Ch 8 - Batter Cell Example

Adam McQuistan

Sunday, May 01, 2016

The dataset

Variable	Description
Y	Number of discharge cycles in battery
X1	Quantitative variable. Charge rate (ampers)
X2	Quantitative variable. Temperature (Celcius)
X_1	Coded variable. (X1 - 1) / 0.4
X_2	Coded variable. (X2 - 20) / 10
X_1^2	Coded variable.
X_2^2	Coded variable.
X_{12}	Coded variable.

```
lines <- readLines("CH08TA01.txt")
Y <- vector(); X1 <- vector(); X2 <- vector()

library(gdata)
for(line in lines){
  line <- trim(line)
  lineAry <- unlist(strsplit(line, "  "))
  Y <- c(Y, lineAry[1])
  X1 <- c(X1, lineAry[2])
  X2 <- c(X2, lineAry[3])
}

Y <- as.numeric(Y); X1 <- as.numeric(X1); X2 <- as.numeric(X2)
X_1 <- (X1 - 1) / 0.4; X_2 <- (X2 - 20) / 10
X_sq1 <- X_1^2; X_sq2 <- X_2^2; X_12 <- X_1 * X_2

df <- data.frame(Y,X1,X2,X_1,X_2,X_sq1,X_sq2,X_12)
str(df)
```

```
## 'data.frame':    11 obs. of  8 variables:
##  $ Y    : num  150 86 49 288 157 131 184 109 279 235 ...
##  $ X1   : num  0.6 1 1.4 0.6 1 1 1 1.4 0.6 1 ...
##  $ X2   : num  10 10 10 20 20 20 20 20 30 30 ...
##  $ X_1  : num  -1 0 1 -1 0 0 0 1 -1 0 ...
##  $ X_2  : num  -1 -1 -1 0 0 0 0 0 1 1 ...
##  $ X_sq1: num  1 0 1 1 0 0 0 1 1 0 ...
##  $ X_sq2: num  1 1 1 0 0 0 0 0 1 1 ...
##  $ X_12 : num  1 0 -1 0 0 0 0 0 -1 0 ...
```

Correlation Among the Predictor Terms

```r
library(knitr)
Terms = c("X1 and X1 Sqr","x_1 and x_1 Sqr", "X2 and X2 Sqr", "x_2 and x_2 Sqr")
Correlations = c(cor(df$X1, df$X1^2), cor(df$X_1, df$X_sq1),
                 cor(df$X2, df$X2^2), cor(df$X_2, df$X_sq2))
Correlations <- round(Correlations, 3)
cor_df <- data.frame(Terms, Correlations)
kable(cor_df)
```

Terms	Correlations
X1 and X1 Sqr	0.991
x_1 and x_1 Sqr	0.000
X2 and X2 Sqr	0.986
x_2 and x_2 Sqr	0.000

Note that recoding the variables reduced the correlation among terms to zero.

Fitting the Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2 + \varepsilon$$

```r
result1 <- lm(Y ~ X_1 + X_2 + X_sq1 + X_sq2 + X_12, data=df)
result1_smry <- summary(result1); print(result1_smry)
```

