Homework 2

Ben Ridenhour

9/21/2021

Contents

Modeling Board Feet of Lumber

In the homework, you were essentially asked to create three different models for board feet of lumber based on the diameter of the tree at waist height. The first two of the models were specified by the book; the third model was up to you to create. For each model, the appropriate scaling relationship needed to be derived based on assumptions. After that, each model needed to be plotted and assessed for fit. Finally, you were asked to choose which model was best based on your analyses.

Model One — Fixed Tree Height

In this model, you were told to assume that all trees are essentially fixed height cylinders. Mathematically, we need the volume of the cylinder to act as an approximation to board feet of lumber. Thus,

$$board\,feet \propto V_{cyl.} = \pi r^2 h.$$

And the volume of tree i having radius r_i is $V_i = \pi r_i^2 h$, if height h is fixed. If we let calculate the ratio of the volume of two trees, we get

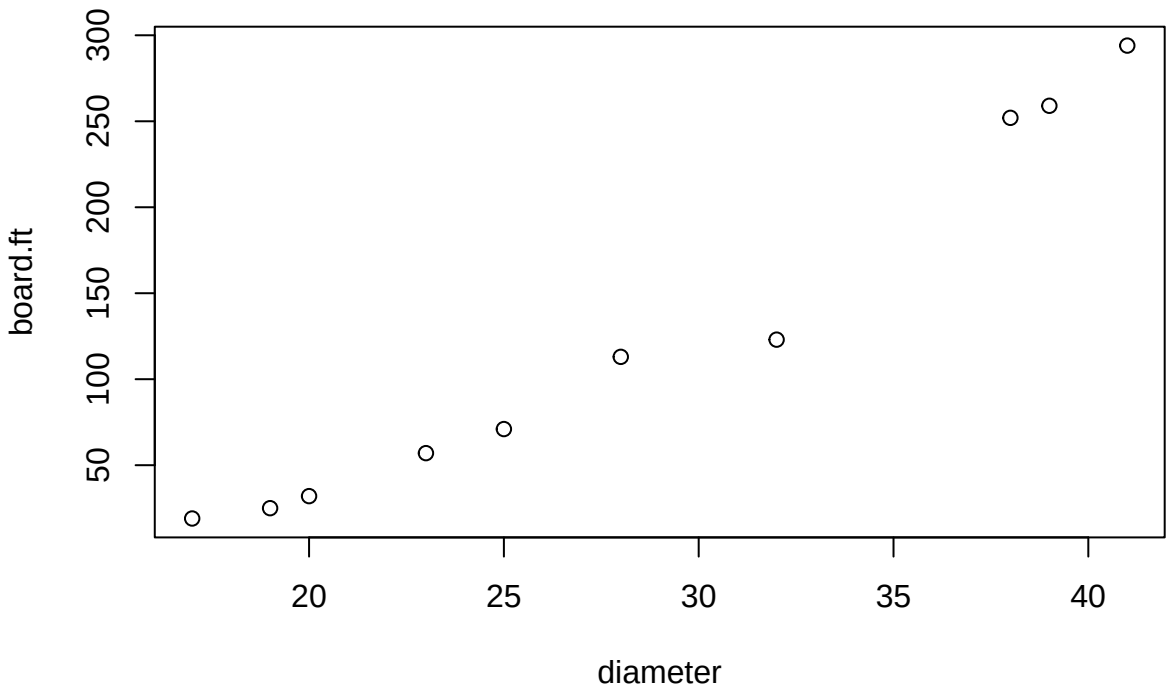
$$\frac{V_1}{V_0} = \frac{\pi r_1^2 h}{\pi r_0^2 h} = \frac{r_1^2}{r_0^2}.$$

Solving the equation for V_1 gives $V_1 = \frac{V_0}{r_0^2} r_1^2 \propto \beta r_1^2$. In other words, the volume (board feet) of a tree in this model is proportional to the squared radius, and also proportional to the squared diameter.

Fitting the Model

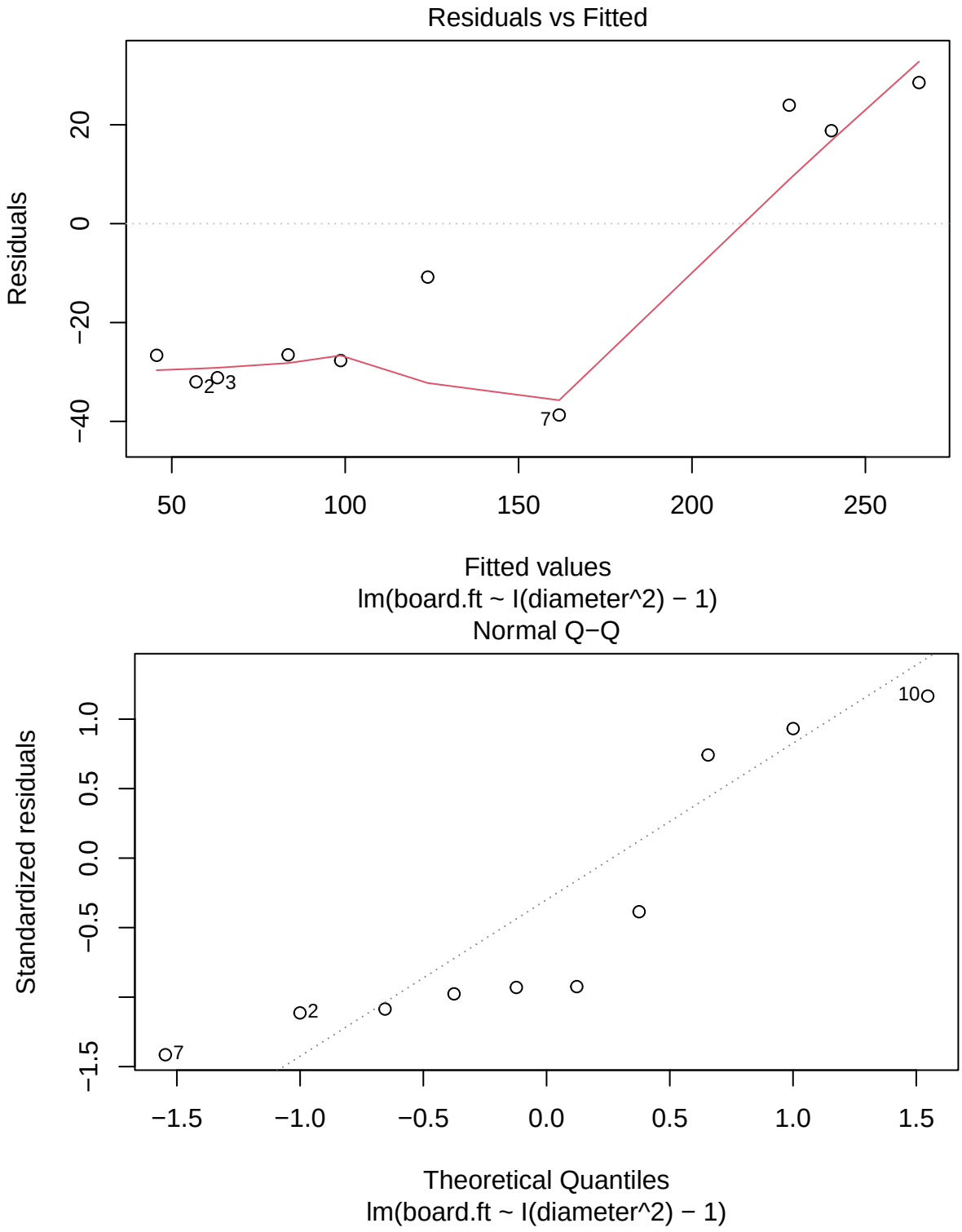
We can simply use the `lm()` function in R to fit this model. Remember, there is no intercept in the model, so wemust specify this in the function call. Before we can fit a model, we need to enter the data and visualize it.

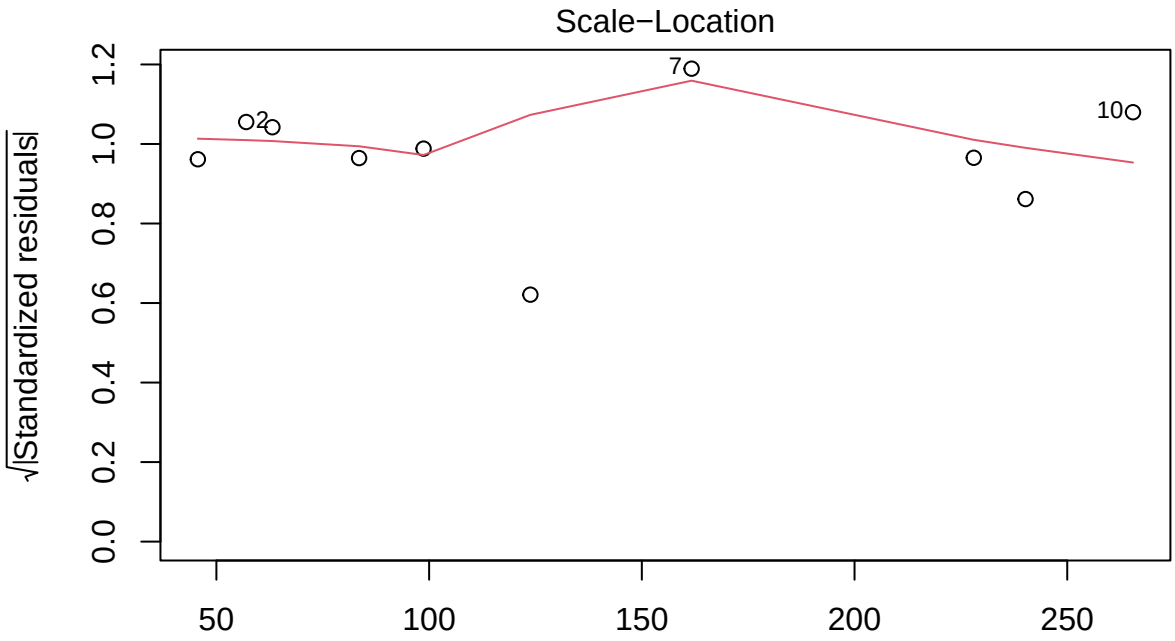
```
lumber <- data.frame(
  "diameter" = c(17,19,20,23,25,28,32,38,39,41),
  "board.ft" = c(19,25,32,57,71,113,123,252,259,294)
)

plot(lumber)
```

