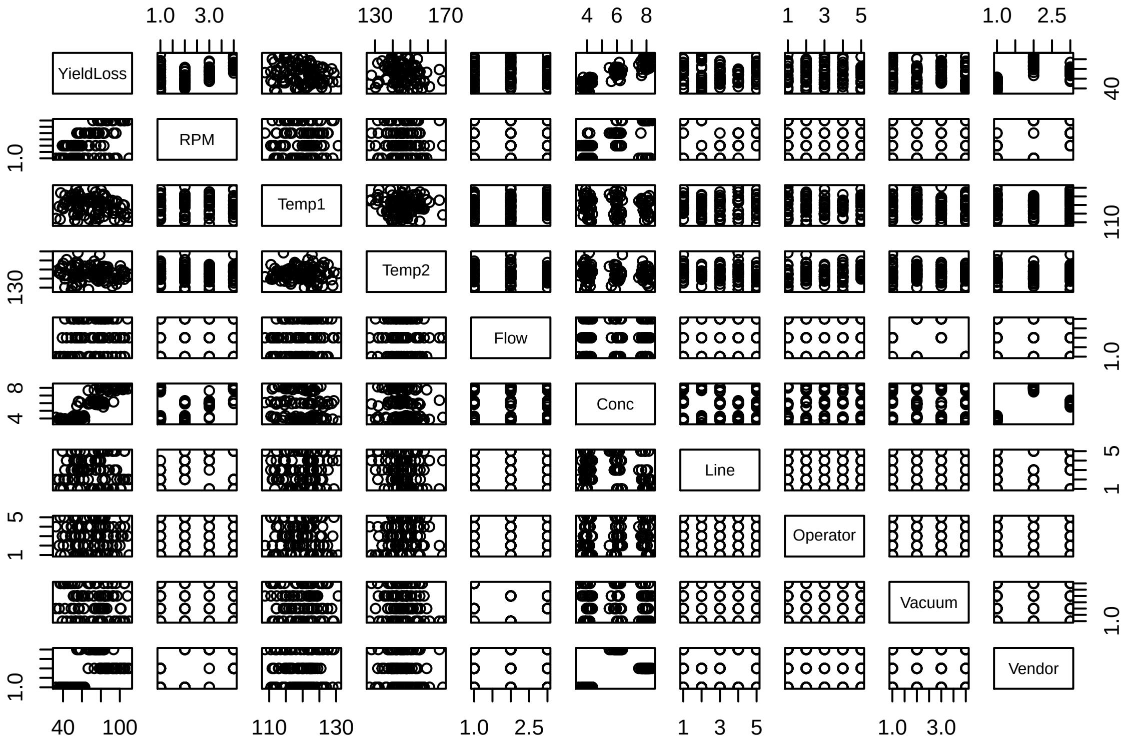
Check missing values

```
which(is.na(yield))
```

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

Matrix plot for all the variables

```
pairs(yield)
```



Fit the model

```
#Fit the model and see the variable significance
fit1=lm(YieldLoss ~ ., data=yield); summary(fit1)
```

```
##
## Call:
## lm(formula = YieldLoss ~ ., data = yield)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0380  -3.7317   0.6335   4.9433  16.9857
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.7580    37.5440   0.819  0.41514
## RPM30         -1.3987     2.7879  -0.502  0.61728
## RPM35          3.1590     3.8199   0.827  0.41076
## RPM40         10.4910     3.0981   3.386  0.00111 **
## Temp1         -0.4147     0.1760  -2.356  0.02099 *
## Temp2          0.1294     0.1141   1.134  0.26032
## Flow250        7.3421     5.0873   1.443  0.15296
## Flow300        5.7100     3.7223   1.534  0.12907
## Conc           9.4880     4.5760   2.073  0.04143 *
## Line7          5.6970     2.8116   2.026  0.04616 *
## Line9         10.1403     3.2696   3.101  0.00268 **
## Line12         8.4116     3.3169   2.536  0.01321 *
## Line15        10.9935     3.2243   3.410  0.00103 **
## OperatorMary  -2.0651     2.6275  -0.786  0.43428
## OperatorMike   0.4670     2.5839   0.181  0.85703
## OperatorSam    0.7062     2.5744   0.274  0.78457
## OperatorSue   -3.5761     2.6184  -1.366  0.17593
## VacuumL        3.2399     4.3596   0.743  0.45961
## VacuumM        1.2058     3.1386   0.384  0.70188
## VacuumN        6.1073     5.6271   1.085  0.28111
## VendorGrumpy  -1.1507    17.8254  -0.065  0.94869
## VendorSloppy  -0.9468     9.8407  -0.096  0.92360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.018 on 78 degrees of freedom
## Multiple R-squared:  0.865,  Adjusted R-squared:  0.8286
## F-statistic: 23.79 on 21 and 78 DF,  p-value: < 2.2e-16
```

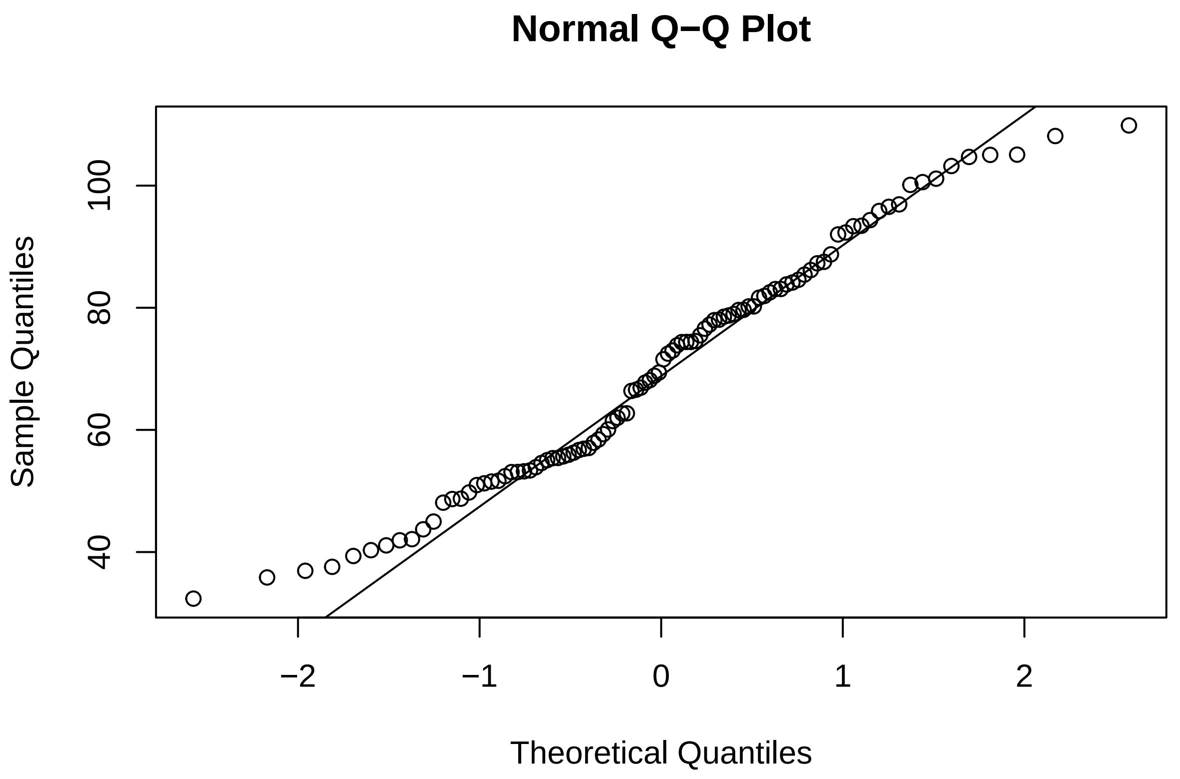
```r
anova(fit1)
```

```
## Analysis of Variance Table
##
## Response: YieldLoss
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## RPM        3 16045.0  5348.3  83.1888 < 2.2e-16 ***
## Temp1      1  1482.1  1482.1  23.0529 7.463e-06 ***
## Temp2      1   434.0   434.0   6.7512  0.011197 *
## Flow       2   853.8   426.9   6.6399  0.002173 **
## Conc       1 11895.4 11895.4 185.0227 < 2.2e-16 ***
## Line       4  1091.2   272.8   4.2430  0.003689 **
## Operator   4   234.6    58.7   0.9123  0.461131
## Vacuum     3    85.5    28.5   0.4432  0.722749
## Vendor     2     1.2     0.6   0.0093  0.990768
## Residuals 78  5014.7    64.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assumption testing

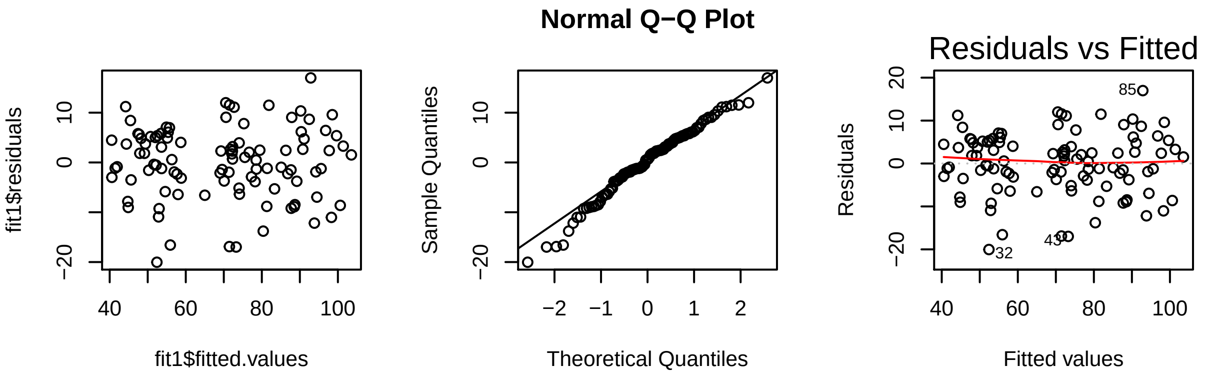
1.Check if the dependent variable is normally distributed:

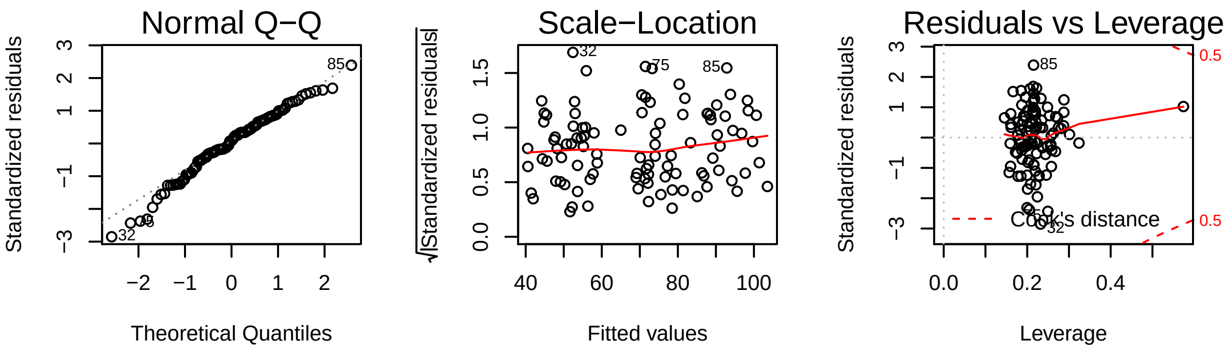
```r
qqnorm(yield$YieldLoss); qqline(yield$YieldLoss) #looks normal
```



2.Residual plots