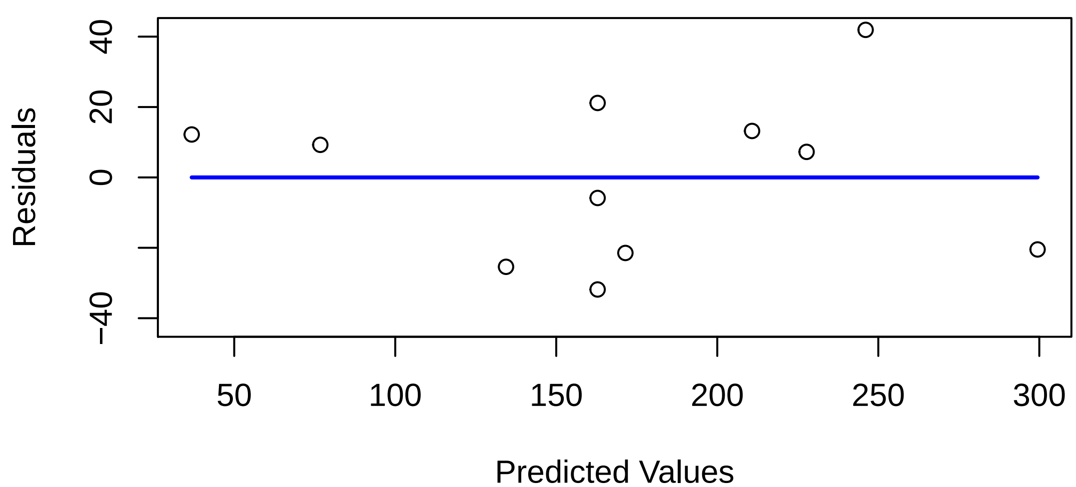
```
##
## Call:
## lm(formula = Y ~ X_1 + X_2 + X_sq1 + X_sq2 + X_12, data = df)
##
## Residuals:
##       1       2       3       4       5       6       7       8       9
## -21.465   9.263  12.202  41.930  -5.842 -31.842  21.158 -25.404 -20.465
##      10      11
##   7.263  13.202
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   162.84      16.61   9.805 0.000188 ***
## X_1           -55.83      13.22  -4.224 0.008292 **
## X_2            75.50      13.22   5.712 0.002297 **
## X_sq1          27.39      20.34   1.347 0.235856
## X_sq2         -10.61      20.34  -0.521 0.624352
## X_12           11.50      16.19   0.710 0.509184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

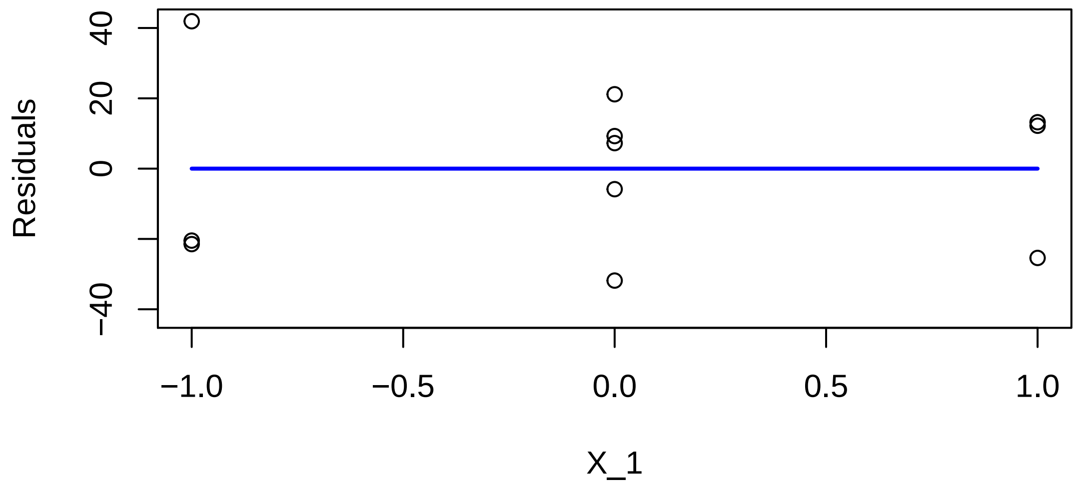
```
## Residual standard error: 32.37 on 5 degrees of freedom
## Multiple R-squared:  0.9135, Adjusted R-squared:  0.8271
## F-statistic: 10.57 on 5 and 5 DF,  p-value: 0.01086
```

```r
result1_aov <- fullRegressionAnova(anova(result1))
```

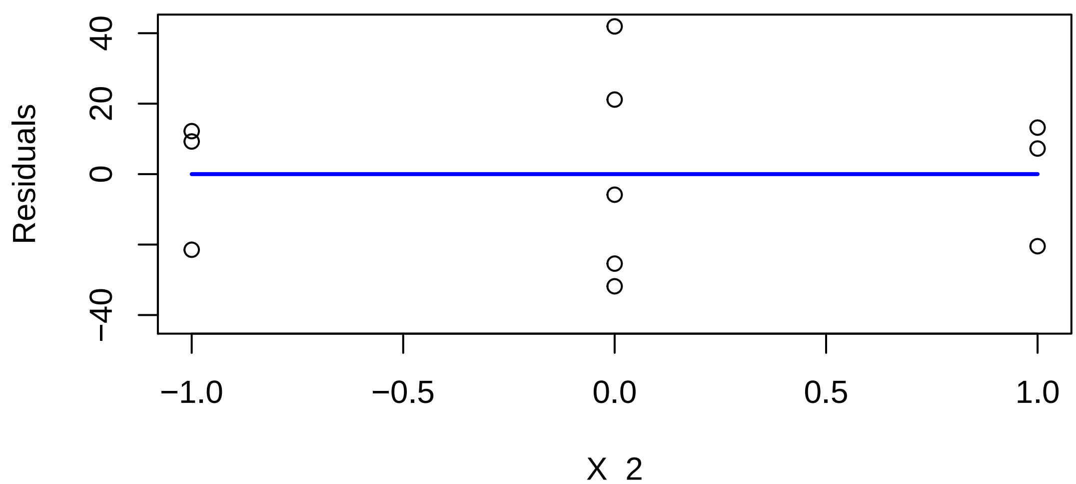
```
##   VariationSource DF         SS         MS    F_stats
## 1      Regression  5 55365.5614 11073.1123 10.5650625
## 2             X_1  1 18704.1667 18704.1667 17.8459935
## 3             X_2  1 34201.5000 34201.5000 32.6322877
## 4           X_sq1  1  1645.9667  1645.9667  1.5704474
## 5           X_sq2  1   284.9281   284.9281  0.2718552
## 6            X_12  1   529.0000   529.0000  0.5047287
## 7       Residuals  5  5240.4386  1048.0877         NA
## 8           Total 10 60606.0000         NA         NA
```

Residuals Plot

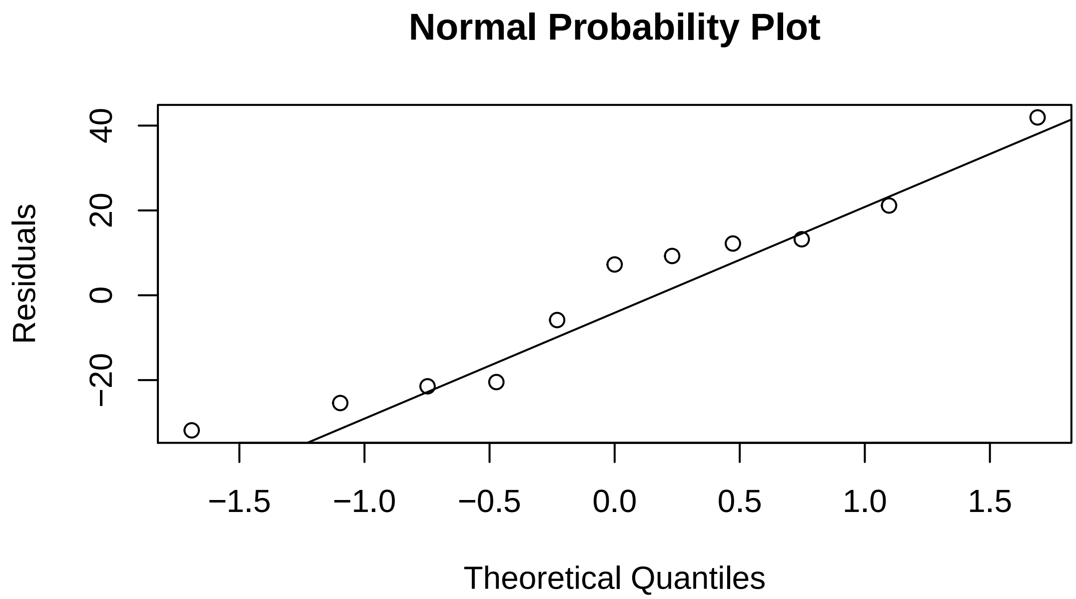




Residuals
40
20
0
−40
−1.0
−0.5
0.0
0.5
1.0
X_1



Residuals
40
20
0
−40
−1.0
−0.5
0.0
0.5
1.0
X_2



Assessing Fit