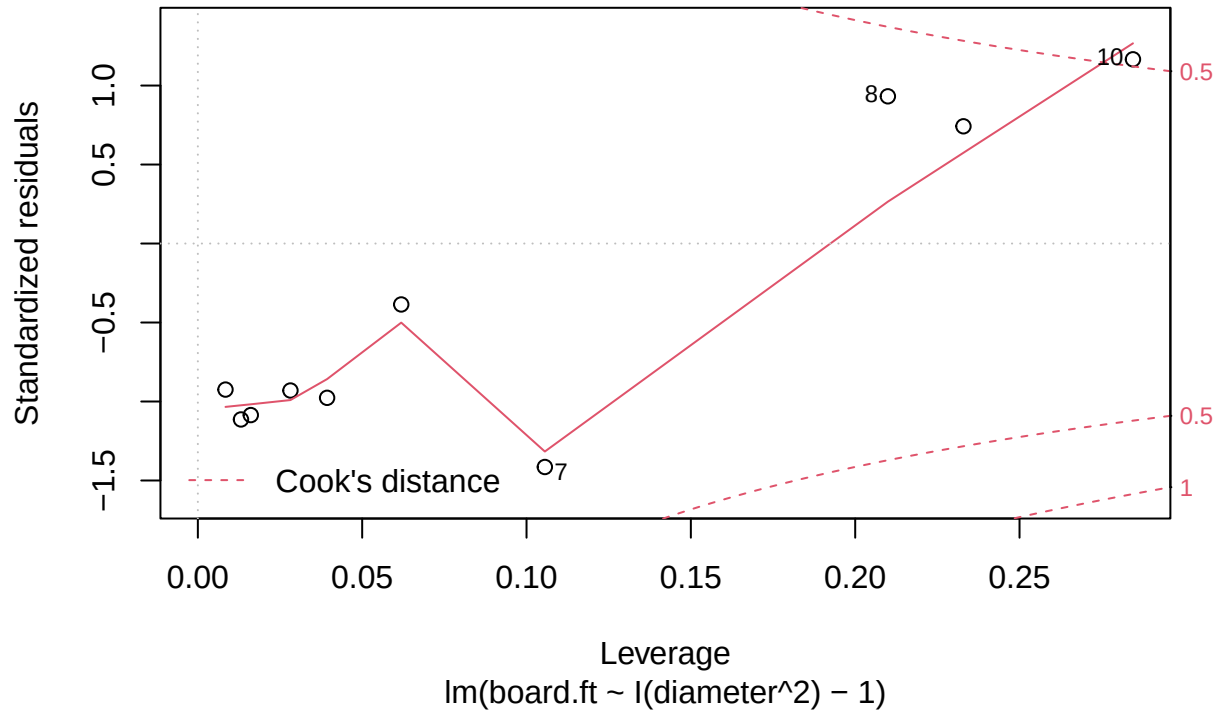


Fitting the model is fairly simple. Note the use of the `I()` function in the model. We can plot the model to see the diagnostics after fitting.

```
m1 <- lm( board.ft ~ I(diameter^2) - 1, data = lumber)
summary(m1)
plot(m1)
```


Residuals vs Fitted
Residuals
20
0
−20
−40
2
3
7
50
100
150
200
250
Fitted values
lm(board.ft ~ I(diameter^2) − 1)
Normal Q−Q
Standardized residuals
1.0
0.5
0.0
−0.5
−1.5
10
2
7
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(board.ft ~ I(diameter^2) − 1)






```
##
## Call:
## lm(formula = board.ft ~ I(diameter^2) - 1, data = lumber)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -38.71 -30.30 -26.59  11.40  28.54
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## I(diameter^2) 0.157919   0.009181    17.2 3.42e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.93 on 9 degrees of freedom
## Multiple R-squared:  0.9705, Adjusted R-squared:  0.9672
## F-statistic: 295.8 on 1 and 9 DF,  p-value: 3.418e-08
```

The model using the squared-diameter fits very well. For the model $R^2 = 0.967$ with AIC $= 98.6$. The diagnostic plots look fairly reasonable for the model; there are a few minor red flags but nothing drastic.

Model Two — Proportional Tree Height

The second model has a slight modification from the first model. Now we assume that $h_i \propto cr_i$, therefore

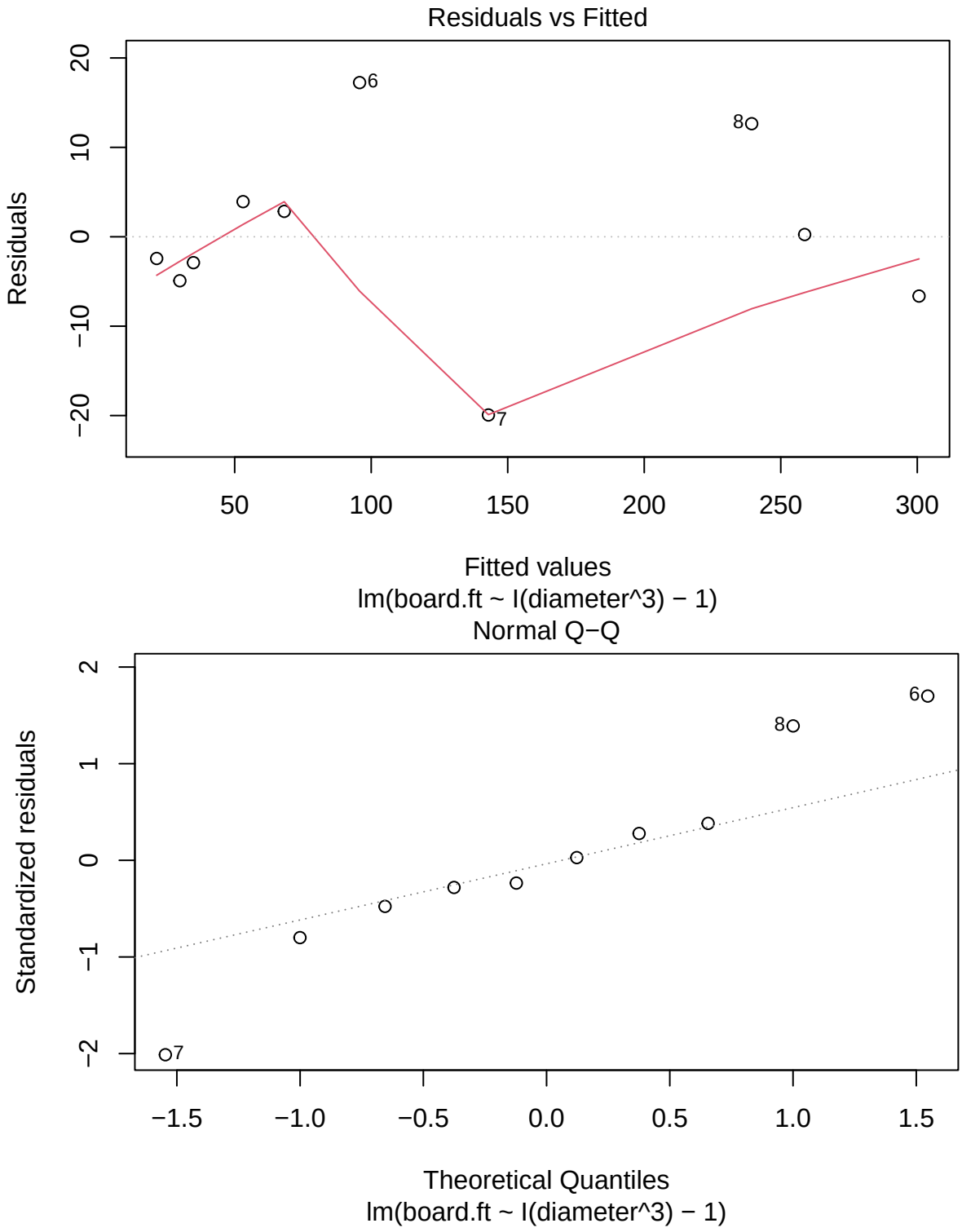
$$\frac{V_1}{V_0} = \frac{\pi r_1^2 h_1}{\pi r_0^2 h_0} = \frac{\pi r_1^2 c r_1}{\pi r_0^2 c r_0} = \frac{r_1^3}{r_0^3}.$$

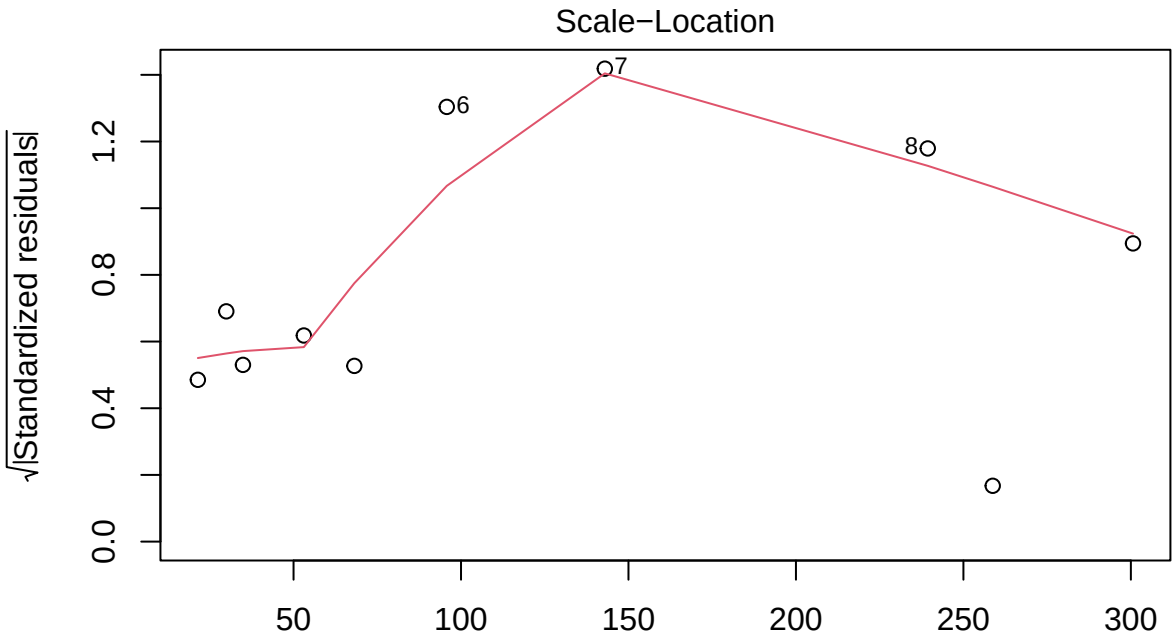
Again, now solving for V_1 yields $V_1 = \frac{V_0}{r_0^3} r_1^3 \propto \beta r_1^3$, i.e., the volume of the tree is now proportional to the cubed radius, as well as the diameter.