```
# residuals vs fitted values
par(mfrow = c(2, 3))
plot(fit1$residuals ~ fit1$fitted.values) # nonlinear trend qqnorm(fit1$res)
qqnorm(fit1$res);qqline(fit1$res);
plot(fit1)
```





The residuals look homoscedastic and normally distributed. Transformation will not be applied.

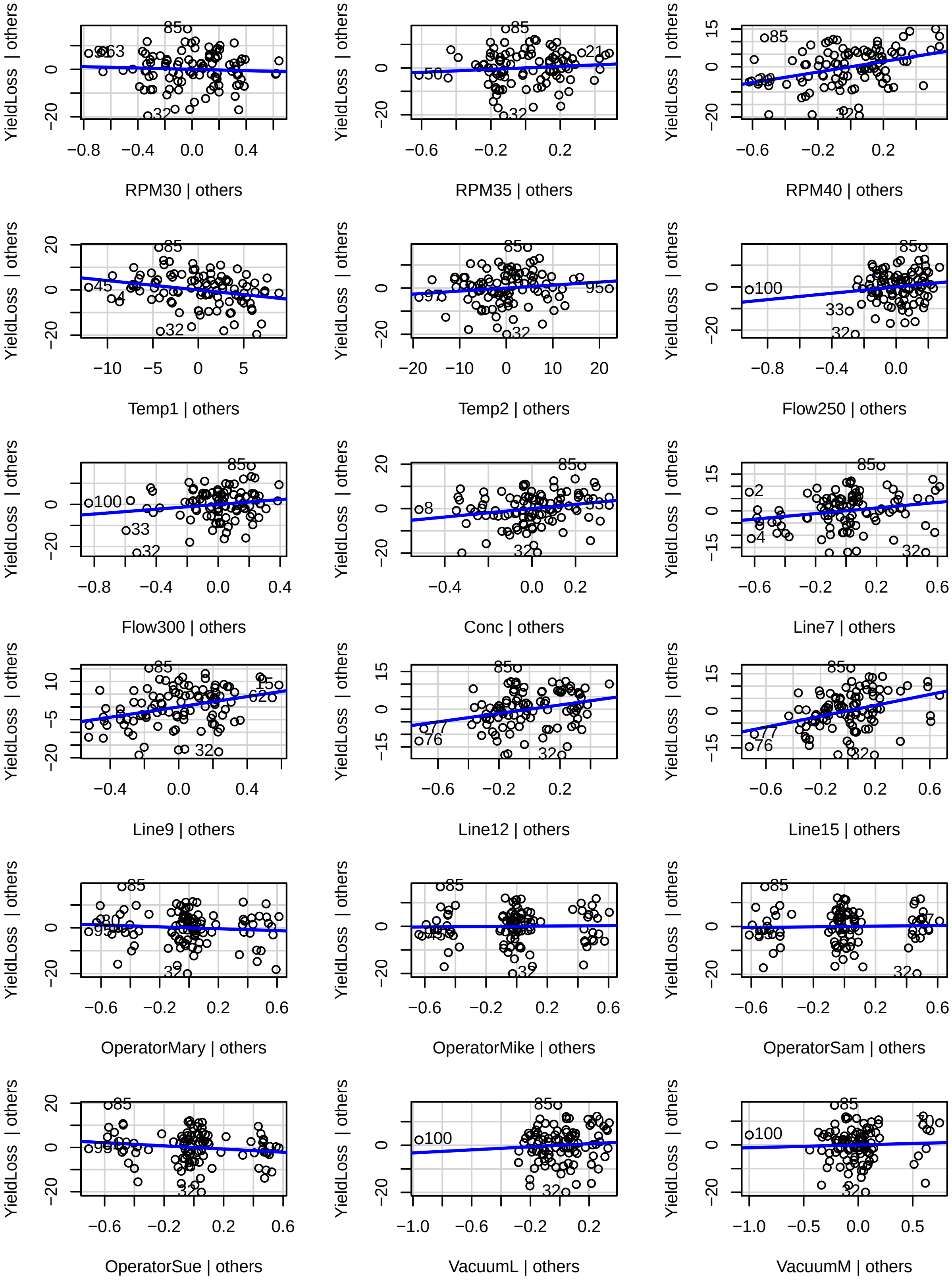
3. Check for outliers

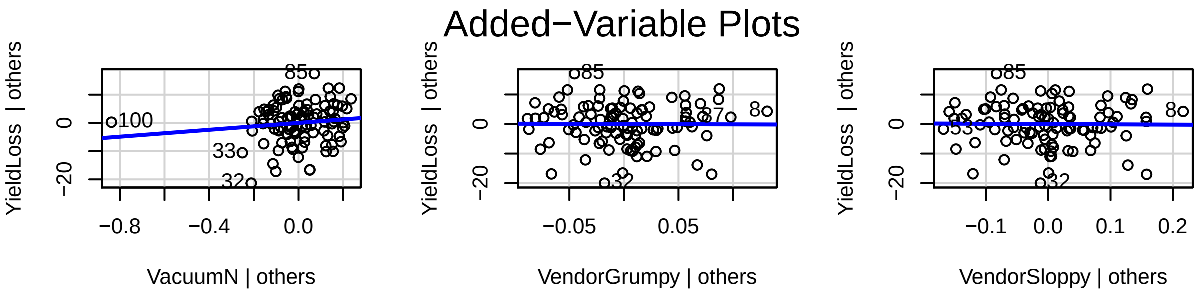
```
outlierTest(fit1)
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 32 -2.993608          0.0037049      0.37049
```

4. Influential observation detection with Cook's D and leverage obs. with hat values

```
avPlots(fit1)
```

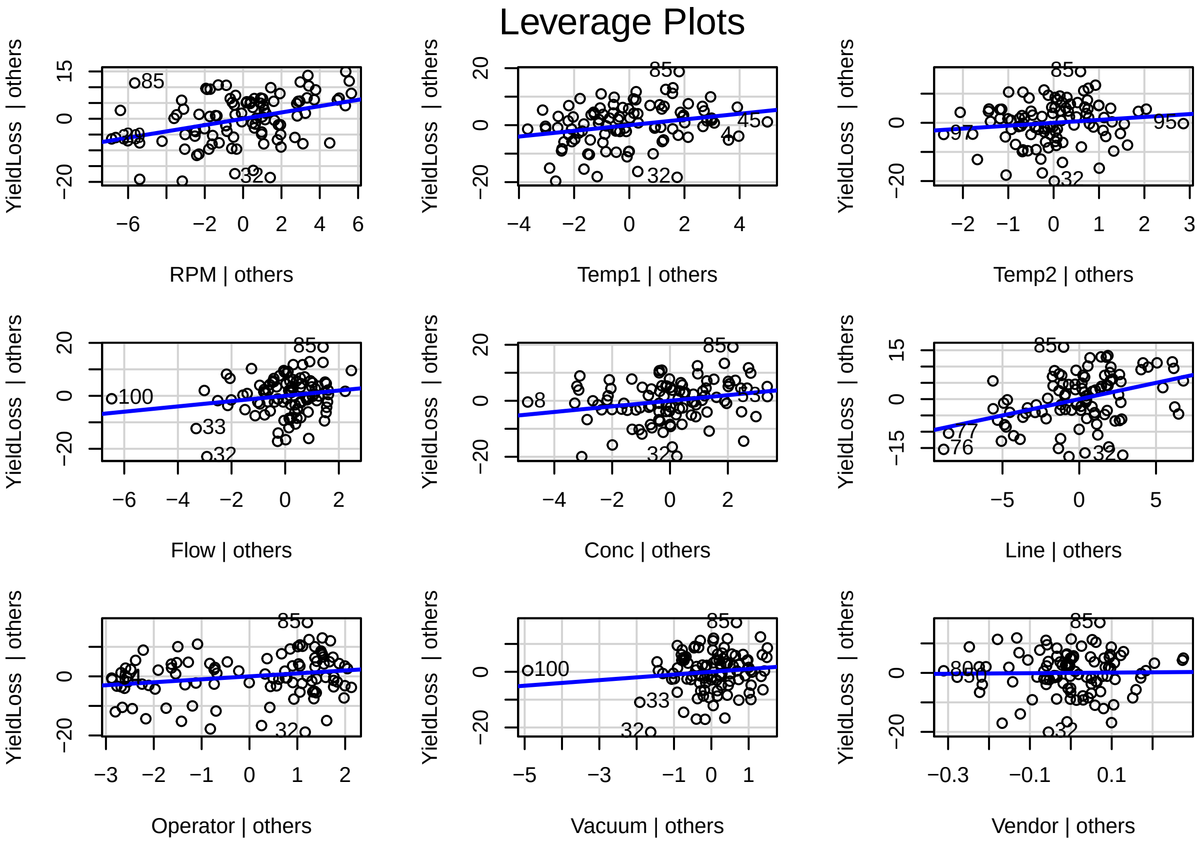

YieldLoss | others
RPM30 | others
RPM35 | others
RPM40 | others
Temp1 | others
Temp2 | others
Flow250 | others
Flow300 | others
Conc | others
Line7 | others
Line9 | others
Line12 | others
Line15 | others
OperatorMary | others
OperatorMike | others
OperatorSam | others
OperatorSue | others
VacuumL | others
VacuumM | others
85
32
63
21
50
45
4
97
95
100
33
8
2
77
76
15
62
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
−1.0
−0.5
0.5
−20
−15
−10
−5
0
5
10
15
20