```
library(dplyr)
df_smry <- tbl_df(df) %>% group_by(X_1, X_2, X_sq1, X_sq2, X_12) %>% summarize(Repeats = n())
kable(as.data.frame(df_smry))
```

X_1	X_2	X_sq1	X_sq2	X_12	Repeats
-1	-1	1	1	1	1
-1	0	1	0	0	1
-1	1	1	1	-1	1
0	-1	0	1	0	1
0	0	0	0	0	3
0	1	0	1	0	1
1	-1	1	1	-1	1
1	0	1	0	0	1
1	1	1	1	1	1

Since on set of variables are replicated three times in the data a formal lack of fit test can be calculated.

$H_o : EY = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2$; a linear relationship exists

$H_a : EY \neq \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2$; there is not relationship

F stat = MSLF / MSPE

F crit = F(1-α; c-p; n-c)

If F Stat < F crit, conclude H_o

If F Stat >= F crit, conclude H_a

```
select_x1 <- df$X_1 == 0; select_x2 <- df$X_2 == 0
select_x1sqr <- df$X_sq2 == 0; select_x1sqr <- df$X_sq1 == 0
select_x2sqr <- df$X_sq2 == 0; select_x12 <- df$X_12 == 0

Y_replicates <- df$Y[select_x1 & select_x2 & select_x1sqr & select_x2sqr & select_x12]
Y_bar_replicates <- mean(Y_replicates)

SSPE <- sum((Y_replicates - Y_bar_replicates)^2)
SSE <- result1_aov$SS[length(result1_aov$SS)-1]
SSLF <- SSE - SSPE
p = 6 # the regression coefficents including intercept
categories = dim(df_smry)[1] # the count of distinct categories among the parameters
n <- dim(df)[1]

SSLF_degFreedom <- categories - p
SSPE_degFreedom <- n - categories
F_stat <- (SSLF / SSLF_degFreedom) / (SSPE / SSPE_degFreedom)
F_crit <- qf(0.95,SSLF_degFreedom,SSPE_degFreedom)

msg = paste("F stat = ", F_stat, "\nF crit = ", F_crit)
result <- ifelse(F_stat < F_crit,
                 "\nConclude Ho, a relationship exists. Its a good fit.",
                 "\nConclude Ha, there is not relationship. Its not a good fit.")
cat(msg, result, sep="")
```