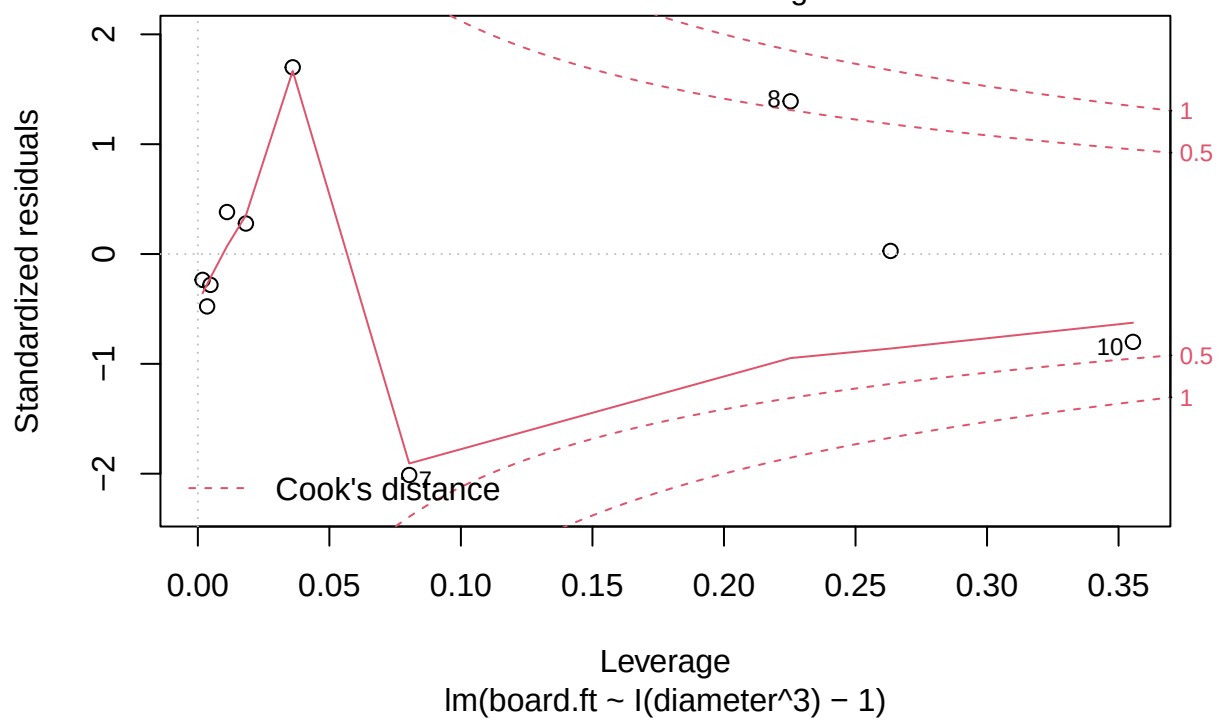
Fitting the Model

We employ the same code as for the first model, with the slight modification of cubing the diameter in the fitting process.

```
m2 <- lm( board.ft ~ I(diameter^3) - 1, data = lumber)
summary(m2)
plot(m2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(board.ft ~ I(diameter^3) − 1)
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(board.ft ~ I(diameter^3) − 1)