```r
any(cooks.distance(fit1) > 1) #no influential observation
```

```
## [1] FALSE
```

```r
#any observations 2-3 times greater than the average hat value to be considered as "leverage" observati
#hatvalues(fit1)
leveragePlots(fit1)
```



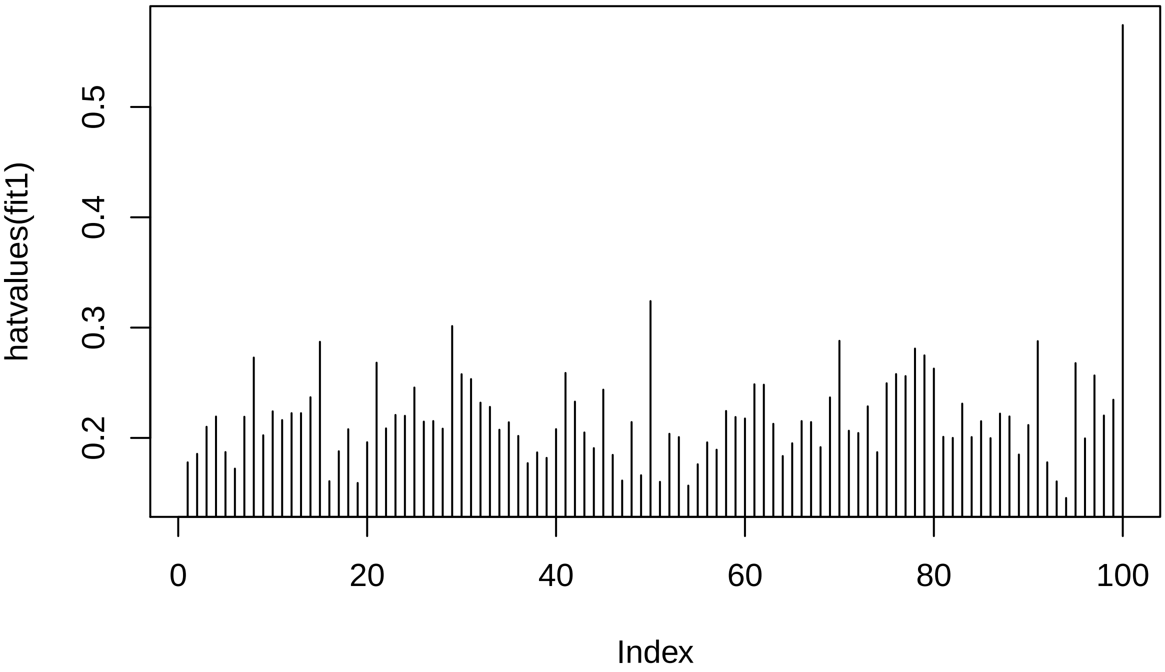
```r
hv <- as.data.frame(hatvalues(fit1))
mn <-mean(hatvalues(fit1))
hv$warn <- ifelse(hv[, 'hatvalues(fit1)']>3*mn, 'x3',
   ifelse(hv[, 'hatvalues(fit1)']>2*mn, 'x3', '-' ))
subset(hv, warn=="x3")
```

```
##     hatvalues(fit1) warn
## 100       0.5742299   x3
```

```r
subset(hv, warn%in%c("x2", "x3"))
```

```
##     hatvalues(fit1) warn
## 100       0.5742299   x3
```

```
plot(hatvalues(fit1), type = "h") #100th record is a leverage point.
```



```
#Remove the leverage point and see how it improves the model
fit2 <- lm(YieldLoss ~ ., data=yield[-100,]); summary(fit2)
```

```
##
## Call:
## lm(formula = YieldLoss ~ ., data = yield[-100, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7696  -3.9183   0.6444   5.2353  16.8678
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   19.1306    39.2073   0.488  0.62698
## RPM30         -1.6597     2.7986  -0.593  0.55488
## RPM35          2.2329     3.9240   0.569  0.57098
## RPM40         10.1988     3.1102   3.279  0.00156 **
## Temp1         -0.3962     0.1769  -2.240  0.02799 *
## Temp2          0.1577     0.1173   1.344  0.18302
## Flow250       11.9761     6.8035   1.760  0.08233 .
## Flow300        7.9807     4.3301   1.843  0.06917 .
## Conc           9.5369     4.5748   2.085  0.04041 *
## Line7          5.5672     2.8135   1.979  0.05142 .
## Line9          9.8439     3.2813   3.000  0.00364 **
## Line12         7.9875     3.3415   2.390  0.01927 *
## Line15        10.1042     3.3378   3.027  0.00336 **
## OperatorMary  -1.6956     2.6513  -0.640  0.52438
## OperatorMike   1.0528     2.6454   0.398  0.69176
## OperatorSam    1.1469     2.6092   0.440  0.66147
## OperatorSue   -3.2863     2.6327  -1.248  0.21572
## VacuumL        6.8109     5.5787   1.221  0.22586
## VacuumM        3.1347     3.6583   0.857  0.39416
## VacuumN       11.3025     7.5705   1.493  0.13954
## VendorGrumpy  -0.4955    17.8310  -0.028  0.97790
## VendorSloppy  -0.6262     9.8424  -0.064  0.94944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.016 on 77 degrees of freedom
## Multiple R-squared:  0.8663, Adjusted R-squared:  0.8298
## F-statistic: 23.75 on 21 and 77 DF,  p-value: < 2.2e-16
```

```
#model didn't improve drastically, so keep the leverage point in the dataset.
```

5. Check for multicollinearity

```
#Check the VIF and see if any VIF value is greater than 10
vif(fit1) #No VIF is larger than 10. Thus, ridge or PCA will not be used.
```

```
##                 GVIF Df GVIF^(1/(2*Df))
## RPM        9.606887  3        1.458021
## Temp1      1.186563  1        1.089295
## Temp2      1.207993  1        1.099087
## Flow      15.464910  2        1.983065
## Conc      88.883720  1        9.427816
## Line       4.337246  4        1.201301
## Operator   1.179372  4        1.020837
## Vacuum    15.619520  3        1.581046
## Vendor   219.948771  2        3.851061
```

Conc seems to have a potential multicollinearity but the VIF is less than 10. Thus, ridge or PCA will not be used.

Model selection

Select the model by using AIC in stepwise regression

```
summary(selectedMod)
```

```
##
## Call:
## lm(formula = YieldLoss ~ RPM + Temp1 + Conc + Line, data = yield)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.1355  -3.6375   0.2227   4.5799  19.4156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.7538    20.3340   2.693 0.008452 **
## RPM30        -1.0724     2.6250  -0.409 0.683850
## RPM35         3.2295     2.8788   1.122 0.264932
## RPM40        10.8757     2.9318   3.710 0.000359 ***
## Temp1        -0.3980     0.1613  -2.467 0.015518 *
## Conc          9.0584     0.6677  13.567  < 2e-16 ***
## Line7         5.8473     2.6196   2.232 0.028086 *  
## Line9        10.9286     2.9980   3.645 0.000447 ***
## Line12        9.5922     3.1154   3.079 0.002753 ** 
## Line15       11.6245     2.8822   4.033 0.000115 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.811 on 90 degrees of freedom
## Multiple R-squared:  0.8521, Adjusted R-squared:  0.8373 
## F-statistic: 57.63 on 9 and 90 DF,  p-value: < 2.2e-16
```

Final model is: YieldLoss ~ RPM + Temp1 + Conc + Line

2. Inverter Data

Read the data

```
Inverter <- read_excel("InverterData.xlsx", col_names = T)
```

Descriptive statistics

