Cherry Trees - Part 4

```r
# Cherry tree data is already a dataframe in R

#help(trees)
#str(trees)

#x <- trees$Girth (this is mislabeled and is really the tree diameter in inches)
#y <- trees$Volume (timber amount in cubic feet)

Diameter <- trees$Girth
trees1 <- cbind(trees, Diameter)

# 1st order polynomial

model <- lm(Volume ~ Diameter, data = trees1)
summary(model)
```

```
##
## Call:
## lm(formula = Volume ~ Diameter, data = trees1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Diameter      5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Volume
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diameter   1 7581.8  7581.8  419.36 < 2.2e-16 ***
## Residuals 29  524.3    18.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 2nd order polynomial
```

```r
model1 <- lm(Volume ~ Diameter + I(Diameter^2), data = trees1)
summary(model1)
```

```
##
## Call:
## lm(formula = Volume ~ Diameter + I(Diameter^2), data = trees1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4889 -2.4293 -0.3718  2.0764  7.6447
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    10.78627   11.22282   0.961 0.344728
## Diameter       -2.09214    1.64734  -1.270 0.214534
## I(Diameter^2)   0.25454    0.05817   4.376 0.000152 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.335 on 28 degrees of freedom
## Multiple R-squared:  0.9616, Adjusted R-squared:  0.9588
## F-statistic: 350.5 on 2 and 28 DF,  p-value: < 2.2e-16
```

```r
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: Volume