```
##
## Call:
## lm(formula = board.ft ~ I(diameter^3) - 1, data = lumber)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -19.935  -4.413  -1.091   3.656  17.245
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## I(diameter^3) 4.362e-03  8.938e-05    48.8 3.19e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.33 on 9 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9958
## F-statistic:  2382 on 1 and 9 DF,  p-value: 3.193e-12
```

The model using the cubed-diameter seems to fit even better. For the model $R^2 = 0.996$ with AIC $= 78$. The diagnostic plots look better for this model as well; there are a few minor red flags but nothing drastic.

Model 3 — A Hollow, Truncated Cone

For my third model, I chose to think of a tree as being a very narrow cone. Once the top of the cone becomes too narrow, the wood is no longer useful. Likewise, at some radius, r_{min}, the tree no long has usable board feet. Think of cedars trees from the Northwest which are largely hollow once they reach a certain size.

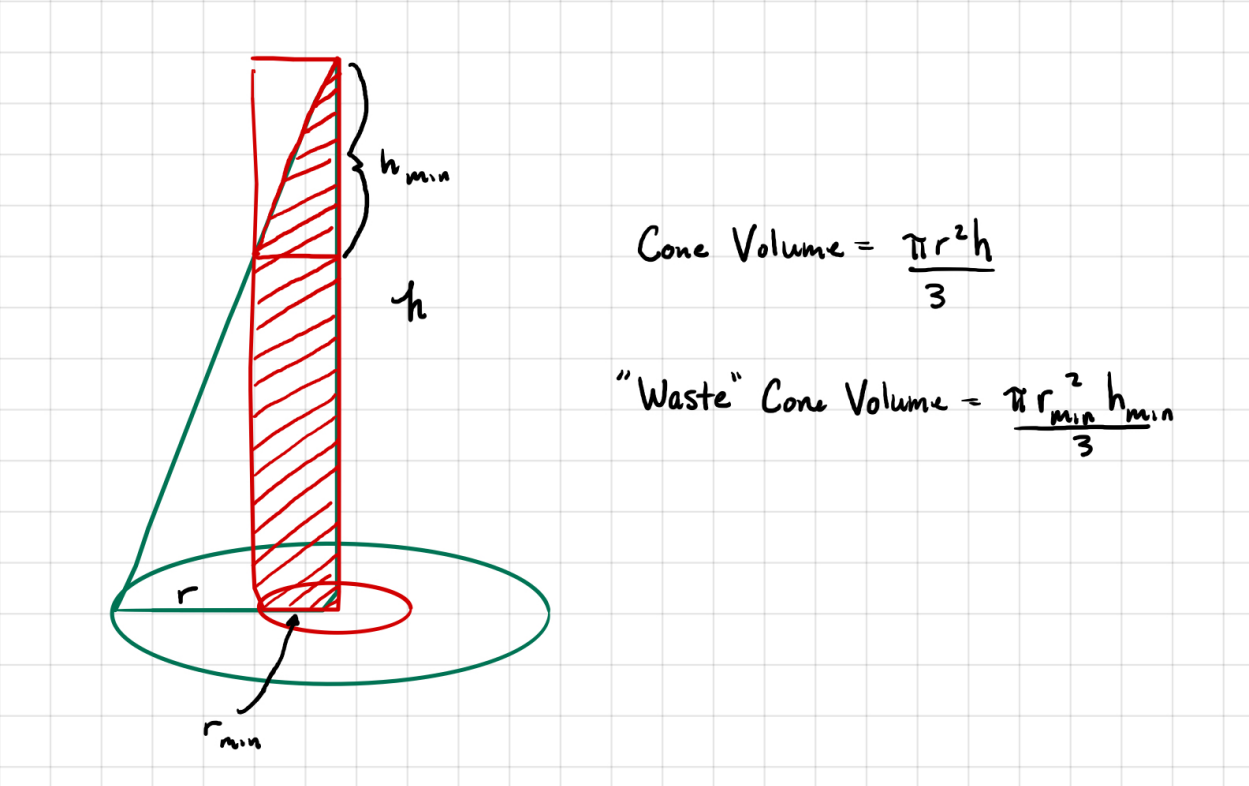


Figure 1: Cone

The above figure shows how I thought of the tree, with the relevant measures. How can we model this?

The answer is slightly trickier than our previous models. I chose to think of the volume of the full tree, $V_t = \frac{1}{3}\pi r^2 h$, the volume of the *cylinder* at the center, $V_c = \pi r_{min}^2 h$. If we take $V_t - V_c$, we have subtracted out too much volume (see the negative space at the top left of the figure). How much volume is necessary to add back in? Well, if we add back in the cylinder at the top given by $V_{top} = \pi r_{min}^2 h_{min}$, we've added back

in too much, but we can see we need to subtract the cone given by $V_{min} = \frac{1}{3}\pi r_{min}^2 h_{min}$ to fix this. The difference $V_{top} - V_{min} = \frac{1}{3}\pi r_{min}^2 h_{min} = 2V_{min}$. So now we now the volume of board feet is given by

$$board\,feet = V_t - V_c + 2V_{min}.$$

Using similarity we can further simplify the model. Specifically, using the relation $k = \frac{r}{r_{min}} = \frac{h}{h_{min}}$, we get $V_t = \frac{1}{3}\pi r^2 h = \frac{1}{3}k^3\pi r_{min}^2 h_{min} = k^3 V_{min}$ and $V_c = \pi r_{min}^2 h = k\pi r_{min}^2 h_{min} = 3kV_{min}$. Our formula for board feet now becomes

$$board\,feet = V_t - V_c + 2V_{min} = V_{min}(k^3 - 3k + 2) = V_{min}\left(\left(\frac{d}{d_{min}}\right)^3 - 3\frac{d}{d_{min}} + 2\right)$$

We can now attempt to model board feet with this relationship using the diameter d of the tree. Note however, there is a *specific* relationship between the coefficients in our model. To see this, assume

$$board\,feet = \beta_2 d^3 - \beta_1 d + \beta_0.$$

It is easy to see that $\beta_0 = 2V_{min}$. For the other coefficients, $\beta_1 = \frac{3V_{min}}{d_{min}} = \frac{3}{2d_{min}}\beta_0$ and $\beta_2 = \frac{V_{min}}{d_{min}^3} = \frac{1}{2d_{min}^3}\beta_0$. Substituting these back into our equation yields

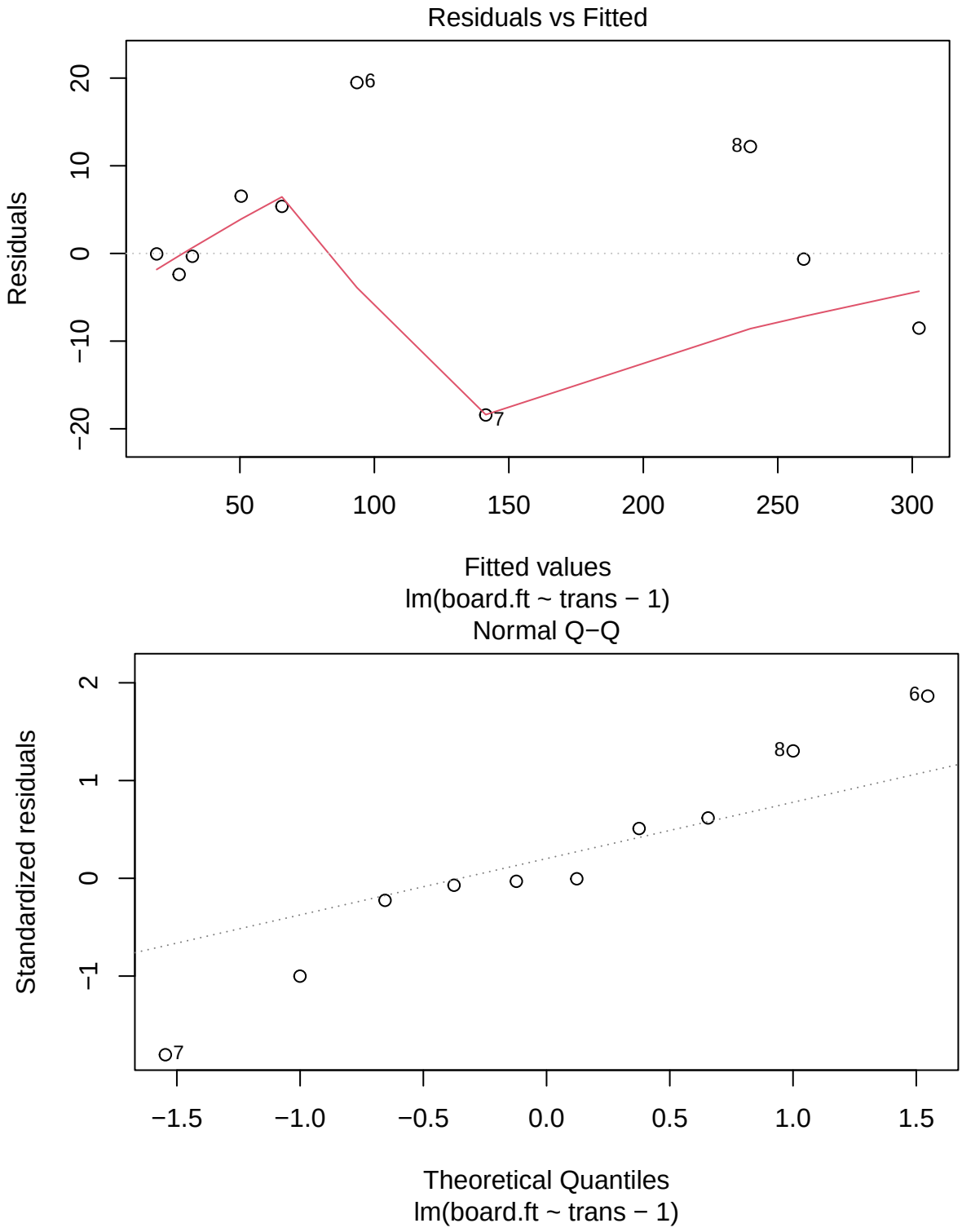
$$board\,feet = \beta_0\left(\frac{d^3}{2d_{min}^3} - \frac{3d}{2d_{min}} + 1\right),$$

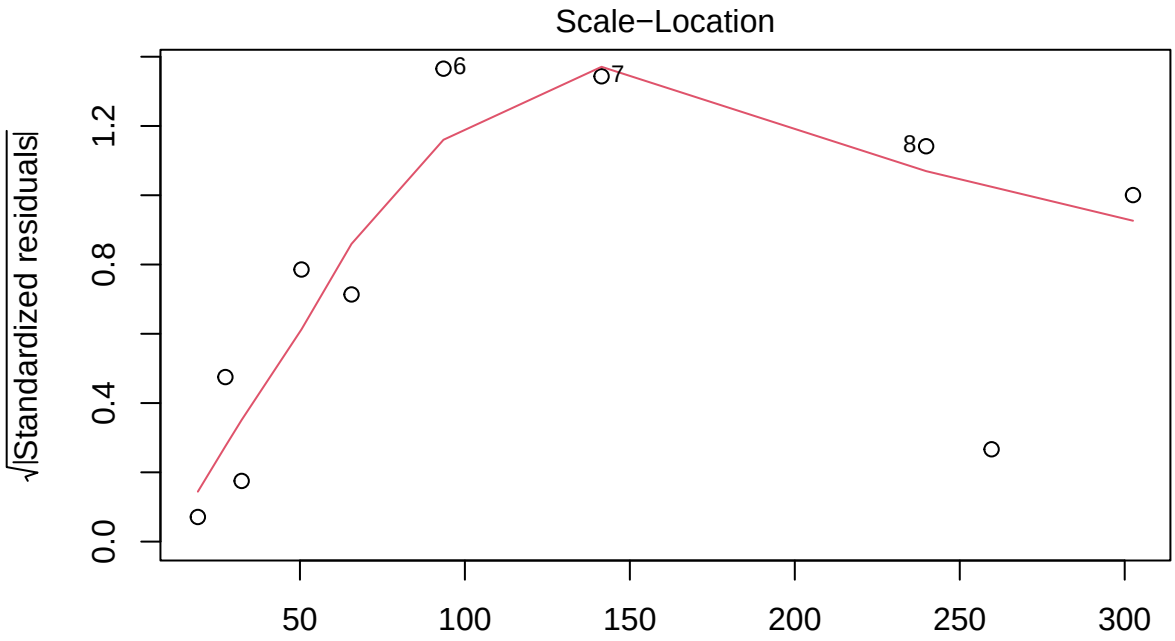
which could be more simply thought of as $board\,feet = \beta_0 f(d, d_{min})$. Obviously, we are missing one critical piece of information for fitting this model. What is it?

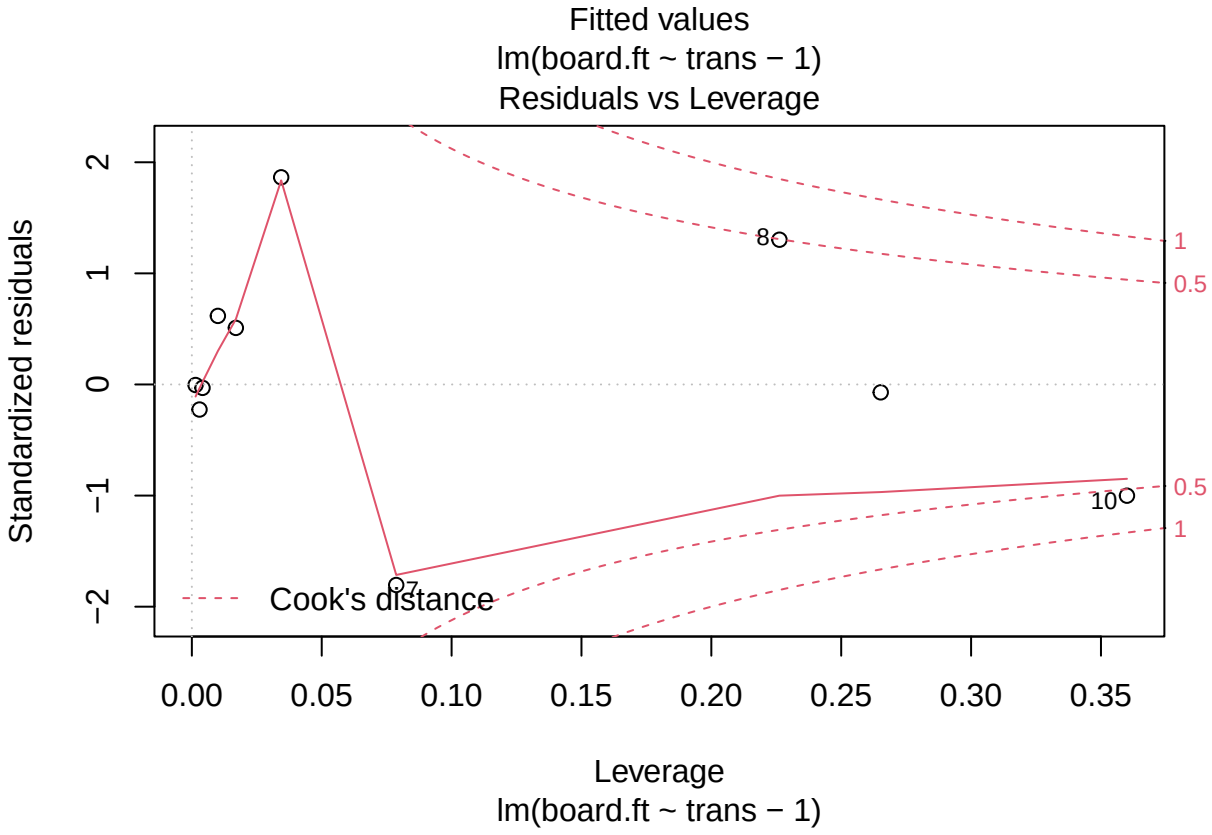
Fitting the Model

Hopefully you figured out the missing piece of the puzzle is d_{min}. For the purposes of fitting our model, we will just assume a value of d_min. Here is the code for fitting the model assuming $d_{min} = 4$. Note that we have to transform d using the RHS of the equation for board feet.

```
lumber2 <- lumber
dm <- 4 #assumed value of d_min
lumber2$trans <- with(lumber, diameter^3/(2*dm^3) - 3*diameter/(2*dm) + 1) #transform the diameter

summary(mm <- lm(board.ft ~ trans - 1 , data = lumber2)) #fit the model
plot(mm)
```


Residuals vs Fitted
Residuals
20
10
0
−10
−20
6
8
7
50
100
150
200
250
300
Fitted values
lm(board.ft ~ trans − 1)
Normal Q−Q
Standardized residuals
2
1
0
−1
6
8
7
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(board.ft ~ trans − 1)






```
##
## Call:
## lm(formula = board.ft ~ trans - 1, data = lumber2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