```
## F stat =  1.82048976261251
## F crit =  19.1642921275113
## Conclude Ho, a relationship exists. Its a good fit.
```

First Order Model

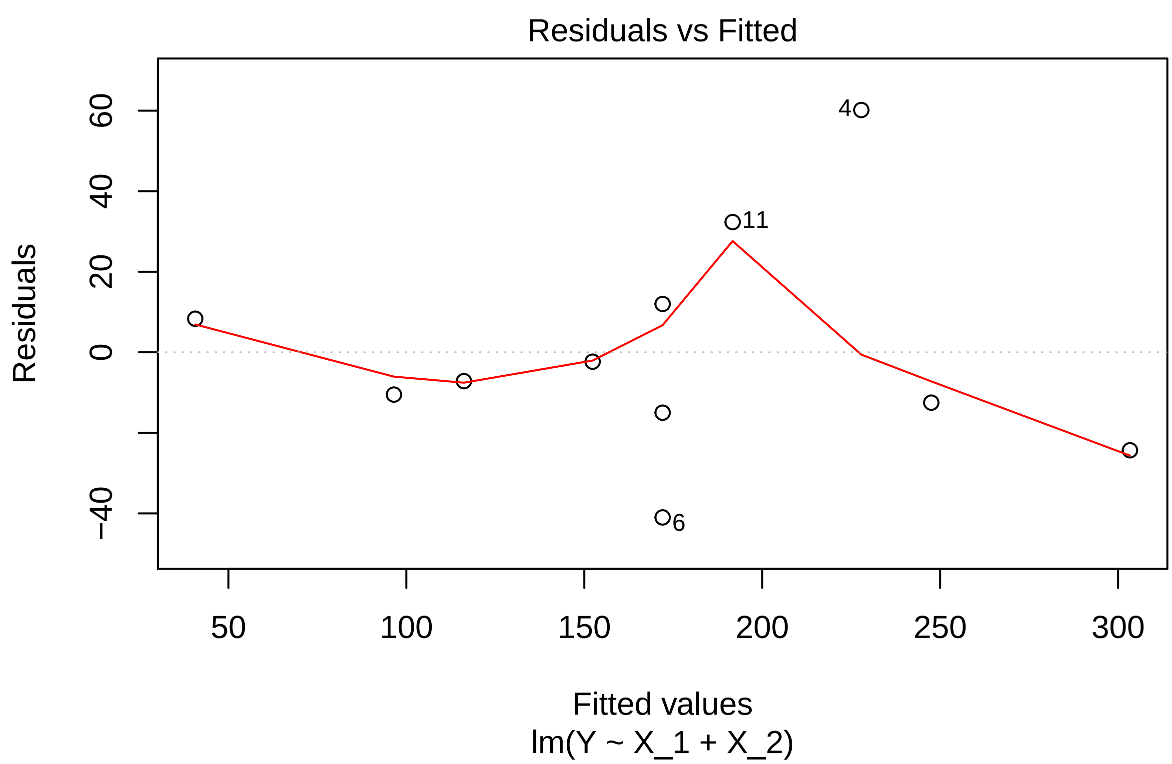
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

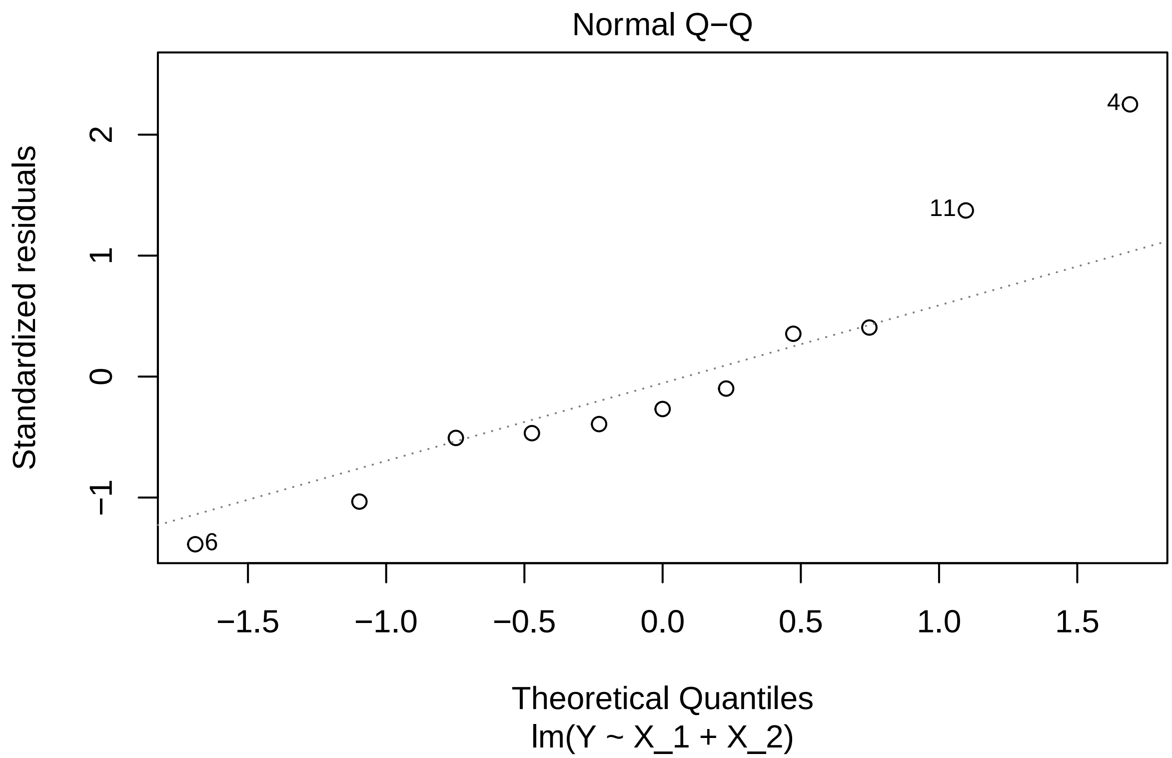
```
result2 <- lm(Y ~ X_1 + X_2, data=df)
summary(result2)
```