```
summary(Inverter)
```

```
##    WidthNMOS       LengthNMOS      WidthPMOS      LengthPMOS   
##  Min.   : 2.00   Min.   : 2.00   Min.   : 2.0   Min.   : 2.00  
##  1st Qu.: 3.00   1st Qu.: 4.00   1st Qu.: 3.0   1st Qu.: 3.00  
##  Median : 5.00   Median : 6.00   Median : 4.0   Median : 4.00  
##  Mean   : 6.48   Mean   : 8.48   Mean   : 4.8   Mean   : 4.64  
##  3rd Qu.: 8.00   3rd Qu.:10.00   3rd Qu.: 6.0   3rd Qu.: 6.00  
##  Max.   :16.00   Max.   :30.00   Max.   :12.0   Max.   :12.00  
##     Setpoint   TransientPt   
##  Min.   : 0   Min.   :0.201  
##  1st Qu.:25   1st Qu.:0.379  
##  Median :50   Median :0.806  
##  Mean   :38   Mean   :2.372  
##  3rd Qu.:50   3rd Qu.:3.345  
##  Max.   :75   Max.   :9.210
```

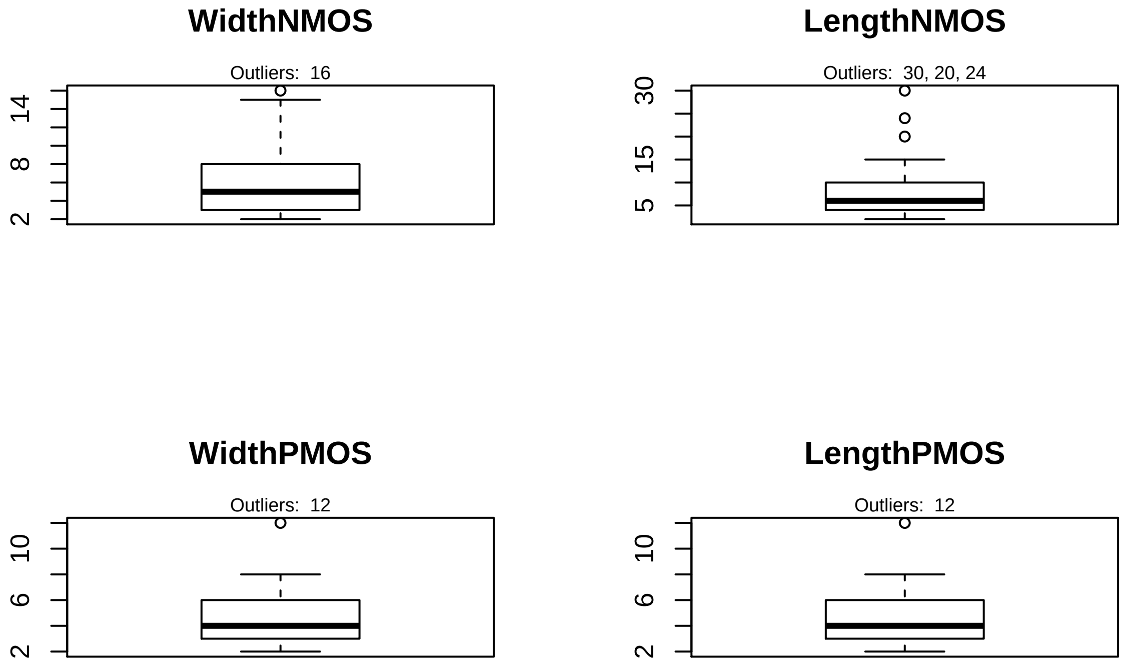
```
#Correlation between variables
cor(Inverter) #No strong correlation
```

```
##                WidthNMOS LengthNMOS  WidthPMOS LengthPMOS     Setpoint
## WidthNMOS    1.000000000  0.2478995 -0.2222020 -0.2588852 -0.002109001
## LengthNMOS   0.247899530  1.0000000  0.1289057  0.1598969 -0.387679001
## WidthPMOS   -0.222202020  0.1289057  1.0000000  0.3463582 -0.131881027
## LengthPMOS  -0.258885196  0.1598969  0.3463582  1.0000000 -0.336189478
## Setpoint    -0.002109001 -0.3876790 -0.1318810 -0.3361895  1.000000000
## TransientPt -0.250791397  0.4024670  0.4731332 -0.1584868 -0.057718614
##              TransientPt
## WidthNMOS   -0.25079140
## LengthNMOS   0.40246702
## WidthPMOS    0.47313318
## LengthPMOS  -0.15848683
## Setpoint    -0.05771861
## TransientPt  1.00000000
```

```r
#Check the variables resembling categorical variables. If categorical, convert to factors
str(Inverter) ##There are only 25 records, Setpoint looks like a categorical variable
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    25 obs. of  6 variables:
##  $ WidthNMOS  : num  3 8 3 4 8 10 8 6 4 16 ...
##  $ LengthNMOS : num  3 30 6 4 7 20 6 24 10 12 ...
##  $ WidthPMOS  : num  3 5 6 4 6 5 3 4 12 8 ...
##  $ LengthPMOS : num  3 8 6 12 5 5 3 4 4 4 ...
##  $ Setpoint   : num  0 0 0 0 0 0 25 25 25 25 ...
##  $ TransientPt: num  0.787 0.293 1.71 0.203 0.806 ...
```

```r
Inverter$Setpoint <- as.factor(Inverter$Setpoint)
#See the distribution of the independent variables
par(mfrow = c(2, 2))
boxplot(Inverter$WidthNMOS, main="WidthNMOS")
outlierWidthNMOS <- boxplot.stats(Inverter$WidthNMOS)$out
mtext(paste("Outliers: ", paste(outlierWidthNMOS, collapse=", ")), cex=0.6)
boxplot(Inverter$LengthNMOS, main="LengthNMOS")
outlierLengthNMOS <- boxplot.stats(Inverter$LengthNMOS)$out
mtext(paste("Outliers: ", paste(outlierLengthNMOS, collapse=", ")), cex=0.6)
boxplot(Inverter$WidthPMOS, main="WidthPMOS")
outlierWidthPMOS <- boxplot.stats(Inverter$WidthPMOS)$out
mtext(paste("Outliers: ", paste(outlierWidthPMOS, collapse=", ")), cex=0.6)
boxplot(Inverter$LengthPMOS, main="LengthPMOS")
outlierLengthPMOS <- boxplot.stats(Inverter$LengthPMOS)$out
mtext(paste("Outliers: ", paste(outlierLengthPMOS, collapse=", ")), cex=0.6)
```

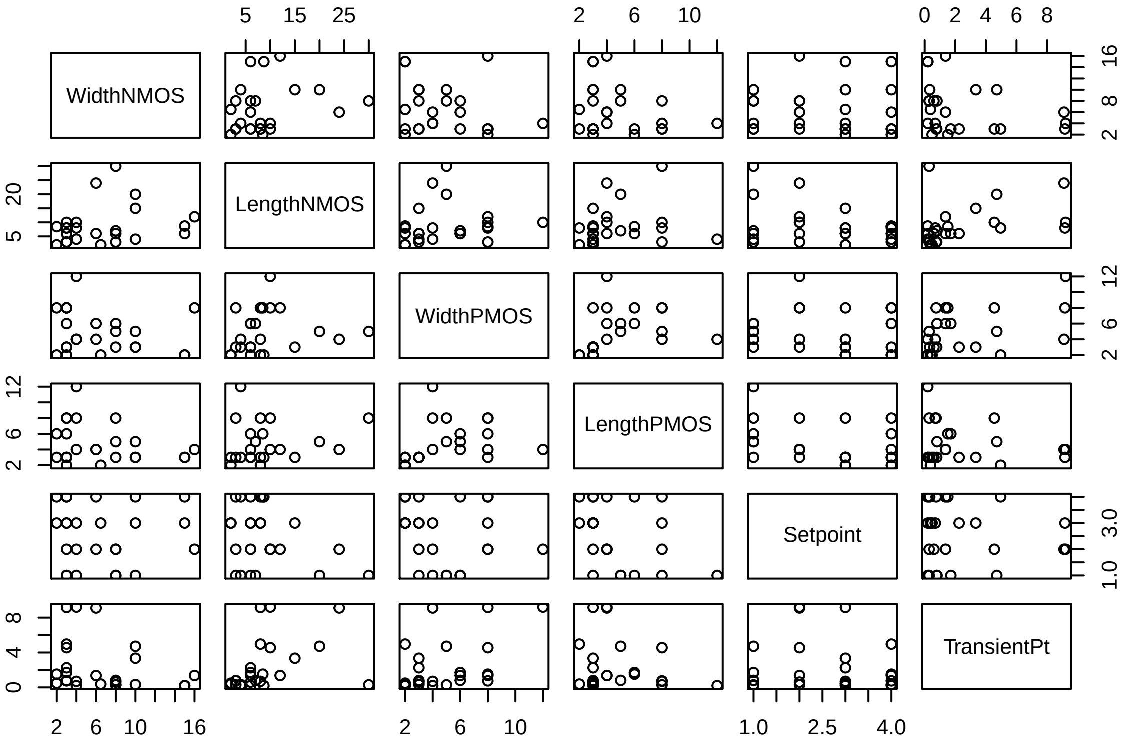


```r
#Change the outliers to the mean values
Inverter[16,"WidthNMOS"] = mean(Inverter$WidthNMOS)
Inverter[24,"LengthNMOS"] = mean(Inverter$LengthNMOS)
Inverter[20,"LengthNMOS"] = mean(Inverter$LengthNMOS)
Inverter[30,"LengthNMOS"] = mean(Inverter$LengthNMOS)
```

```
Inverter[12,"LengthPMOS"] = mean(Inverter$LengthPMOS)
Inverter[12,"WidthPMOS"] = mean(Inverter$WidthPMOS)

### Matrix plot for all the variables
```

```
pairs(Inverter)
```



Fit the model

```
#Fit the model and see the variable significance
fiti=lm(TransientPt ~ ., data=Inverter); summary(fiti)
```