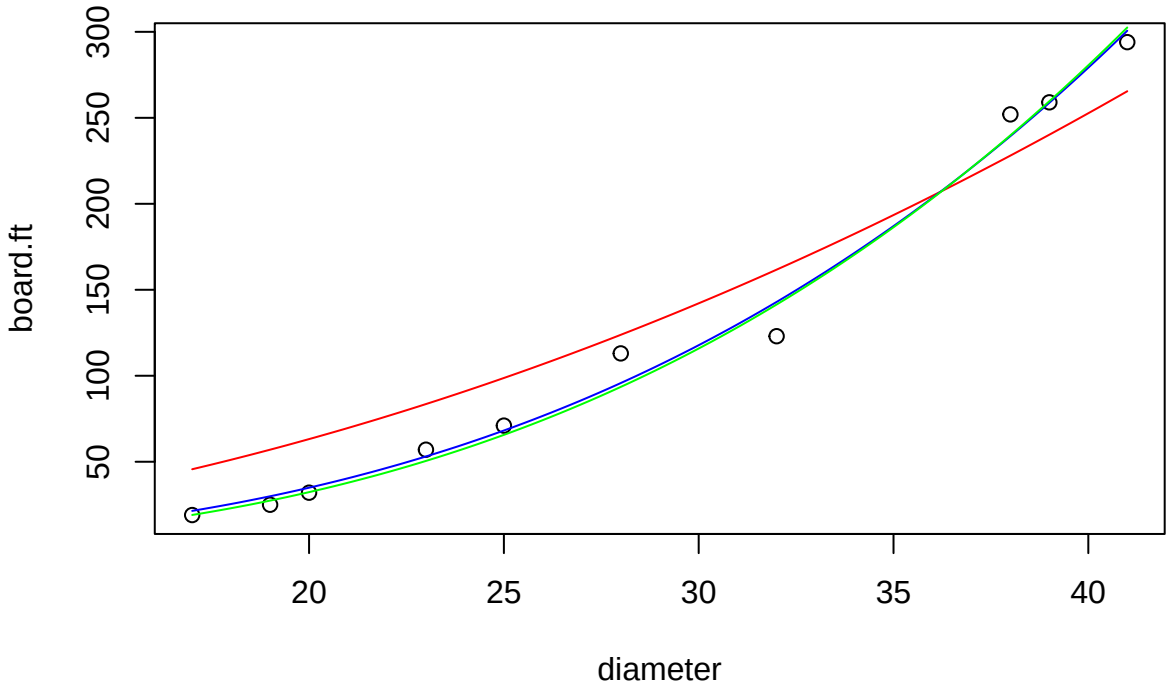
```
## -18.4251  -1.9594  -0.1897   6.2412  19.4863
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## trans  0.57725    0.01218    47.4 4.14e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.64 on 9 degrees of freedom
## Multiple R-squared:  0.996,  Adjusted R-squared:  0.9956
## F-statistic:  2247 on 1 and 9 DF,  p-value: 4.145e-12
```

As with our previous models, the fit is quite good. For the model where we assume $d_{min} = 4$, $R^2 = 0.996$ with AIC = 78.6. The diagnostics are quite comparable to those of our other two models.

Model Comparison/Selection

Now that we have three functioning models for board feet, we need to choose which one is best. First, let's plot the models with the data to visually compare the results.

```
plot(lumber)
predictIt <- data.frame(diameter = seq(17,41,len=100))
### why did I do this instead of just getting fitted values?
lines(predictIt$diameter,predict(m1, predictIt), col = "red")
lines(predictIt$diameter,predict(m2, predictIt), col = "blue")
bf <- function(d,Vm,dm) Vm/dm^3 * d^3 - 3*Vm/dm*d + 2*Vm
lines(predictIt$diameter, bf(predictIt$diameter,coef(mm)/2,dm), col="green")
```



From the plots, it looks like model 2 and 3 are better than model 1. Let's use Akaike's Information Criterion (AIC) to compare between our models. Remember, lower values of AIC are better.

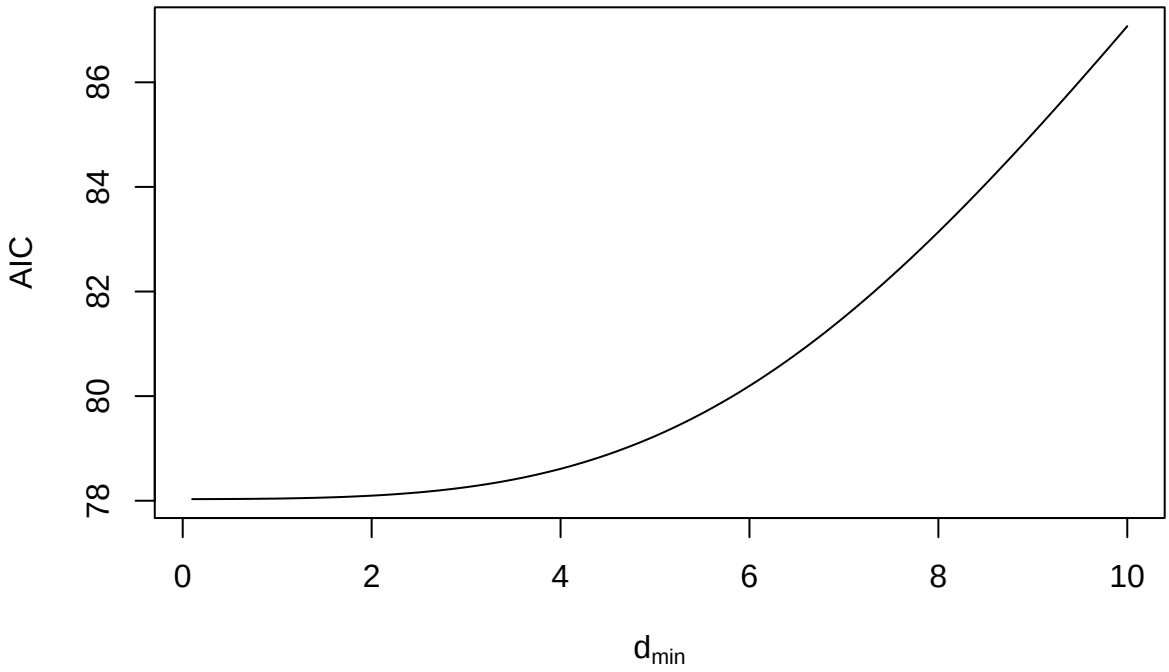
```
AIC(m1,m2,mm)
```

```
##    df      AIC
## m1  2 98.62576
```

```
## m2  2 78.03075
## mm  2 78.61007
```

Based on AIC values, it looks like model 1 is the winner, but it narrowly beats model 3. But remember, we just assumed $d_{min} = 4$ for the third model. What happens if we try to choose a better value of d_{min}? Let's do this by looping over various values d_{min} and comparing AIC values. Afterwards, we can plot the values and look for an optimum.

```
AICvals <- c()
dVals <- seq(0.1,10,by=0.1)
for(dmin in dVals){
  dm <- dmin
  lumber2$trans <- with(lumber, diameter^3/(2*dm^3) - 3*diameter/(2*dm) + 1)
  val <- AIC(mm <- lm(board.ft ~ trans - 1 , data = lumber2))
  AICvals <- c(AICvals,val)
}

#Plot the values to see if there is an optimum
plot(dVals,AICvals,type="l", xlab = expression(d[min]), ylab="AIC")
```



Our plot clearly suggests that the best value d_{min} is simply $d_{min} = 0$, at which point our more complex model 3 converges back to the simpler model based on the cubed-diameter of a tree! Thus, in the end, it looks like model 2 would be our choice based on something like AIC. Does that mean we have to choose this model?

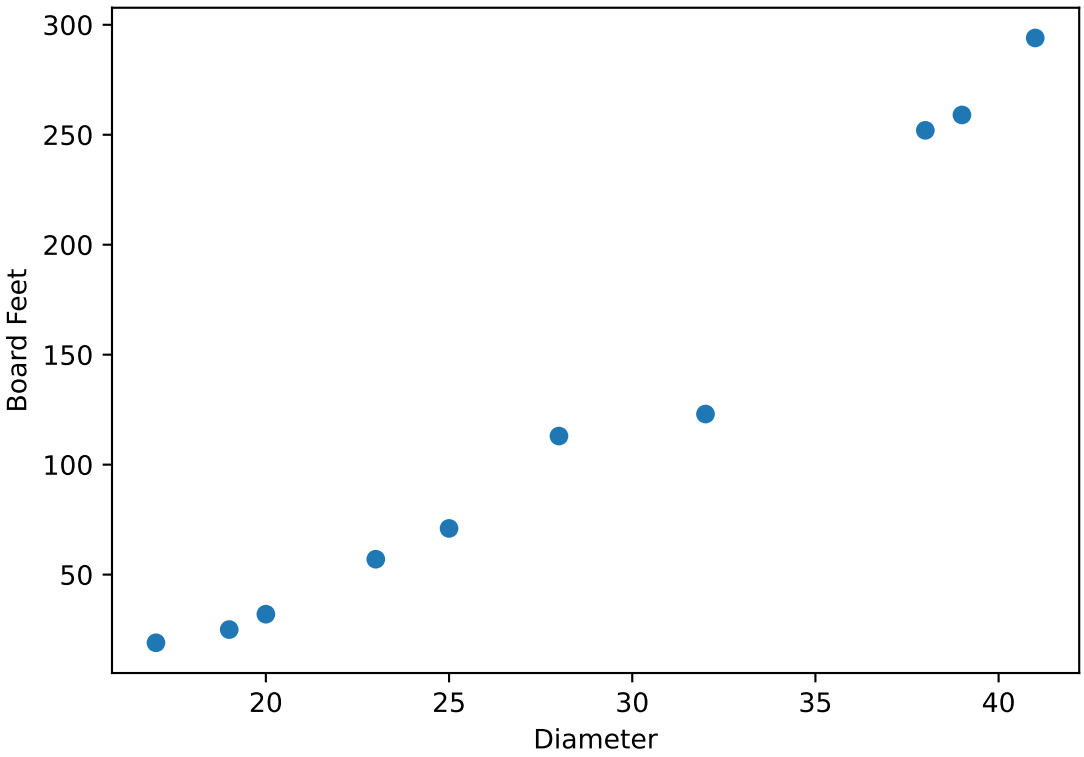
Doing it in Python

Obviously, we can do the exact same modeling exercises using the `reticulate` package and Python. Here is how to do the same calculations with Python (if you prefer it).