```
##
## Call:
## lm(formula = Y ~ X_1 + X_2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.000 -13.750  -7.167  10.167  60.167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

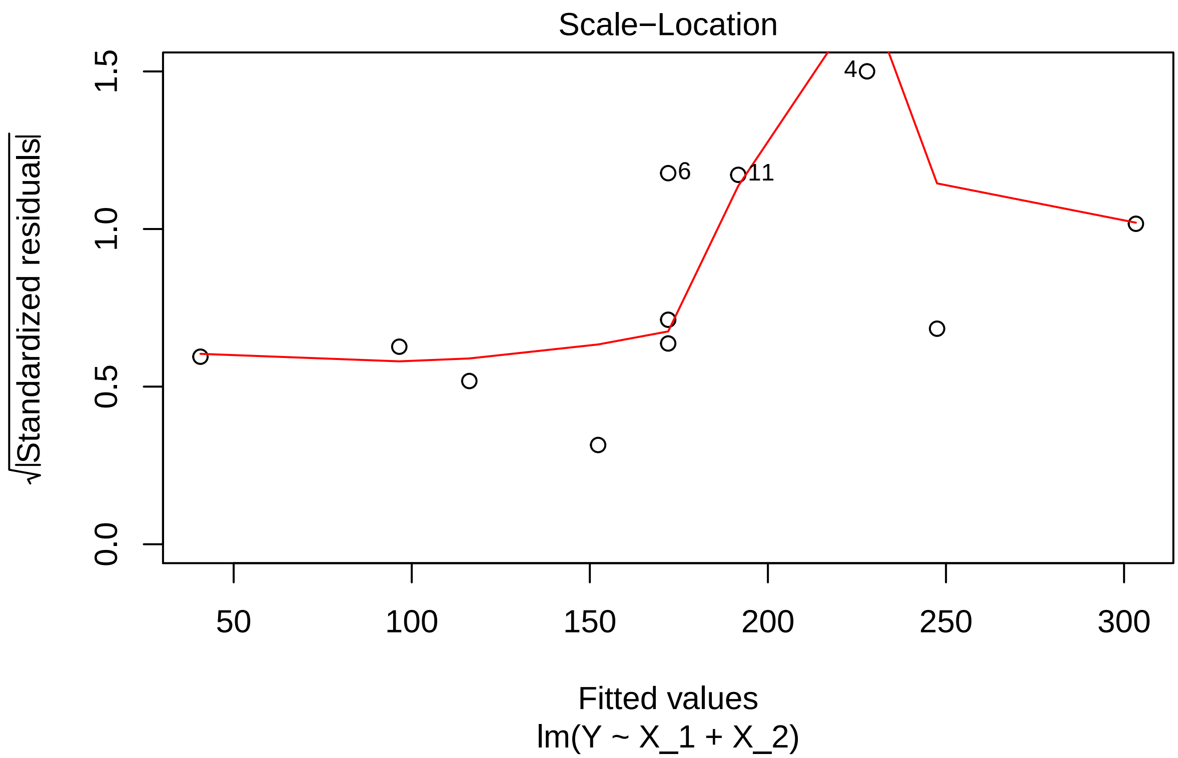
```
## (Intercept)  172.000      9.354  18.387 7.88e-08 ***
## X_1          -55.833     12.666  -4.408 0.002262 **
## X_2           75.500     12.666   5.961 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.02 on 8 degrees of freedom
## Multiple R-squared:  0.8729, Adjusted R-squared:  0.8412
## F-statistic: 27.48 on 2 and 8 DF,  p-value: 0.0002606
```