```
##
## Call:
## lm(formula = TransientPt ~ ., data = Inverter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5114 -1.0422 -0.3162  1.1988  3.2672
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07198    2.05445   1.009   0.3282
## WidthNMOS   -0.29083    0.11523  -2.524   0.0226 *
## LengthNMOS   0.20076    0.07183   2.795   0.0130 *
## WidthPMOS    0.44761    0.20426   2.191   0.0436 *
## LengthPMOS  -0.52523    0.21439  -2.450   0.0262 *
## Setpoint25   1.84211    1.45059   1.270   0.2223
## Setpoint50   1.08910    1.34599   0.809   0.4303
## Setpoint75   0.17415    1.34140   0.130   0.8983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.097 on 16 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.6532, Adjusted R-squared:  0.5014
## F-statistic: 4.305 on 7 and 16 DF,  p-value: 0.007434
```

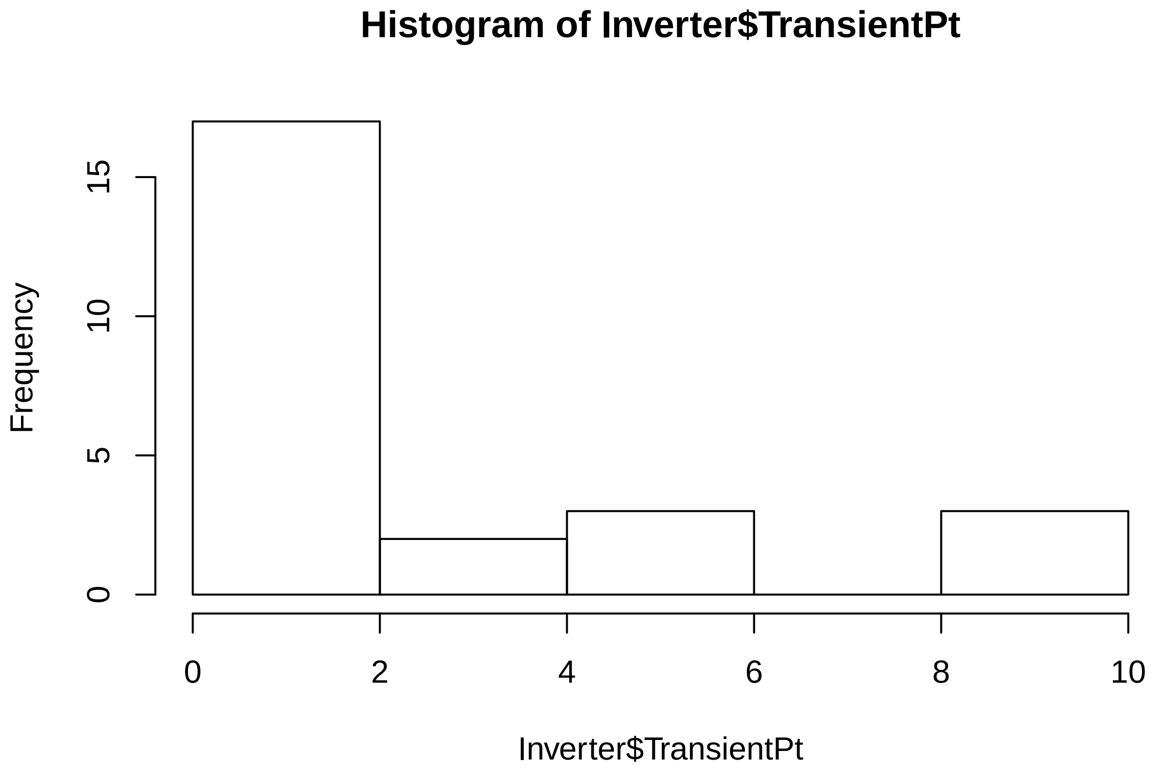
```
anova(fiti)
```

```
## Analysis of Variance Table
##
## Response: TransientPt
##             Df Sum Sq Mean Sq F value   Pr(>F)
## WidthNMOS    1 13.108  13.108  2.9803 0.103539
## LengthNMOS   1 41.187  41.187  9.3643 0.007479 **
## WidthPMOS    1 22.080  22.080  5.0202 0.039594 *
## LengthPMOS   1 45.872  45.872 10.4295 0.005243 **
## Setpoint     3 10.280   3.427  0.7791 0.522694
## Residuals   16 70.373   4.398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

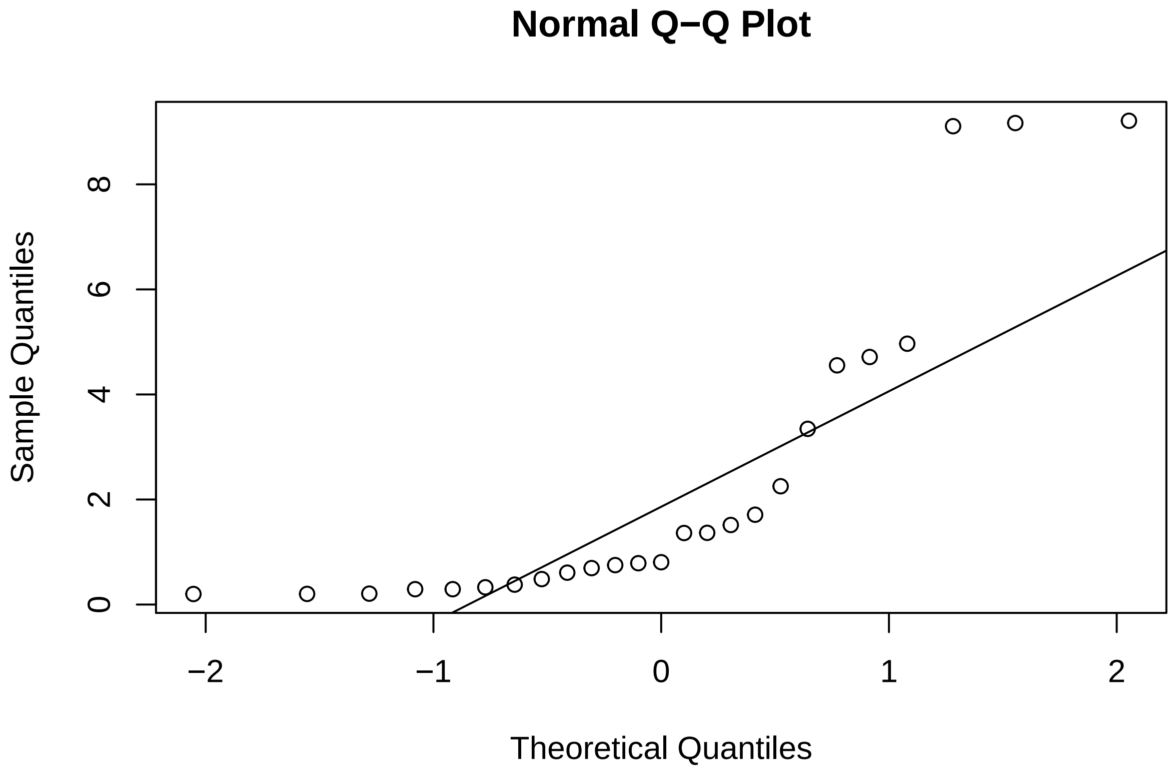
Assumption testing

1.Check if the dependent variable is normally distributed:

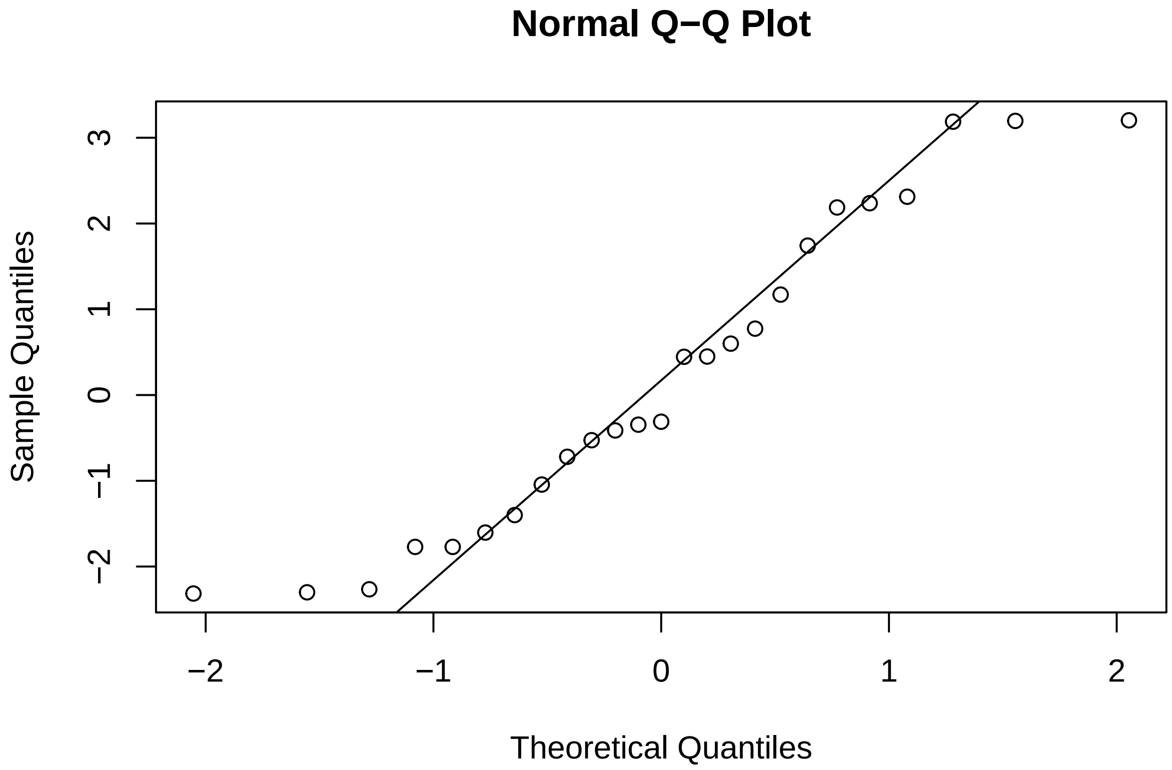
```
hist(Inverter$TransientPt) #Not normal
```



```
#QQ-plot to check the normality
qqnorm(Inverter$TransientPt); qqline(Inverter$TransientPt) #NOT normal; try transformation
```



```
#looks improved with log transformation
qqnorm(log2(Inverter$TransientPt)); qqline(log2(Inverter$TransientPt))
```



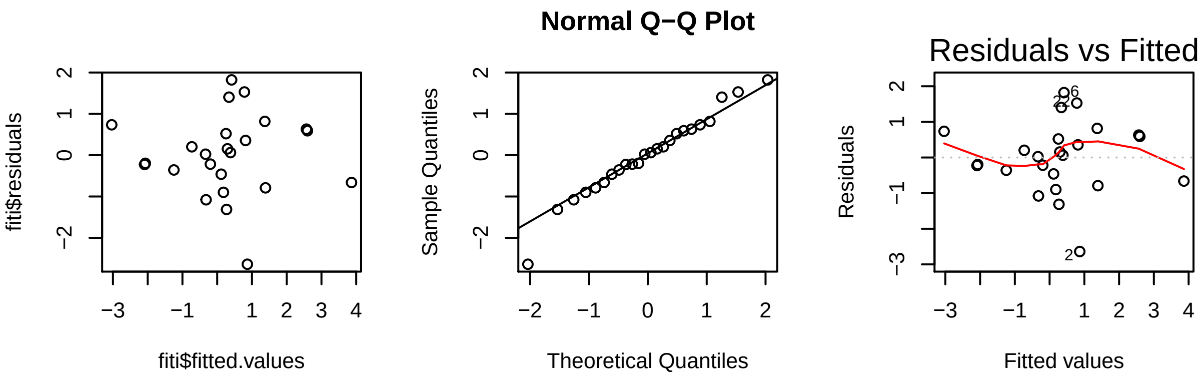
Change the starting model for regression with the log transformed dependent variable