Model 1

```
#Get all the necessary libraries
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels as sm
import statsmodels.formula.api as smf
import OLSplot


lumber = pd.DataFrame(data = {'diameter': [17,19,20,23,25,28,32,38,39,41],
  'board_ft': [19,25,32,57,71,113,123,252,259,294]})


plt.scatter(lumber.diameter, lumber.board_ft)
plt.xlabel('Diameter')
plt.ylabel('Board Feet')
plt.show()
plt.close()
```

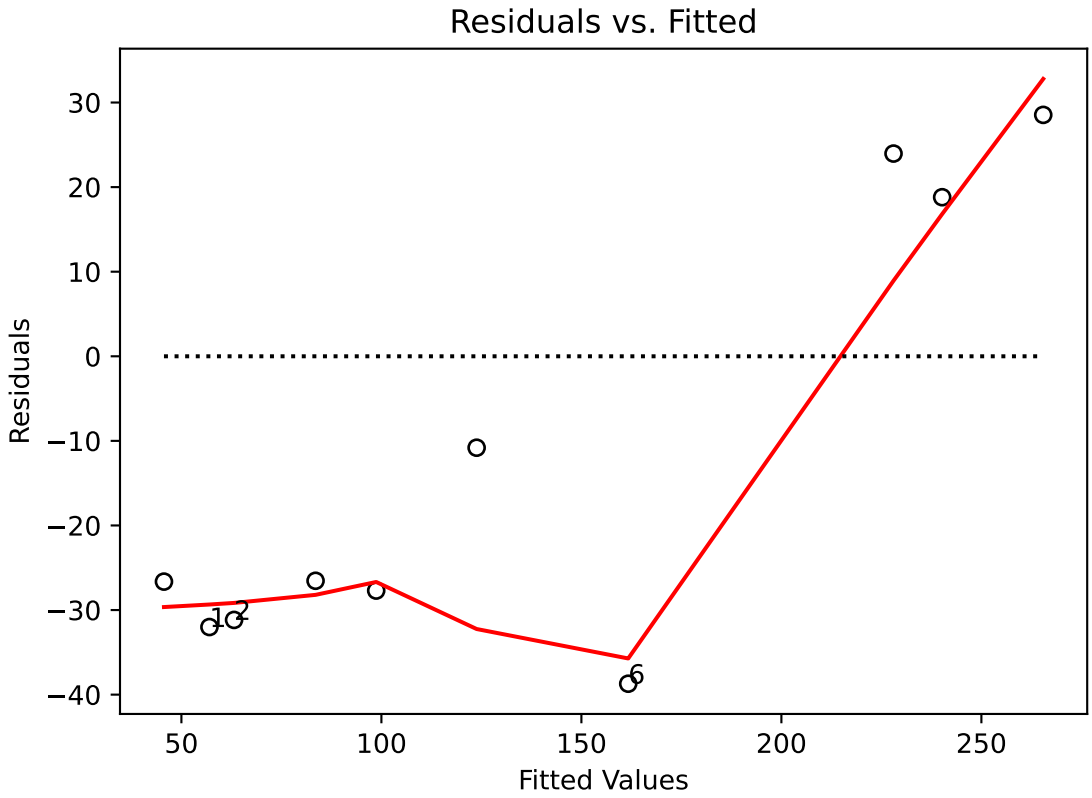


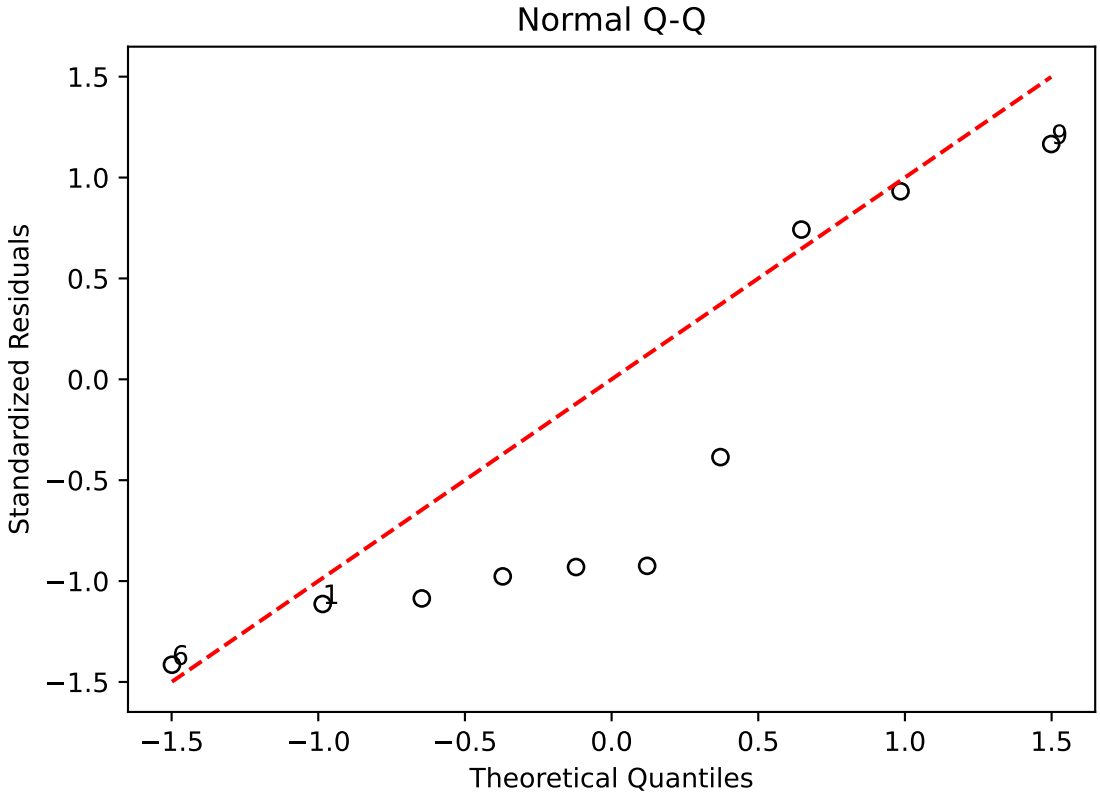

```
m1 = smf.ols('board_ft ~ np.power(diameter,2) - 1', data=lumber)
results1 = m1.fit()
results1.summary()
```