##               Df Sum Sq Mean Sq F value    Pr(>F)
## Diameter       1 7581.8  7581.8 681.766 < 2.2e-16 ***
## I(Diameter^2)  1  212.9   212.9  19.146 0.0001524 ***
## Residuals     28  311.4    11.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Compare the two models

anova(model, model1)
```

```
## Analysis of Variance Table
##
## Model 1: Volume ~ Diameter
## Model 2: Volume ~ Diameter + I(Diameter^2)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     29 524.30
## 2     28 311.38  1    212.92 19.146 0.0001524 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Model diagnostics

# Simple linear regression model

round(rstudent(model),2)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
```

```
##  1.32  0.91  0.63  0.04  0.37  0.46 -0.76 -0.14  0.79  0.03  0.94  0.05  0.14
##    14    15    16    17    18    19    20    21    22    23    24    25    26
## -0.24 -1.14 -1.52  1.31 -0.72 -1.67 -2.03  0.12 -0.78 -0.05 -1.43 -0.73  1.17
##    27    28    29    30    31
##  0.99  1.14 -0.68 -0.80  2.84
```

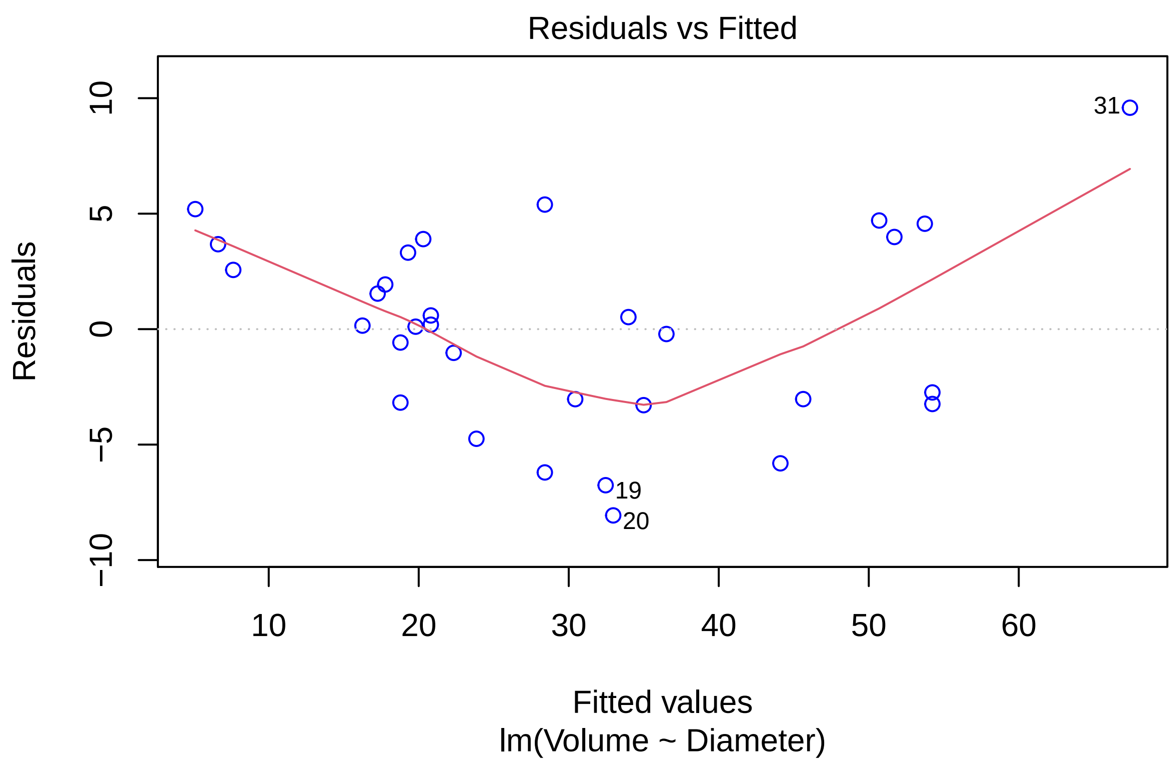
```
round(hatvalues(model),2)
```

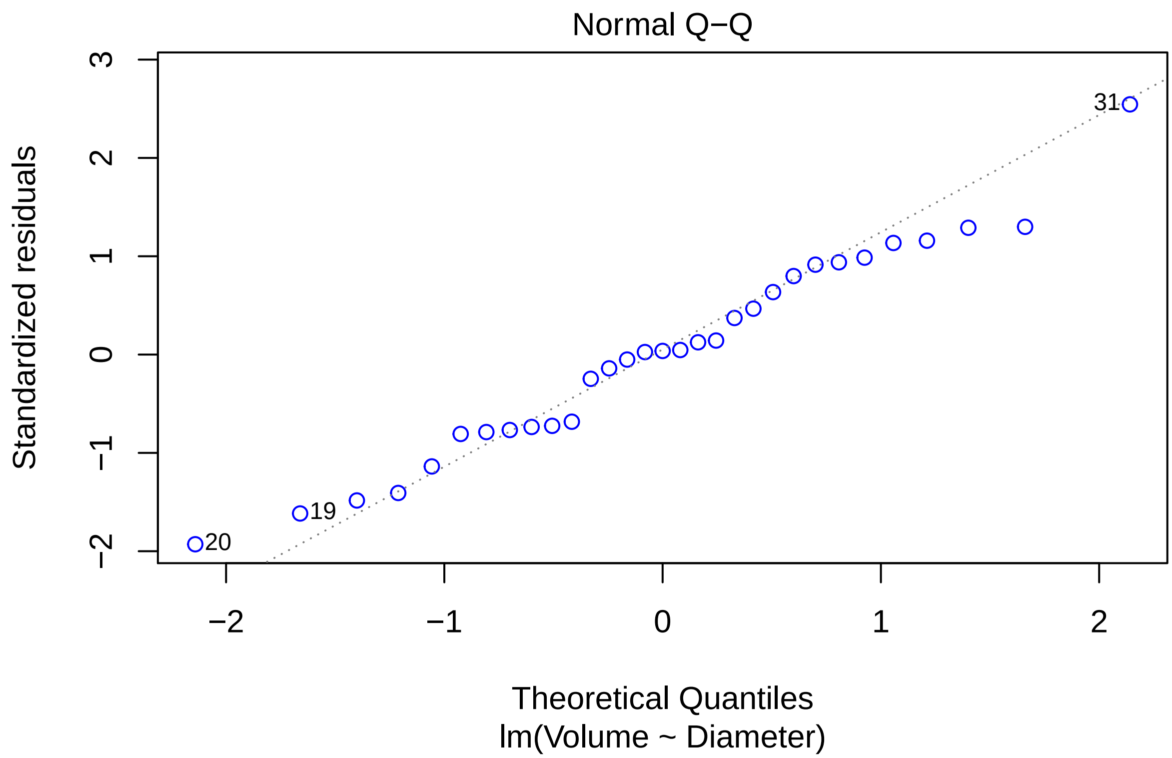
```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.12 0.11 0.10 0.06 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.04 0.04 0.04 0.04 0.03
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 0.03 0.03 0.03 0.03 0.03 0.04 0.04 0.06 0.06 0.09 0.09 0.11 0.11 0.11 0.22
```

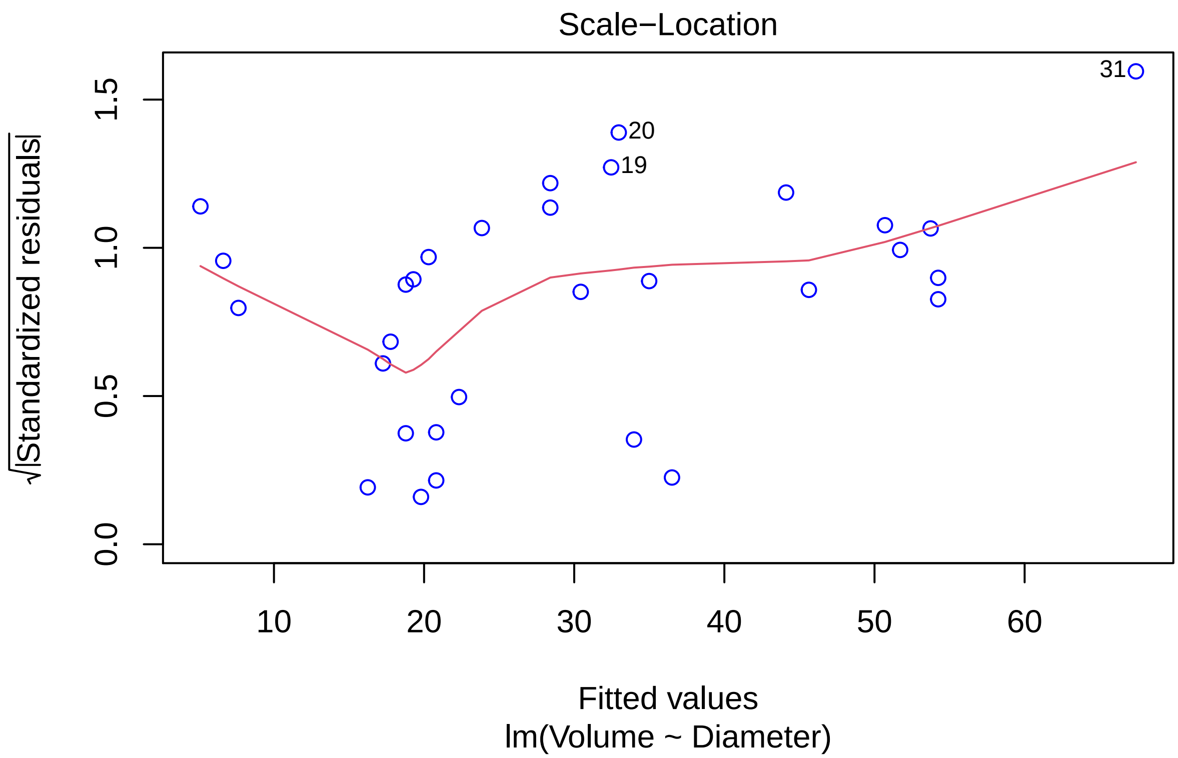
```
round(cooks.distance(model),2)
```

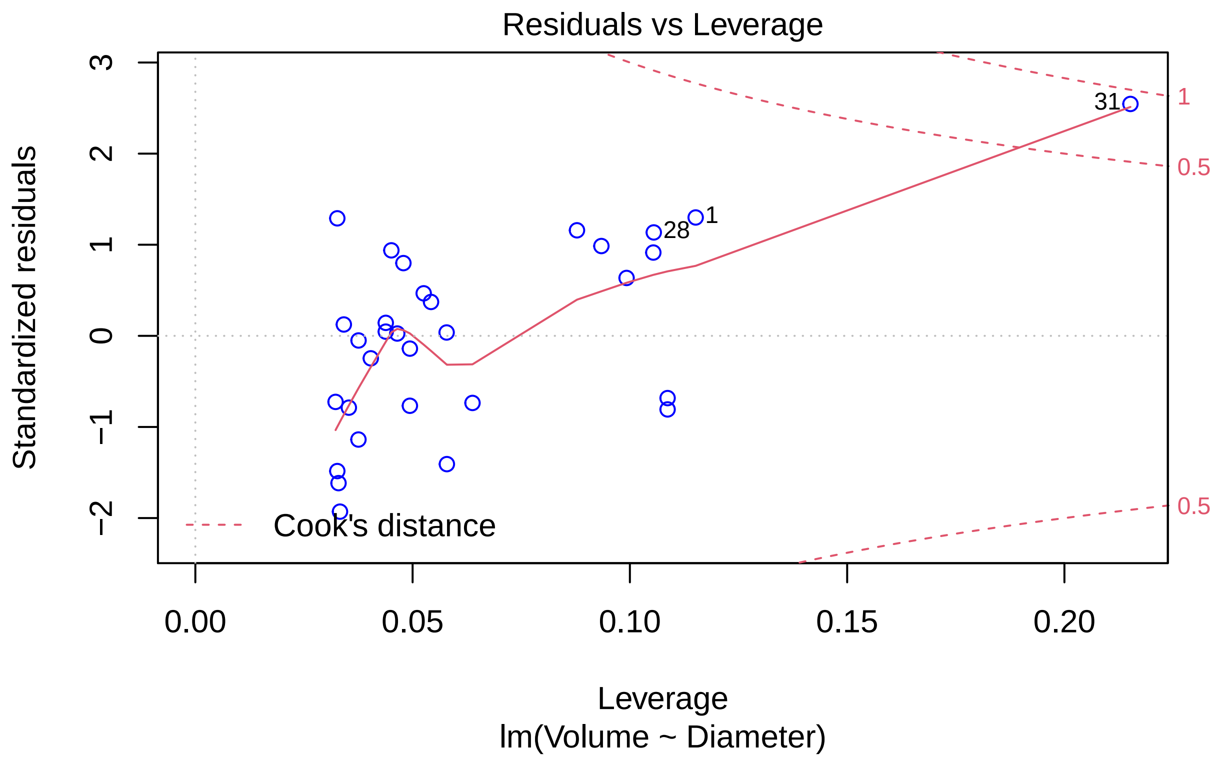