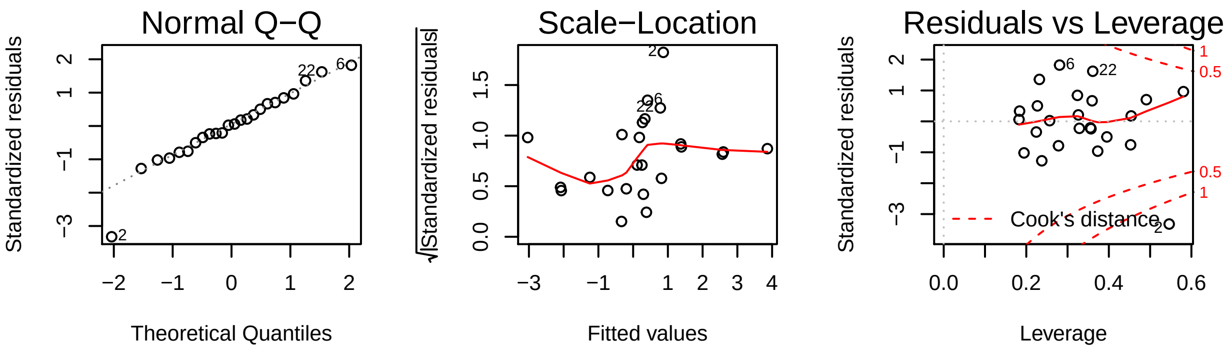
```
fiti=lm(log2(TransientPt) ~ ., data=Inverter); summary(fiti)
```

```
##
## Call:
## lm(formula = log2(TransientPt) ~ ., data = Inverter)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.63923 -0.51104  0.04286  0.60057  1.82170
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.62182    1.15518   0.538  0.59778
## WidthNMOS   -0.21495    0.06479  -3.318  0.00435 **
## LengthNMOS   0.11414    0.04039   2.826  0.01217 *
## WidthPMOS    0.30464    0.11485   2.652  0.01738 *
## LengthPMOS  -0.37269    0.12055  -3.092  0.00700 **
## Setpoint25   0.79652    0.81564   0.977  0.34332
## Setpoint50   0.35829    0.75683   0.473  0.64232
## Setpoint75   0.02917    0.75424   0.039  0.96963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.179 on 16 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.7053, Adjusted R-squared:  0.5764
## F-statistic: 5.471 on 7 and 16 DF,  p-value: 0.00237
```

2.Residual plots

```
# residuals vs fitted values
par(mfrow = c(2, 3))
plot(fiti$residuals ~ fiti$fitted.values) # nonlinear trend qqnorm(fiti$res)
qqnorm(fiti$res);qqline(fiti$res);
plot(fiti)
```





The residuals does not look randomly distributed in the residual vs fitted plot. Further investigation will be conducted.

3. Check for outliers

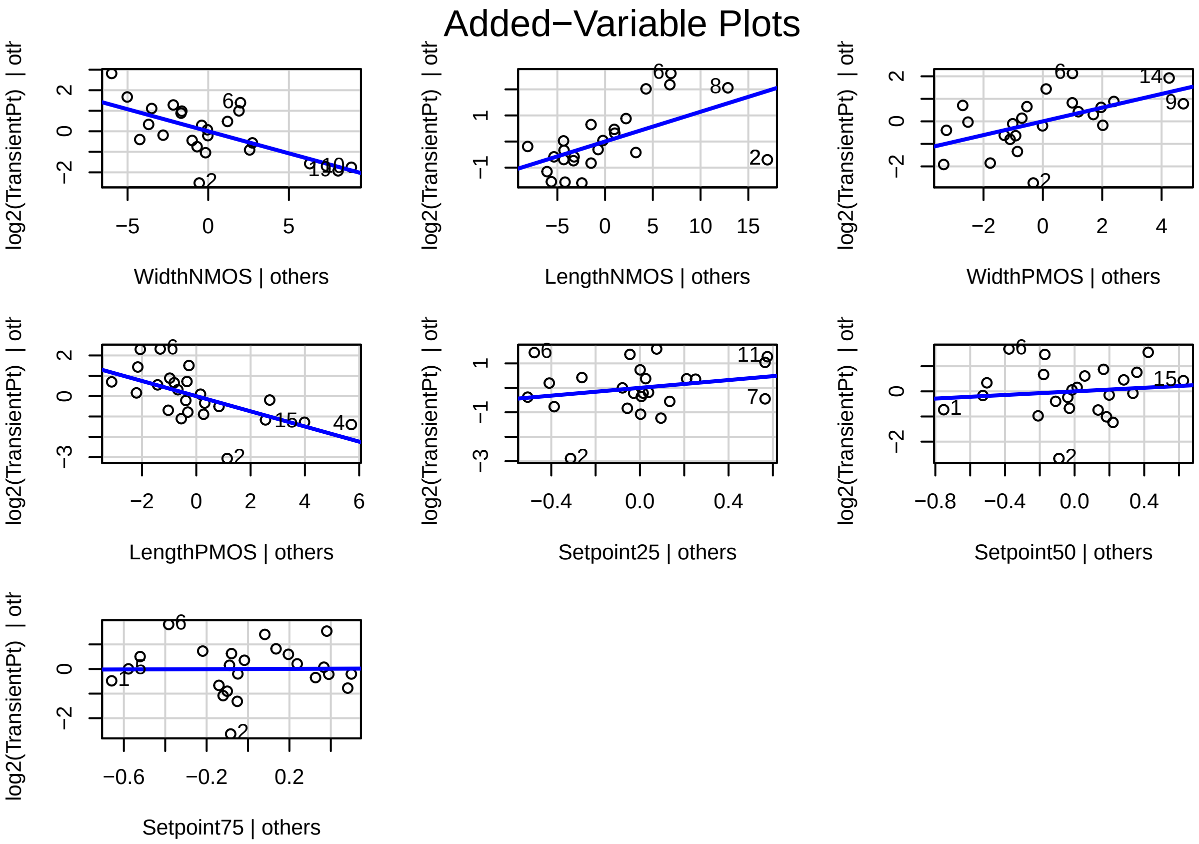
```
outlierTest(fiti) #Looks like the second record is an outlier.
```

```
##    rstudent unadjusted p-value Bonferonni p
## 2 -5.776079         3.6579e-05    0.0008779
```

```
#Further investigation will be conducted
```

4. Influential observation detection with Cook's D and leverage obs. with hat values

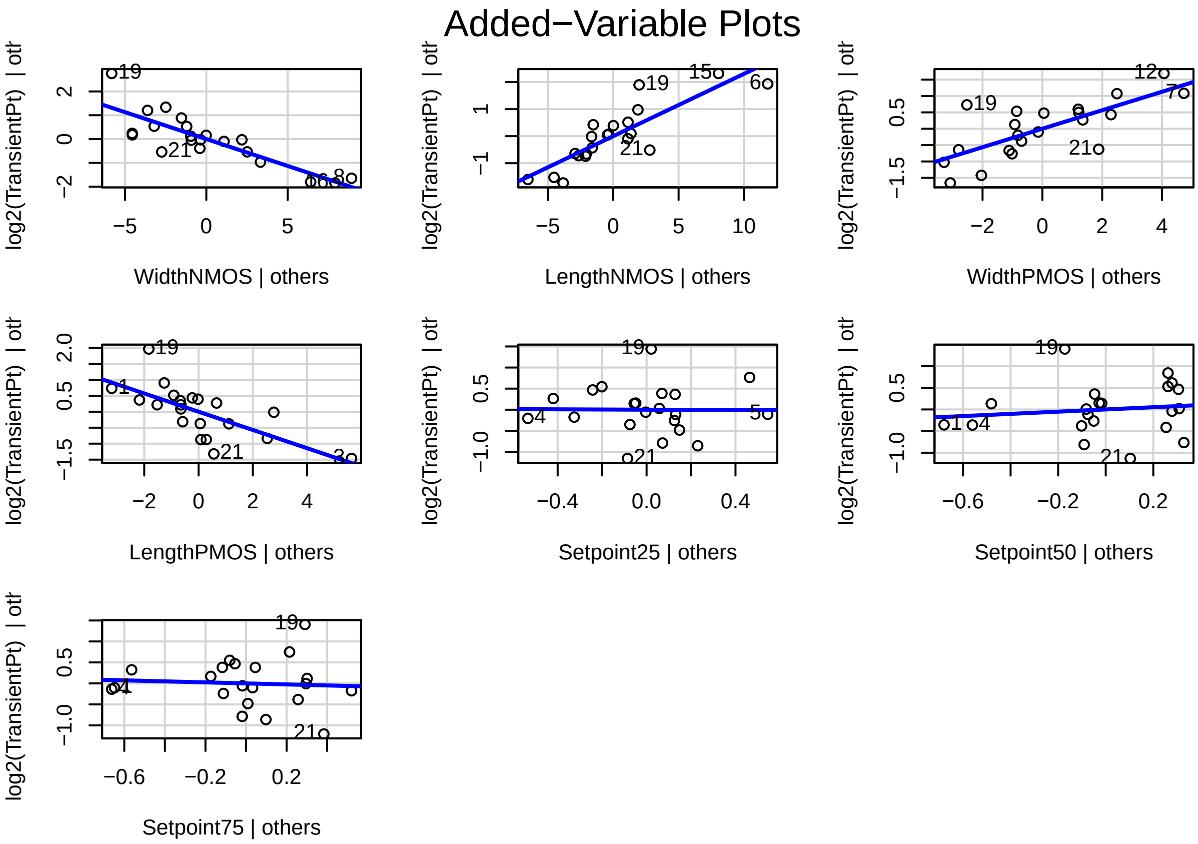
```
avPlots(fiti)
```



```r
any(cooks.distance(fiti) > 1) #there are influential observations based on cook's D
```

```
## [1] TRUE
```