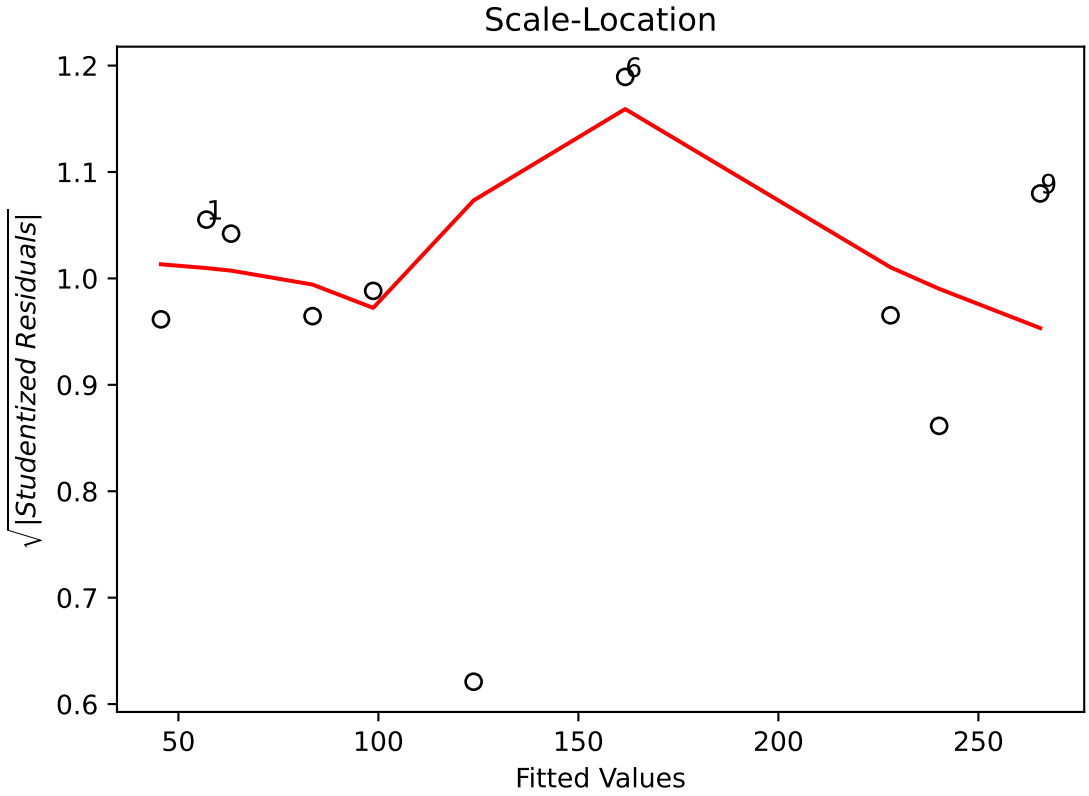
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                                 OLS Regression Results
## ==============================================================================
```

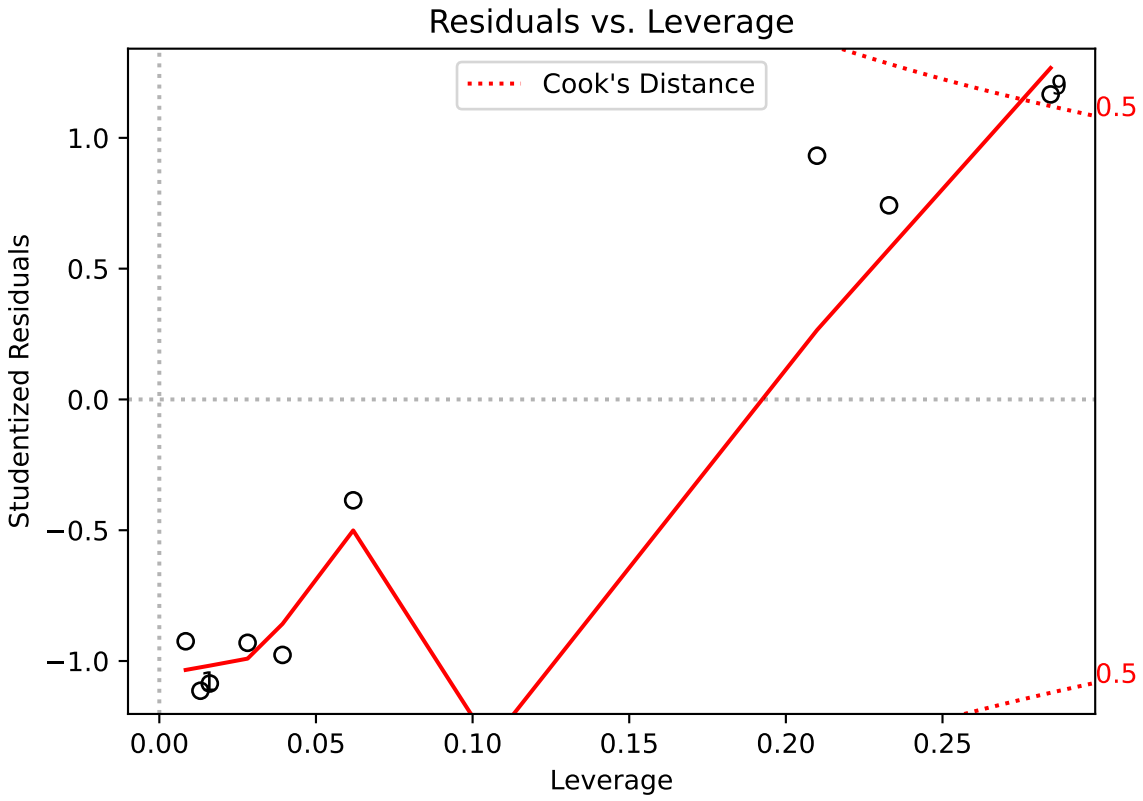
```
## Dep. Variable:               board_ft   R-squared (uncentered):                   0.970
## Model:                            OLS   Adj. R-squared (uncentered):              0.967
## Method:                 Least Squares   F-statistic:                              295.8
## Date:                Mon, 20 Sep 2021   Prob (F-statistic):                    3.42e-08
## Time:                        17:39:21   Log-Likelihood:                         -47.313
## No. Observations:                  10   AIC:                                      96.63
## Df Residuals:                       9   BIC:                                      96.93
## Df Model:                           1
## Covariance Type:            nonrobust
## ==============================================================================================
##                              coef    std err          t      P>|t|      [0.025      0.975]
## ----------------------------------------------------------------------------------------------
## np.power(diameter, 2)     0.1579      0.009     17.200      0.000       0.137       0.179
## ==============================================================================
## Omnibus:                        2.647   Durbin-Watson:                   0.686
## Prob(Omnibus):                  0.266   Jarque-Bera (JB):                1.461
## Skew:                           0.689   Prob(JB):                        0.482
## Kurtosis:                       1.733   Cond. No.                         1.00
## ==============================================================================
##
## Notes:
## [1] R² is computed without centering (uncentered) since the model does not contain a constant.
## [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
##
## /Users/bridenhour/anaconda3/envs/r-reticulate/lib/python3.9/site-packages/scipy/stats/stats.py:1541:
##   warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
OLSplot.ResidFitted(results1)
plt.show()
plt.close()
OLSplot.qqplot(results1)
plt.show()
plt.close()
OLSplot.ScaleLocation(results1)
plt.show()
plt.close()
OLSplot.Leverage(results1)
plt.show()
plt.close()
```


Residuals vs. Fitted
Residuals
Fitted Values
30
20
10
0
−10
−20
−30
−40
50
100
150
200
250
1
2
6


Normal Q-Q
Standardized Residuals
Theoretical Quantiles
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1
6
9


Scale-Location
1.2
1.1
1.0
0.9
0.8
0.7
0.6
√|Studentized Residuals|
50
100
150
200
250
Fitted Values
1
6
9



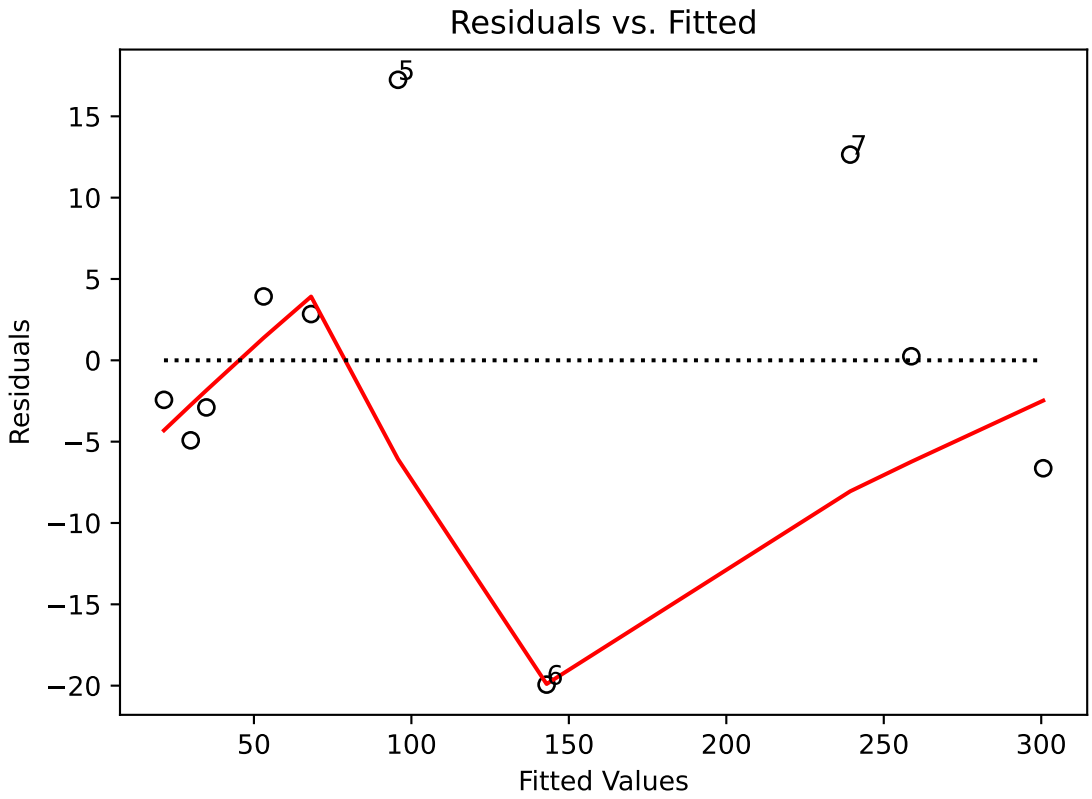
Model 2

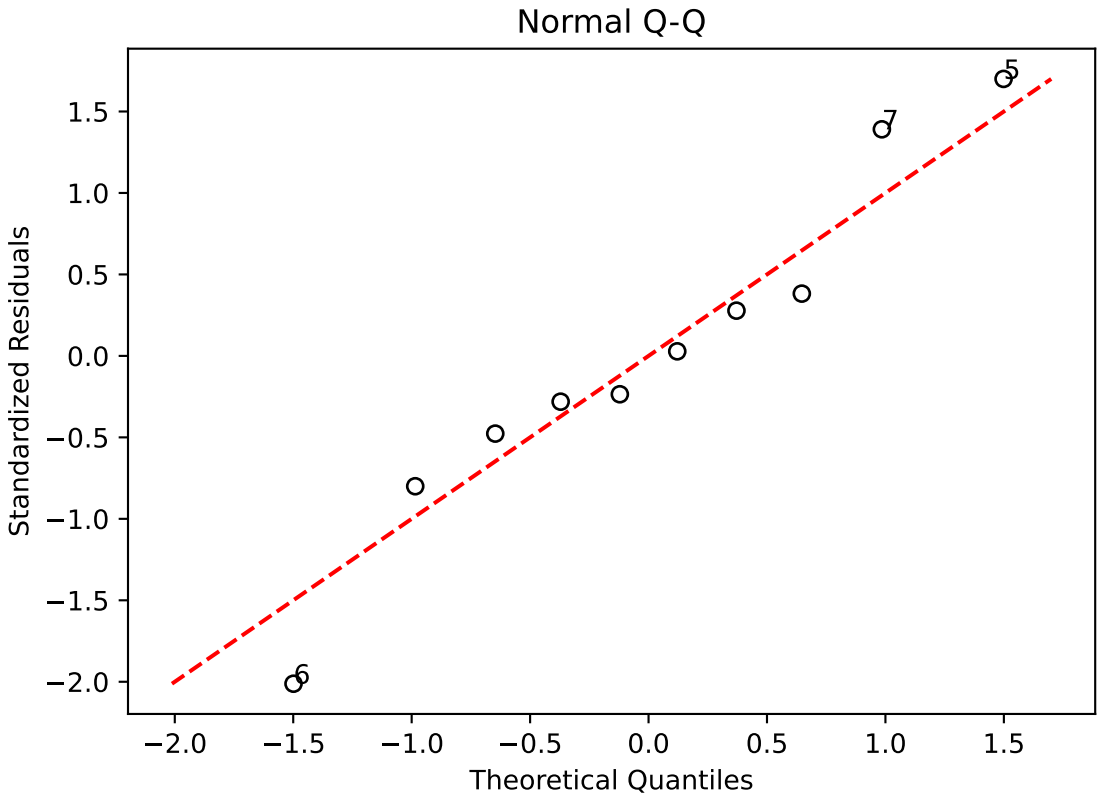
```
m2 = smf.ols('board_ft ~ np.power(diameter,3) - 1', data=lumber)
results2 = m2.fit()
results2.summary()
```

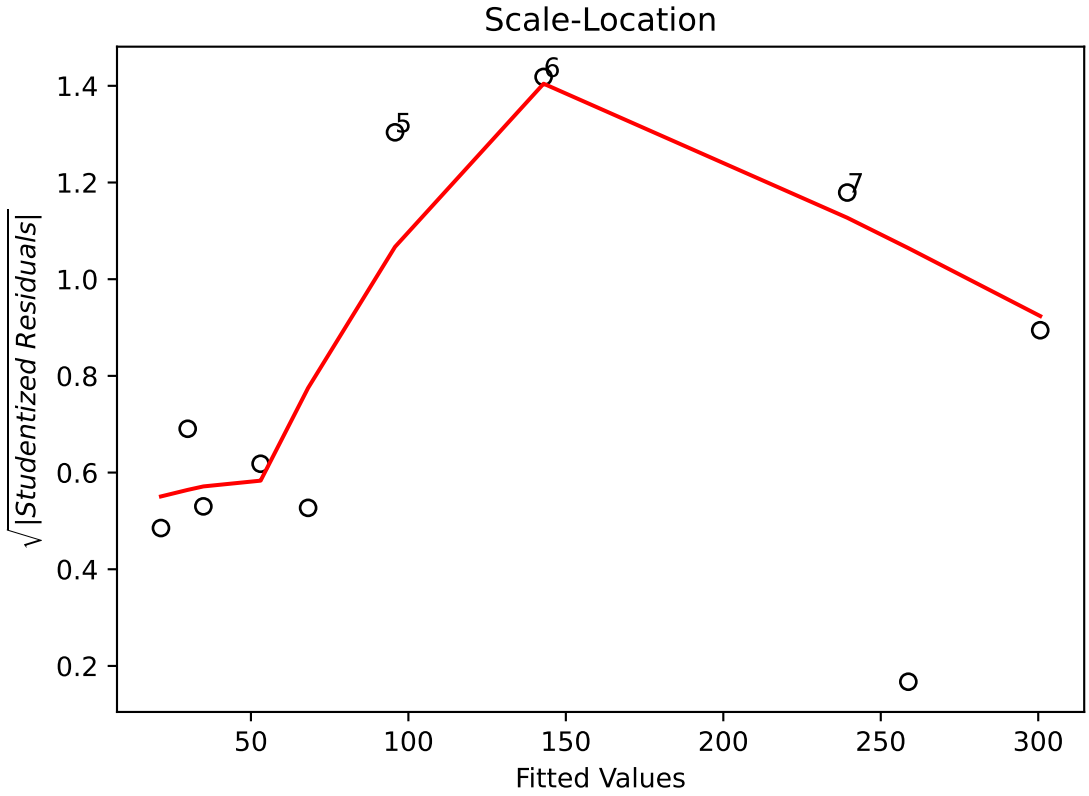
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                                 OLS Regression Results
## ==============================================================================
## Dep. Variable:               board_ft   R-squared (uncentered):                   0.996
## Model:                            OLS   Adj. R-squared (uncentered):              0.996
## Method:                 Least Squares   F-statistic:                              2382.
## Date:                Mon, 20 Sep 2021   Prob (F-statistic):                    3.19e-12
## Time:                        17:39:22   Log-Likelihood:                         -37.015
## No. Observations:                  10   AIC:                                      76.03
## Df Residuals:                       9   BIC:                                      76.33
## Df Model:                           1
## Covariance Type:            nonrobust
## ==============================================================================
##                              coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## np.power(diameter, 3)     0.0044   8.94e-05     48.802      0.000       0.004       0.005
## ==============================================================================
## Omnibus:                        0.764   Durbin-Watson:                   3.029
## Prob(Omnibus):                  0.682   Jarque-Bera (JB):                0.026
```