```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.11 0.05 0.02 0.00 0.00 0.01 0.02 0.00 0.02 0.00 0.02 0.00 0.00 0.00 0.03 0.04
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 0.03 0.01 0.04 0.06 0.00 0.01 0.00 0.06 0.02 0.06 0.05 0.08 0.03 0.04 0.89
```

```
#plot(model, which=c(1,2,3,4), col=c("blue"))
plot(model, col=c("blue"))
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Volume ~ Diameter)
−2
−1
0
1
2
3
20
19
31


Scale−Location
31
20
19
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
20
30
40
50
60
Fitted values
lm(Volume ~ Diameter)




```
shapiro.test(rstandard(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model)
## W = 0.97676, p-value = 0.7179
```

```
# 2nd order polynomial regression model

round(rstudent(model1),2)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## -0.23 -0.44 -0.62 -0.15  0.38  0.55 -0.91 -0.11  1.14  0.19  1.42  0.30  0.42
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##  0.04 -0.99 -1.23  2.59 -0.18 -1.32 -1.76  0.97 -0.21  0.72 -1.31 -0.53  1.48
##    27    28    29    30    31
##  1.13  1.09 -1.33 -1.50  0.56
```

```
round(hatvalues(model1),2)
```

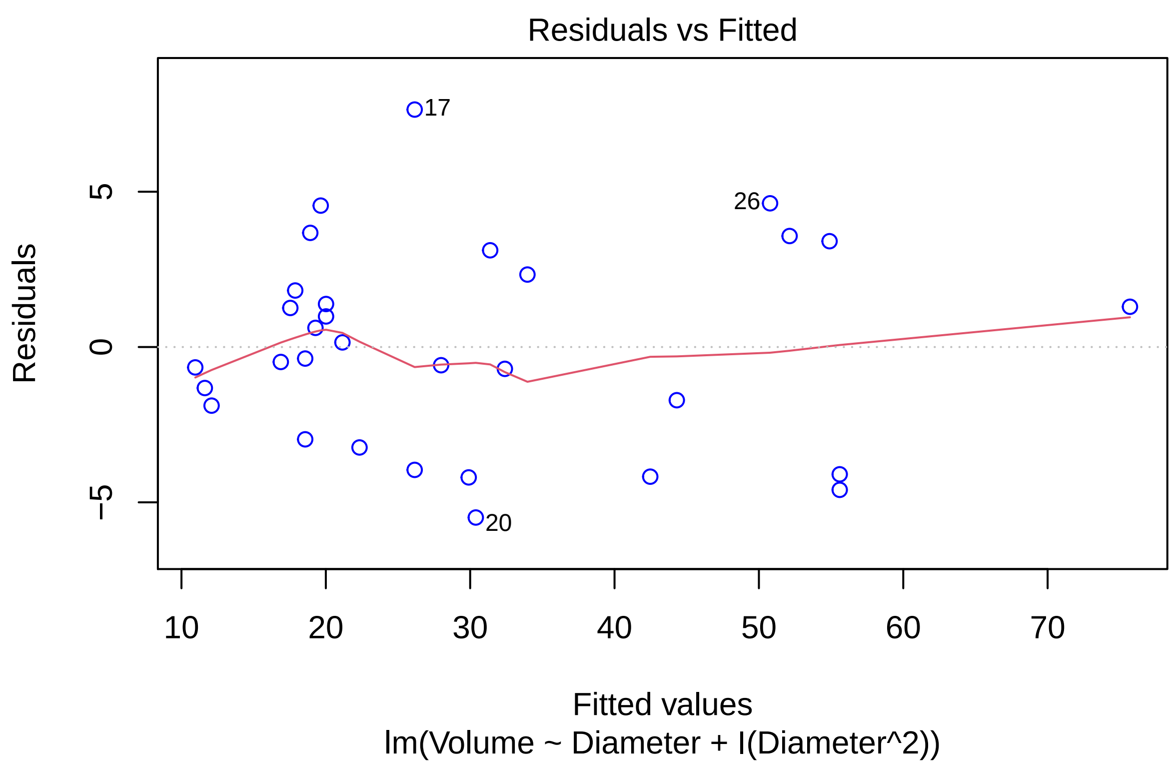
```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 0.28 0.22 0.19 0.06 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.06
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 0.06 0.06 0.06 0.06 0.07 0.07 0.07 0.07 0.07 0.09 0.09 0.11 0.12 0.12 0.54
```

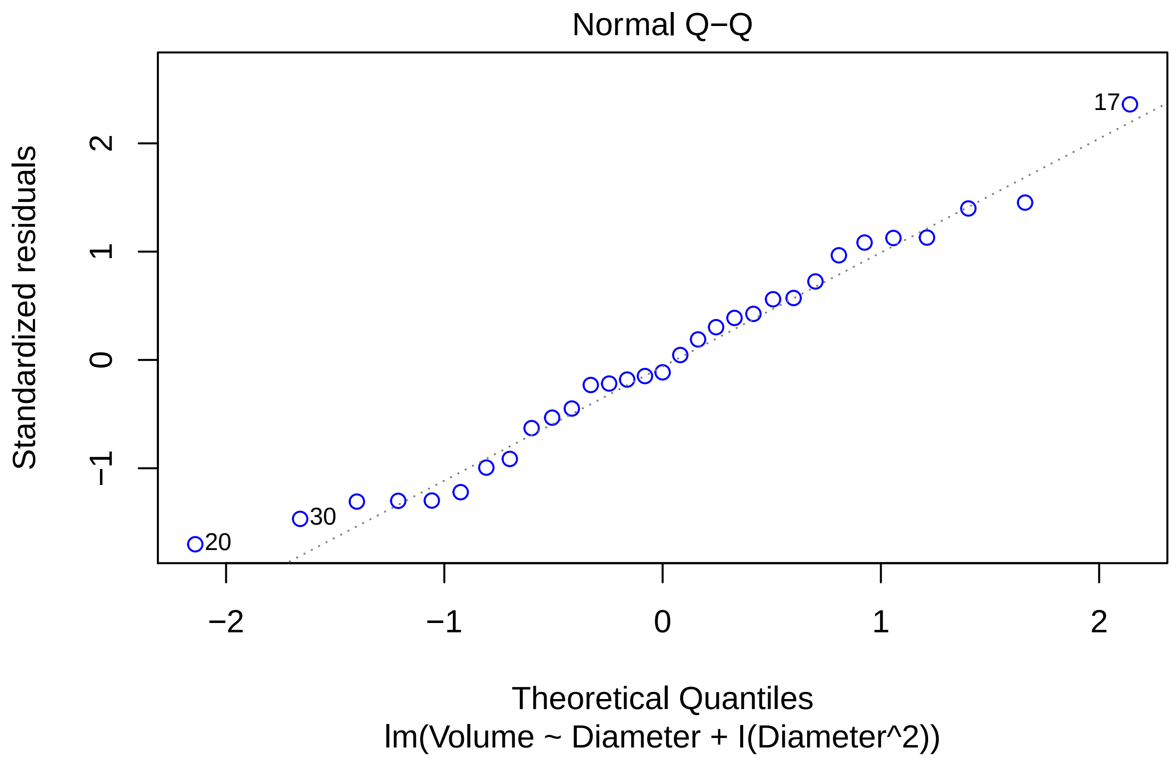
```
round(cooks.distance(model1),2)
```

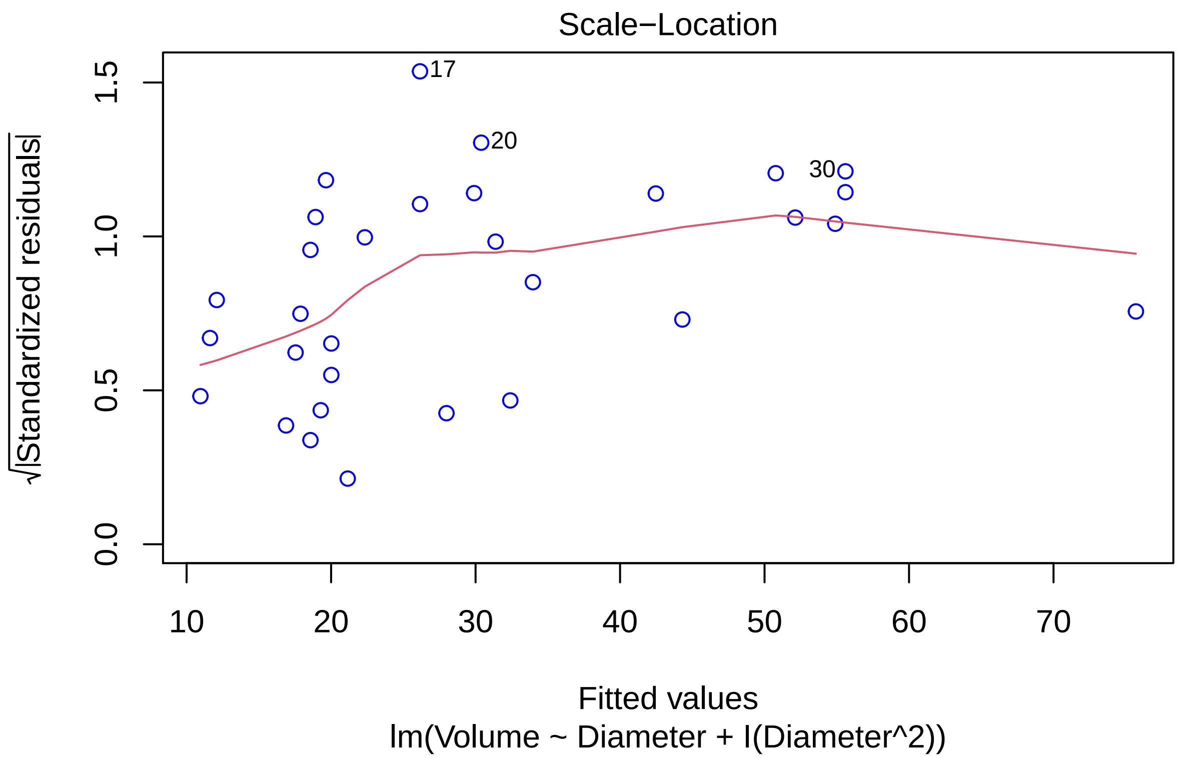