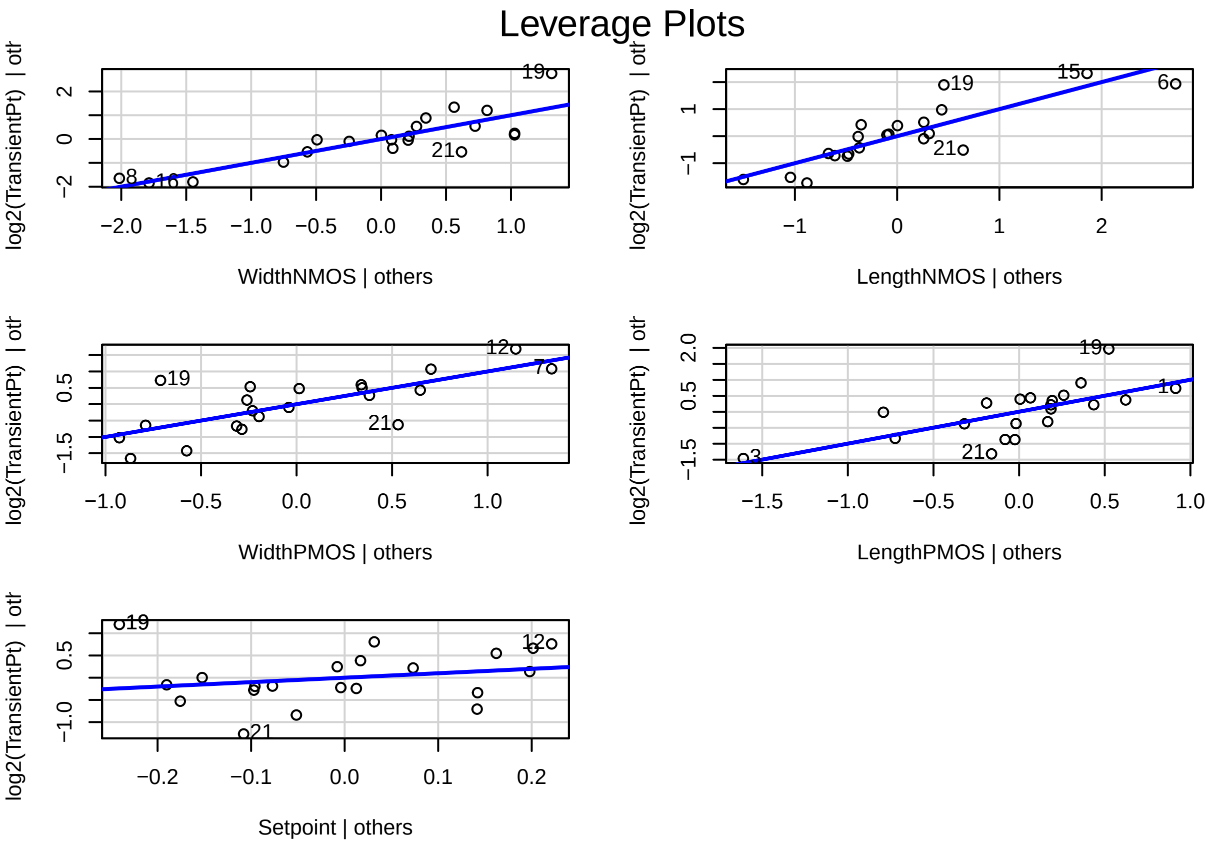
```r
#observations 2, 6, and 15 seem to be influential points; remove them and check the model
Inv2 <- Inverter[-c(2,6,15),]
fitii <- lm(log2(TransientPt)~.,data=Inv2)
avPlots(fitii)
```



```
any(cooks.distance(fitii) > 1)
```

```
## [1] TRUE
```

```
#any observations 2-3 times greater than the average hat value to be considered as "leverage" observati
#hatvalues(fiti)
leveragePlots(fitii)
```



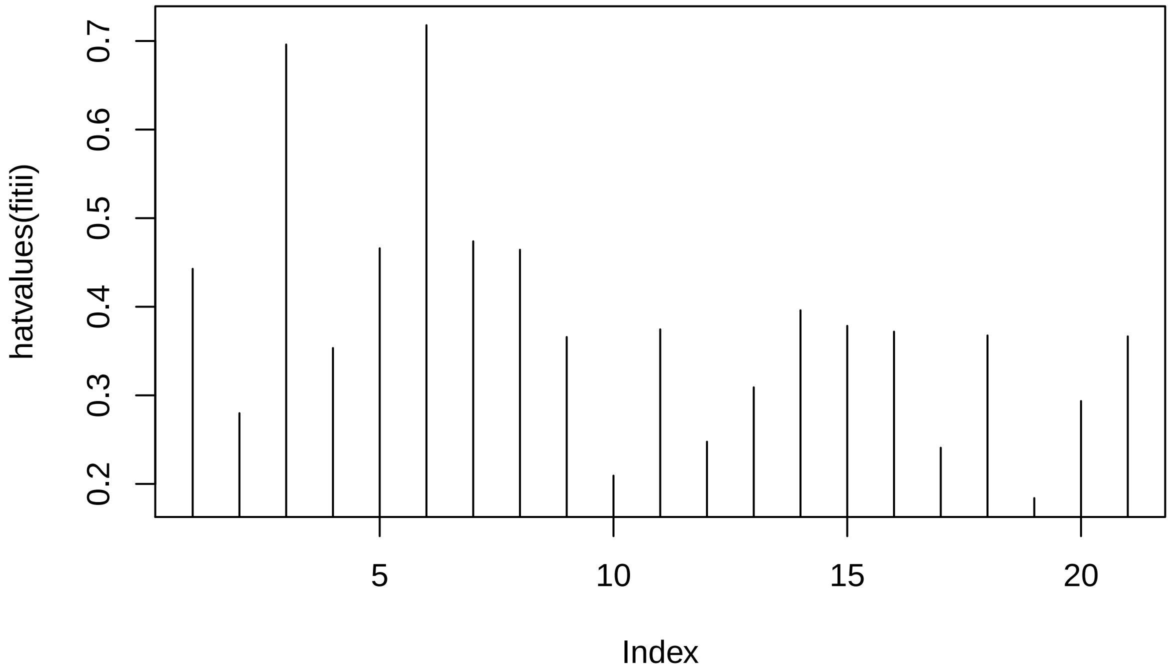
```
hv <- as.data.frame(hatvalues(fitii))
mn <-mean(hatvalues(fitii))
hv$warn <- ifelse(hv[, 'hatvalues(fitii)']>3*mn, 'x3',
   ifelse(hv[, 'hatvalues(fitii)']>2*mn, 'x3', '-' ))
subset(hv, warn=="x3")
```

```
## [1] hatvalues(fitii) warn
## <0 rows> (or 0-length row.names)
```

```
subset(hv, warn%in%c("x2", "x3"))
```

```
## [1] hatvalues(fitii) warn
## <0 rows> (or 0-length row.names)
```

```
plot(hatvalues(fitii), type = "h") #no leverage point
```



```
#Check again for the outliers
outlierTest(fitii) #The model has been cleaned from the outliers.
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 19 3.348449          0.0057968      0.12173
```

5. Check for multicollinearity

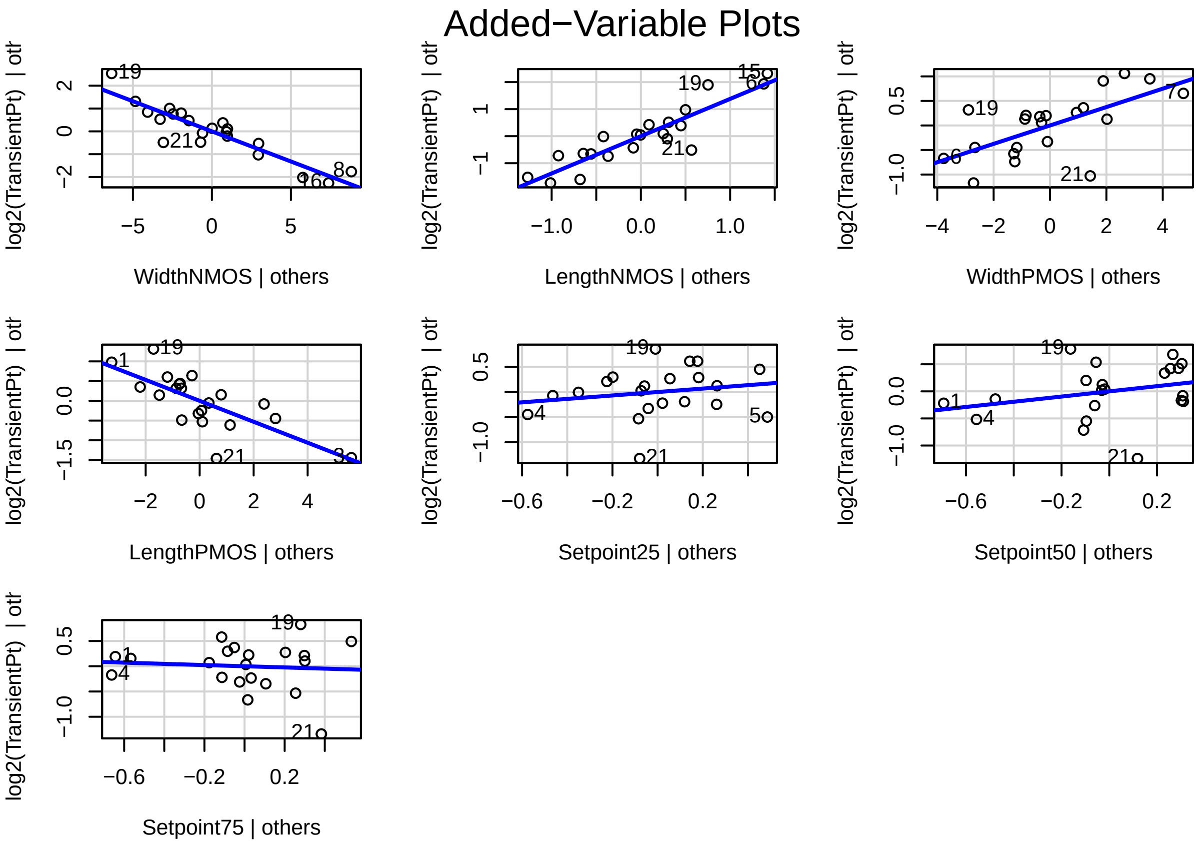
```
#Check the VIF and see if any VIF value is greater than 10
vif(fiti) #No VIF is larger than 10. Thus, ridge or PCA will not be used.
```

```
##                GVIF Df GVIF^(1/(2*Df))
## WidthNMOS  1.279151  1        1.130996
## LengthNMOS 1.297835  1        1.139225
## WidthPMOS  1.561659  1        1.249663
## LengthPMOS 1.494008  1        1.222296
## Setpoint   1.957585  3        1.118459
```