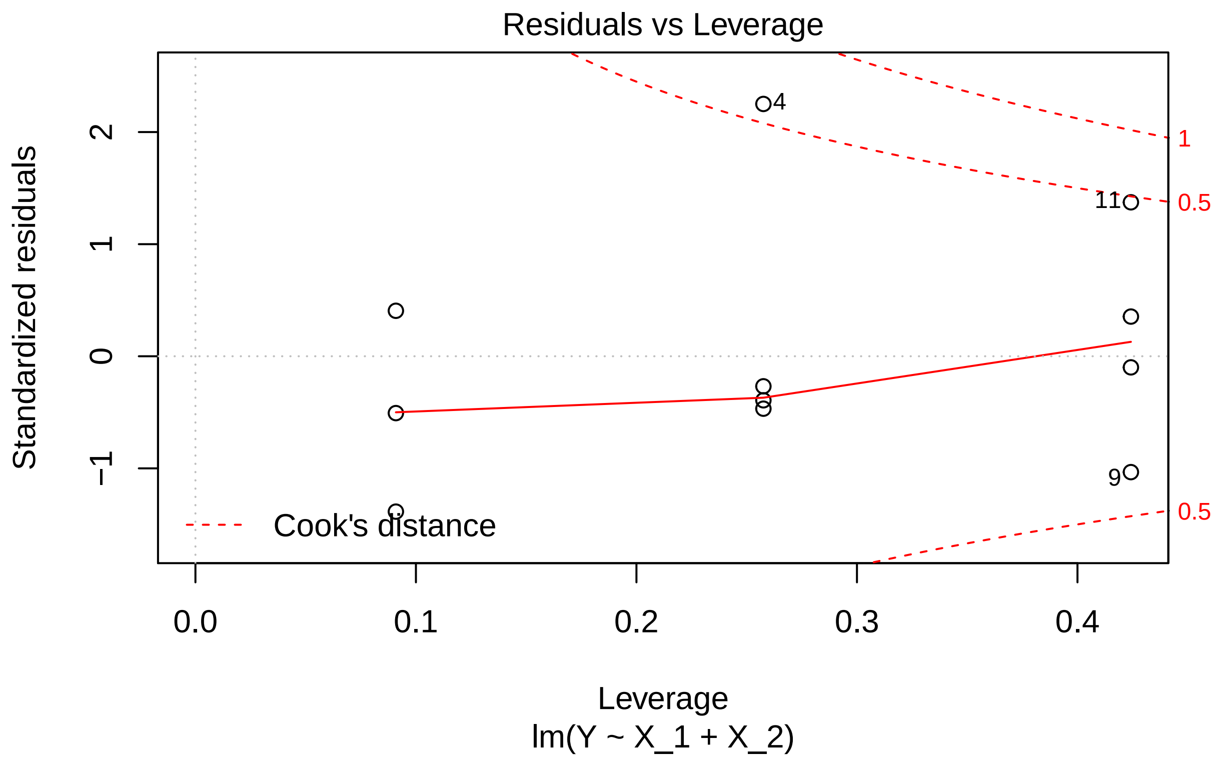
```r
plot(result2)
```





Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Y ~ X_1 + X_2)
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
4
11
6


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
50
100
150
200
250
300
Fitted values
lm(Y ~ X_1 + X_2)
6
11
4