```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
```

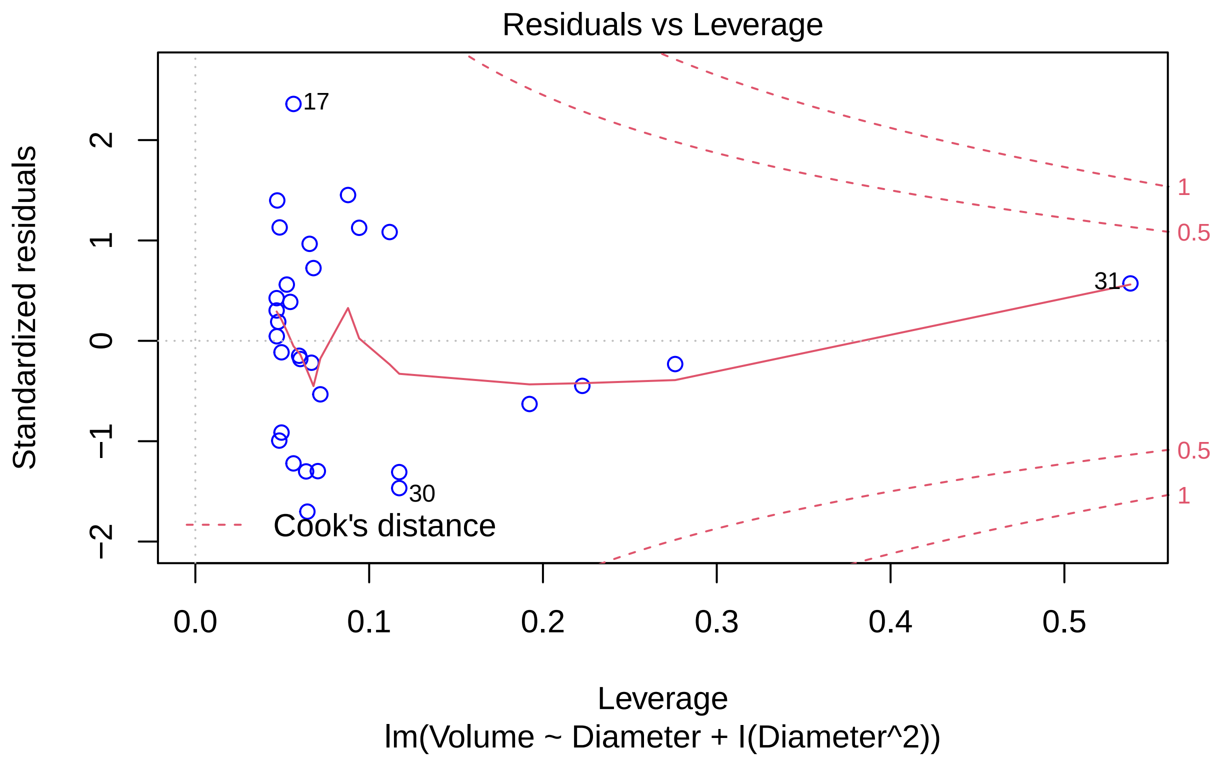
```
## 0.01 0.02 0.03 0.00 0.00 0.01 0.01 0.00 0.02 0.00 0.03 0.00 0.00 0.00 0.02 0.03
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 0.11 0.00 0.04 0.07 0.02 0.00 0.01 0.04 0.01 0.07 0.04 0.05 0.08 0.10 0.13
```

```r
#plot(model1, which=c(1,2,3,4), col=c("blue"))
plot(model1, col=c("blue"))
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Volume ~ Diameter + I(Diameter^2))
17
30
20
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
17
20
30
10
20
30
40
50
60
70
Fitted values
lm(Volume ~ Diameter + I(Diameter^2))




```r
shapiro.test(rstandard(model1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model1)
## W = 0.97453, p-value = 0.6505
```