From the plots above, there seems to be a nonlinear pattern between the response variable and lengthNMOS. Thus, lengthNMOS will be log transformed and the distributions will be re-examined.

Fit another model with log transformed lengthNMOS

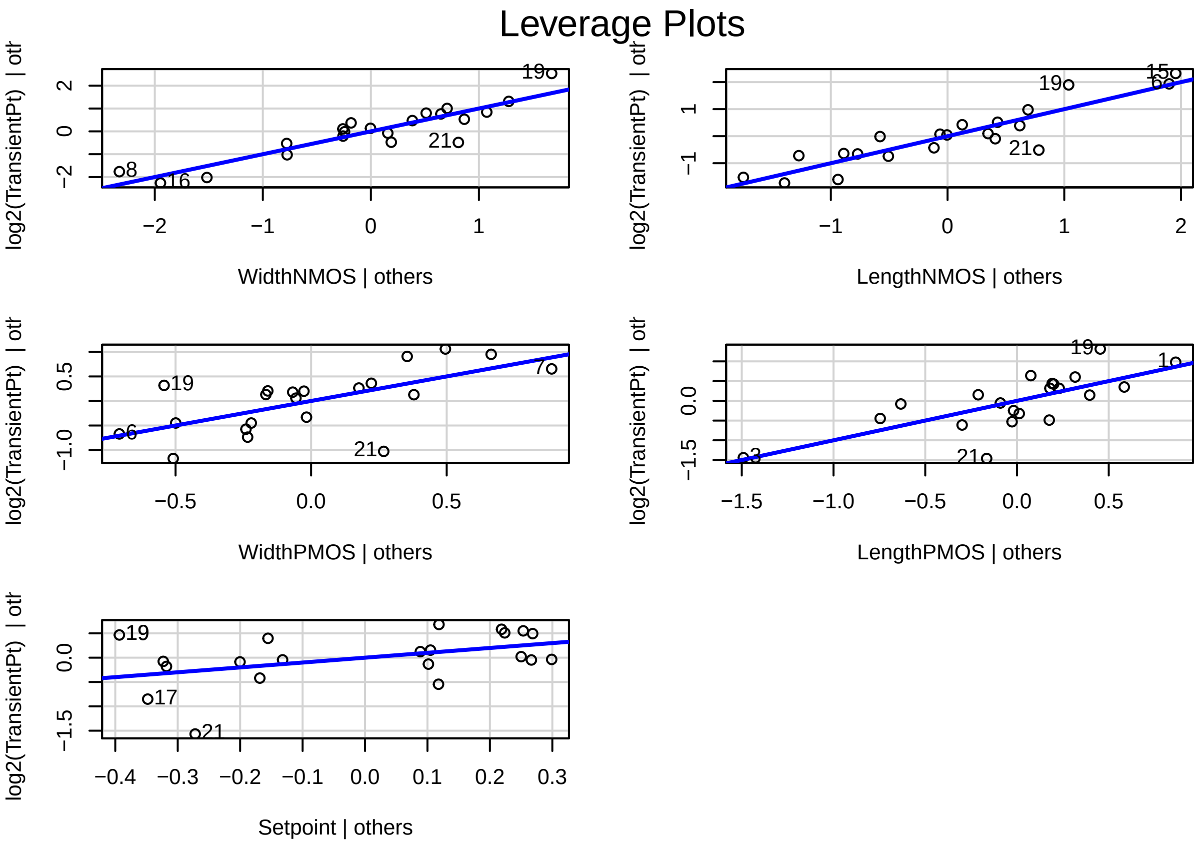
```
Inv2$LengthNMOS <- log2(Inv2$LengthNMOS)
fitiii <- lm(log2(TransientPt)~.,data=Inv2)
avPlots(fitiii)
```



```r
any(cooks.distance(fitiii) > 1)
```

```
## [1] FALSE
```

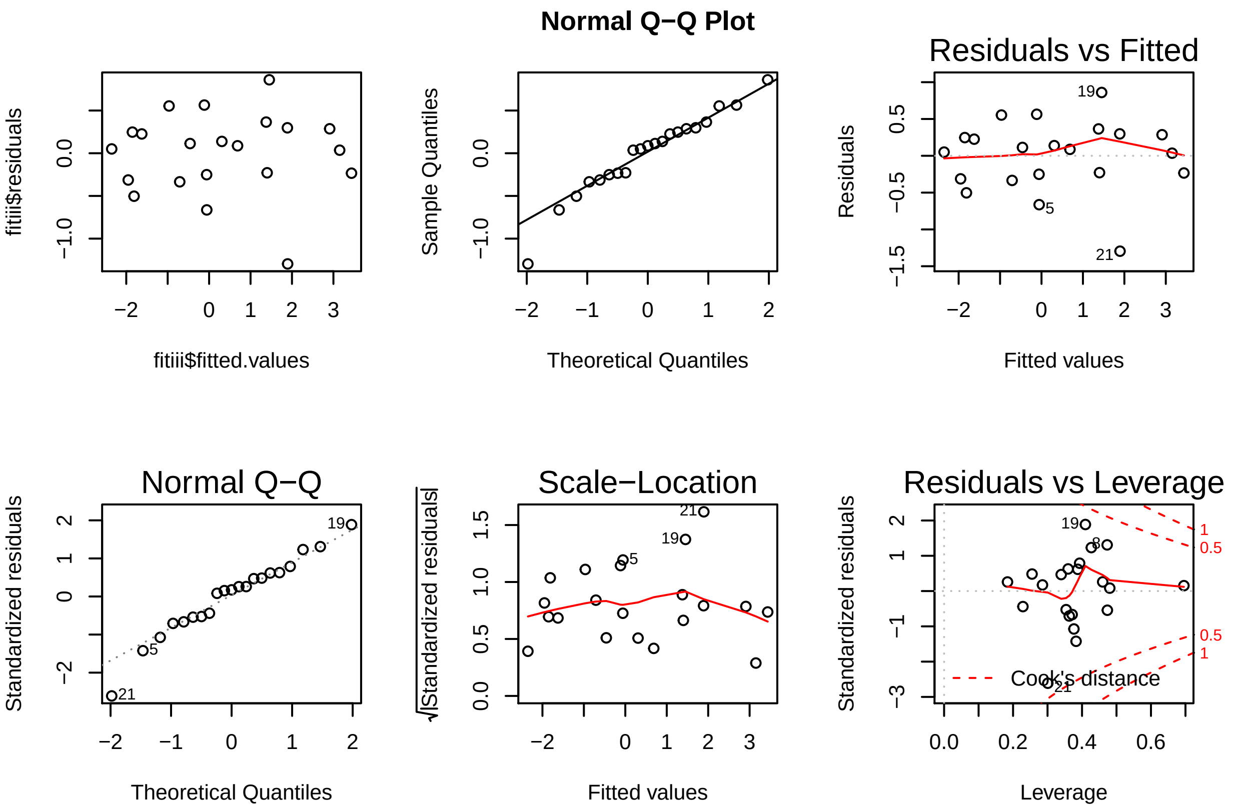
```r
#any observations 2-3 times greater than the average hat value to be considered as "leverage" observati
#hatvalues(fiti)
leveragePlots(fitiii)
```



```
hv <- as.data.frame(hatvalues(fitiii))
mn <-mean(hatvalues(fitiii))
```

Check residuals vs fitted values for fitiii

```
par(mfrow = c(2, 3))
plot(fitiii$residuals ~ fitiii$fitted.values)
qqnorm(fitiii$res);qqline(fitiii$res);
plot(fitiii)
```



Now, the diagnostic plots look fine. Model selection can be performed.

Model Selection

Stepwise model selection with AIC will be employed. Further examinations will be reported with SAS outputs.

```
summary(selectedMod)
```

```
##
## Call:
## lm(formula = log2(TransientPt) ~ WidthNMOS + LengthNMOS + WidthPMOS +
##     LengthPMOS, data = Inv2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56793 -0.13217 -0.03572  0.46670  0.68247
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.36555    0.52008  -2.626 0.018358 *
## WidthNMOS   -0.26486    0.03389  -7.815 7.50e-07 ***
## LengthNMOS   1.40884    0.17013   8.281 3.53e-07 ***
## WidthPMOS    0.19029    0.05664   3.360 0.003983 **
## LengthPMOS  -0.30079    0.06154  -4.888 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5998 on 16 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.9138, Adjusted R-squared:  0.8923
## F-statistic: 42.41 on 4 and 16 DF,  p-value: 2.53e-08
```

Final model: log2(TransientPt) ~ WidthNMOS + LengthNMOS + WidthPMOS + LengthPMOS + Setpoint