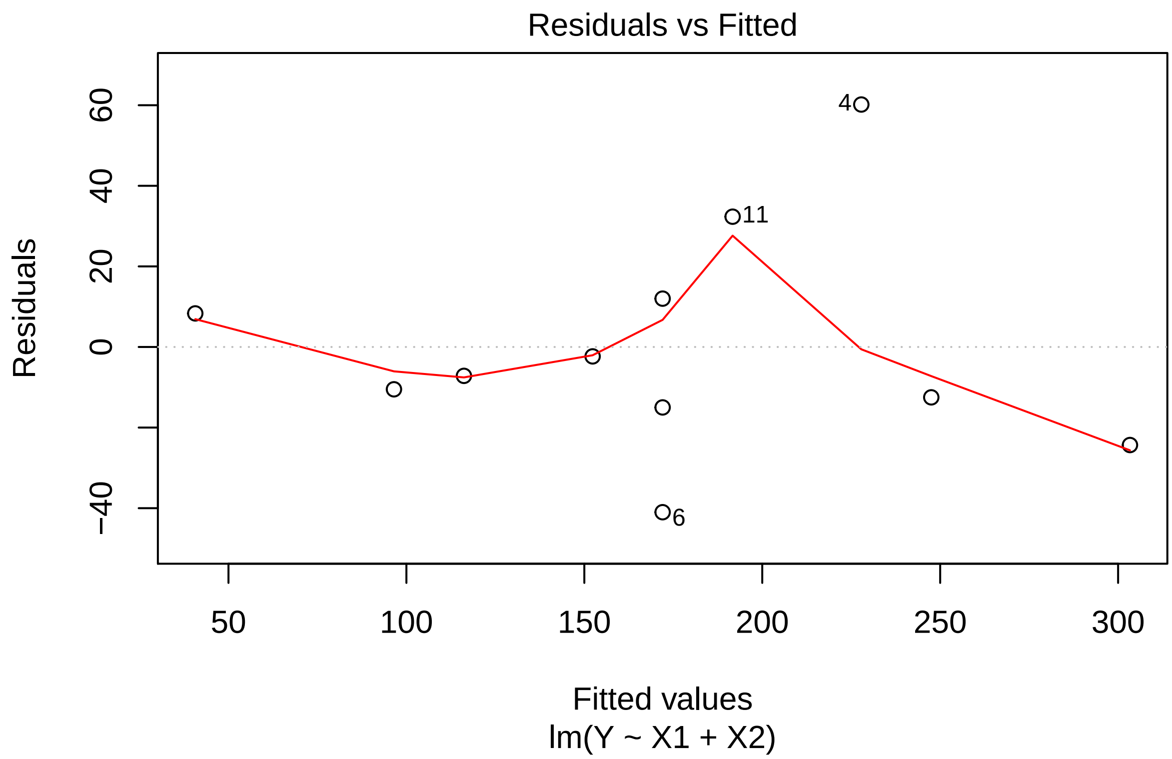
Fitted First Order in Terms of X

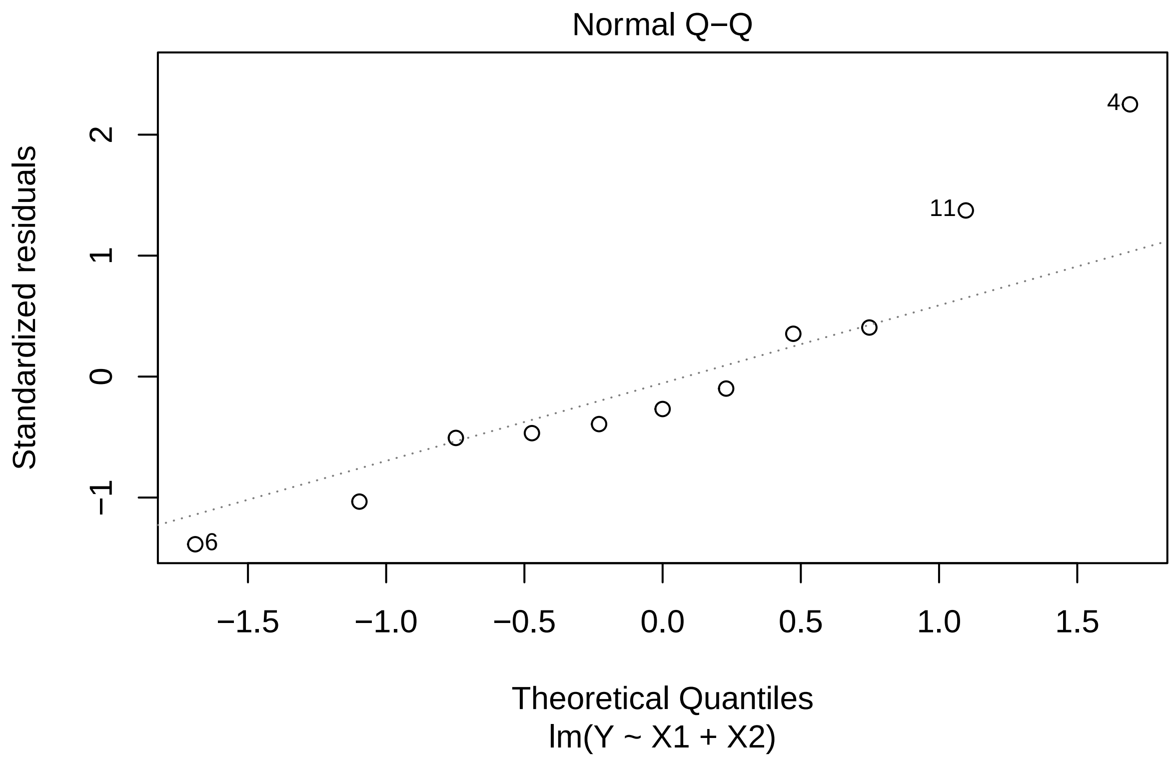
```
result3 <- lm(Y ~ X1 + X2, data=df)
summary(result3)
```

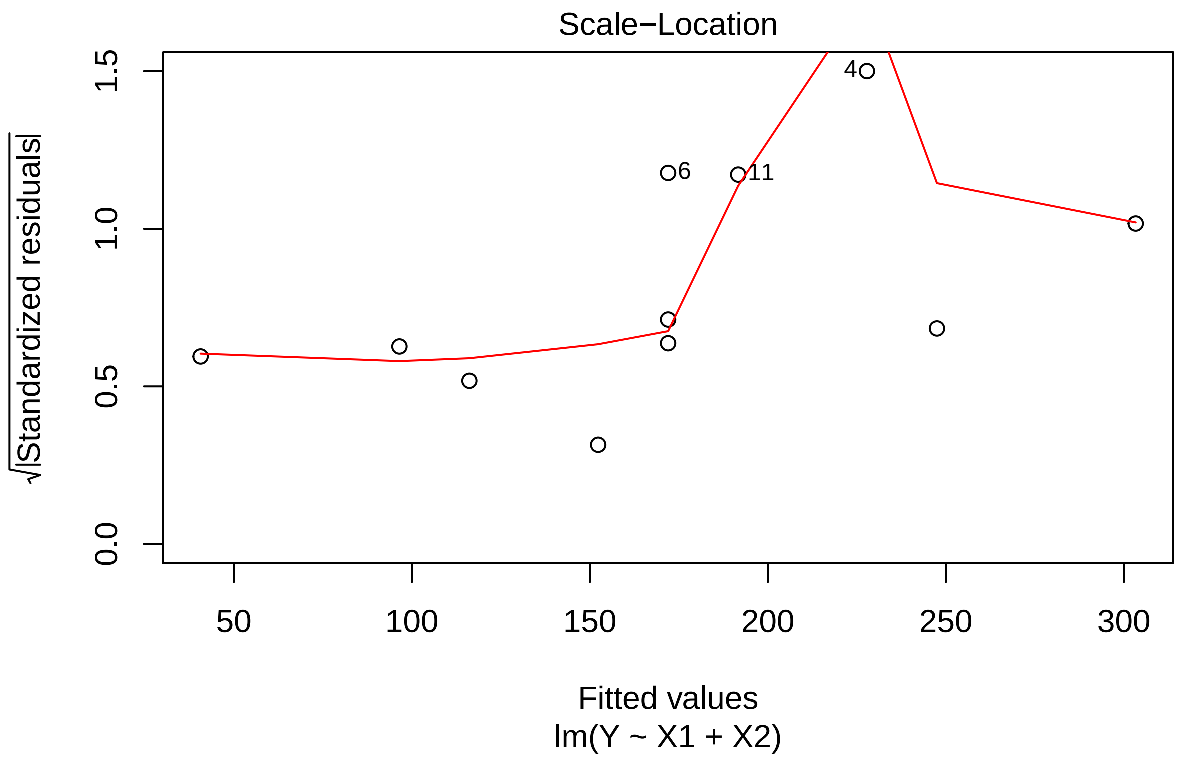
```
##
## Call:
## lm(formula = Y ~ X1 + X2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.000 -13.750  -7.167  10.167  60.167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  160.583     41.615   3.859 0.004817 **
## X1          -139.583     31.665  -4.408 0.002262 **
## X2             7.550      1.267   5.961 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.02 on 8 degrees of freedom
## Multiple R-squared:  0.8729, Adjusted R-squared:  0.8412
```

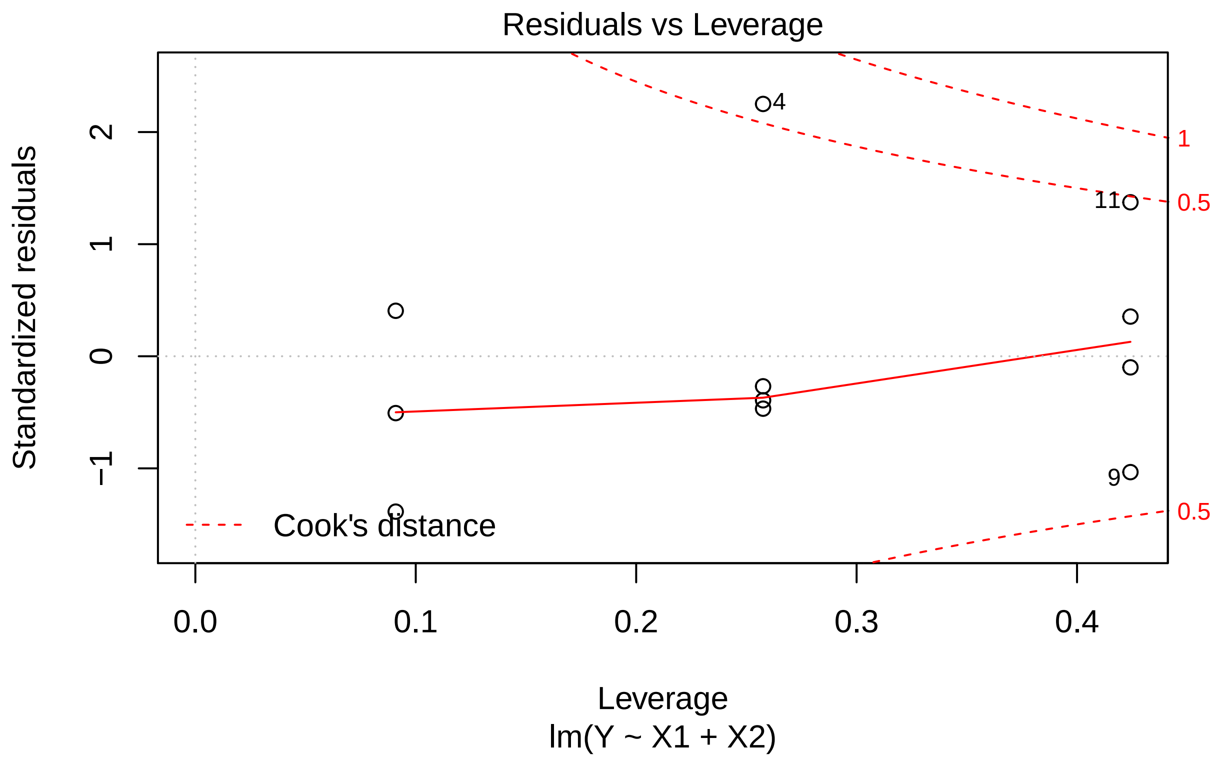
```
## F-statistic: 27.48 on 2 and 8 DF,  p-value: 0.0002606
```

```r
plot(result3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Y ~ X1 + X2)
−1
0
1
2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
4
11
6


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
50
100
150
200
250
300
Fitted values
lm(Y ~ X1 + X2)
4
6
11


Residuals vs Leverage
Standardized residuals
2
1
0
−1
4
11
9
1
0.5
0.5
Cook's distance
0.0
0.1
0.2
0.3
0.4
Leverage
lm(Y ~ X1 + X2)