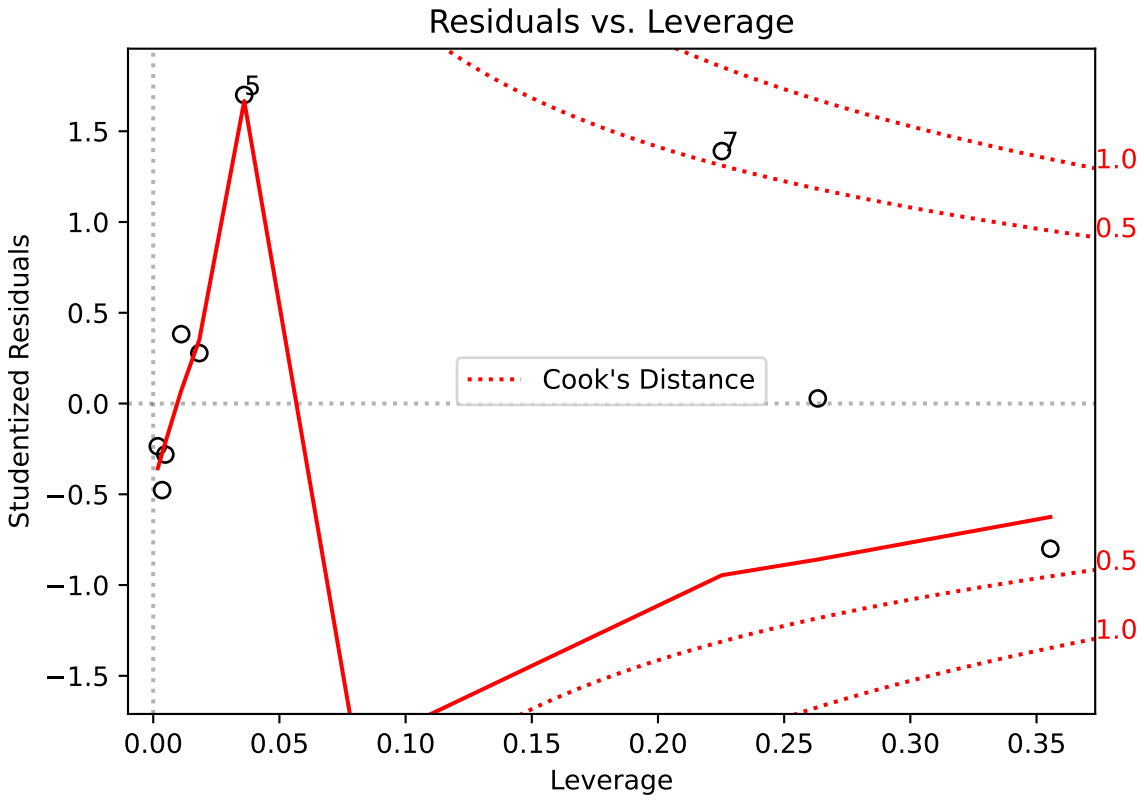
```
## Skew:                          -0.124   Prob(JB):                         0.987
## Kurtosis:                       2.978   Cond. No.                         1.00
## ==============================================================================
##
## Notes:
## [1] R² is computed without centering (uncentered) since the model does not contain a constant.
## [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
##
## /Users/bridenhour/anaconda3/envs/r-reticulate/lib/python3.9/site-packages/scipy/stats/stats.py:1541:
##   warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
OLSplot.ResidFitted(results2)
plt.show()
plt.close()
OLSplot.qqplot(results2)
plt.show()
plt.close()
OLSplot.ScaleLocation(results2)
plt.show()
plt.close()
OLSplot.Leverage(results2)
plt.show()
plt.close()
```




Normal Q-Q
Standardized Residuals
Theoretical Quantiles
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
5
7
6


Scale-Location
√|Studentized Residuals|
1.4
1.2
1.0
0.8
0.6
0.4
0.2
50
100
150
200
250
300
Fitted Values
6
5
7



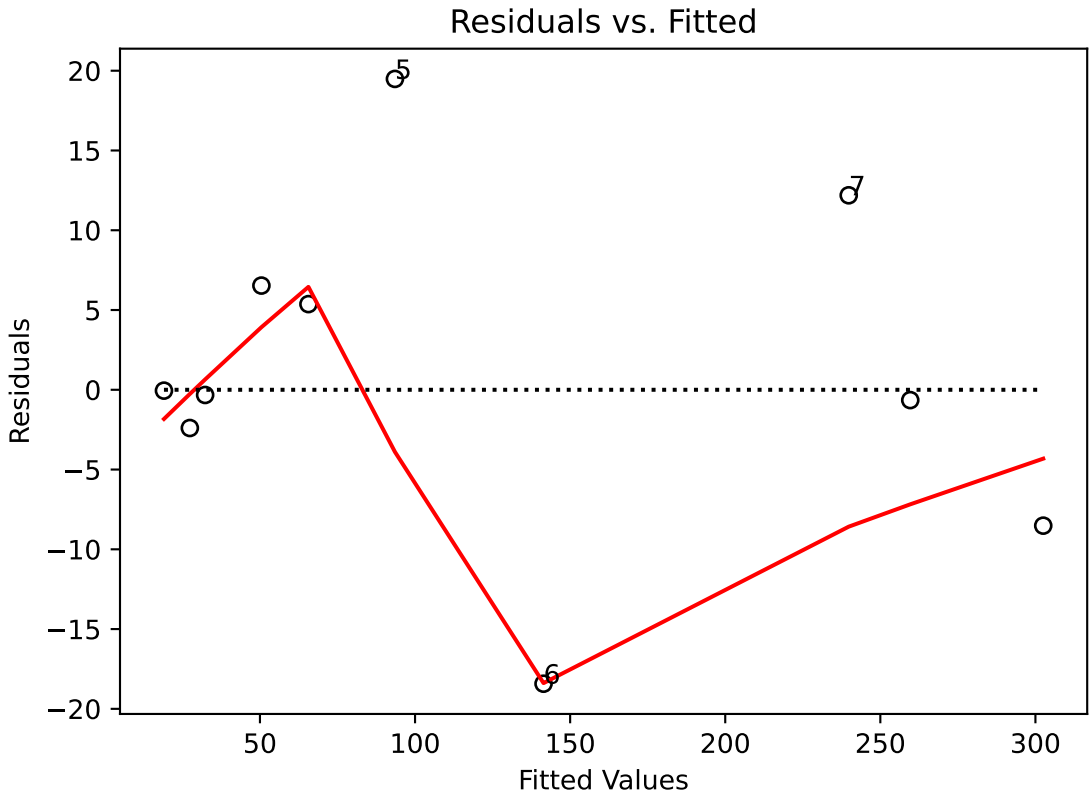
Model 3

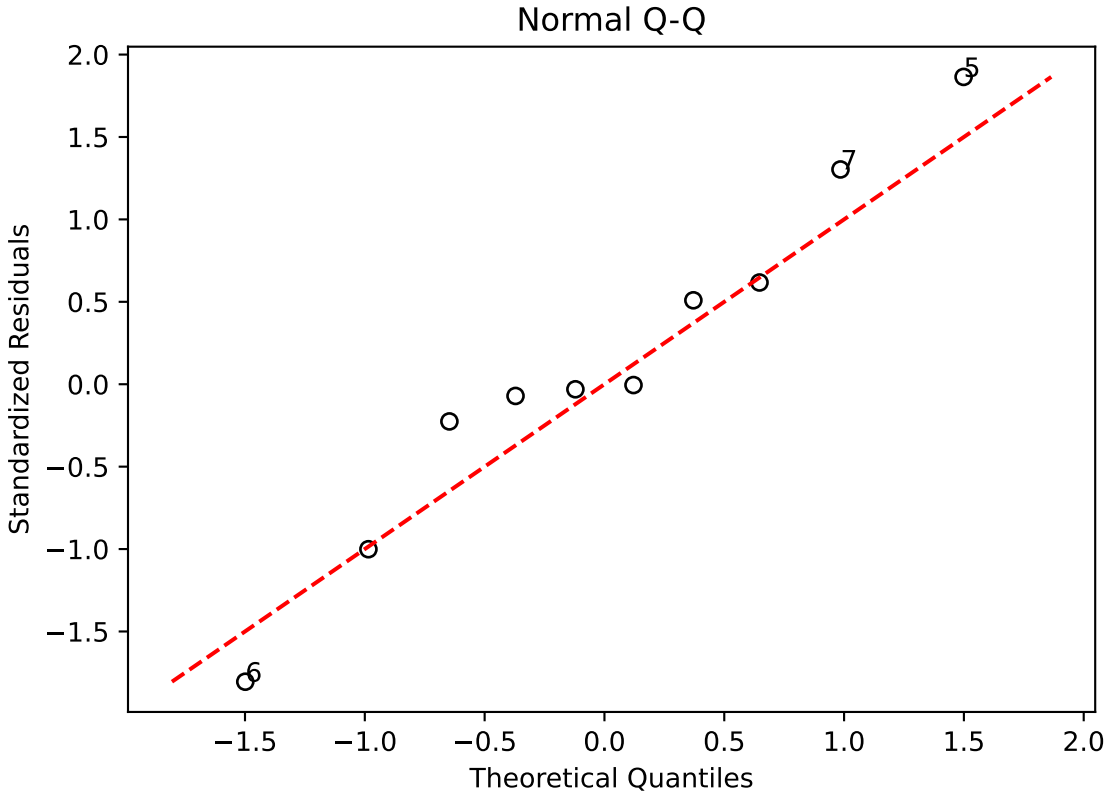
```
dm = 4 #assumed value of d_min
lumber['trans'] = lumber.diameter**3/(2*dm**3) - 3*lumber.diameter/(2*dm) + 1

m3 = smf.ols('board_ft ~ trans - 1', data = lumber)
results3 = m3.fit()
results3.summary()
```

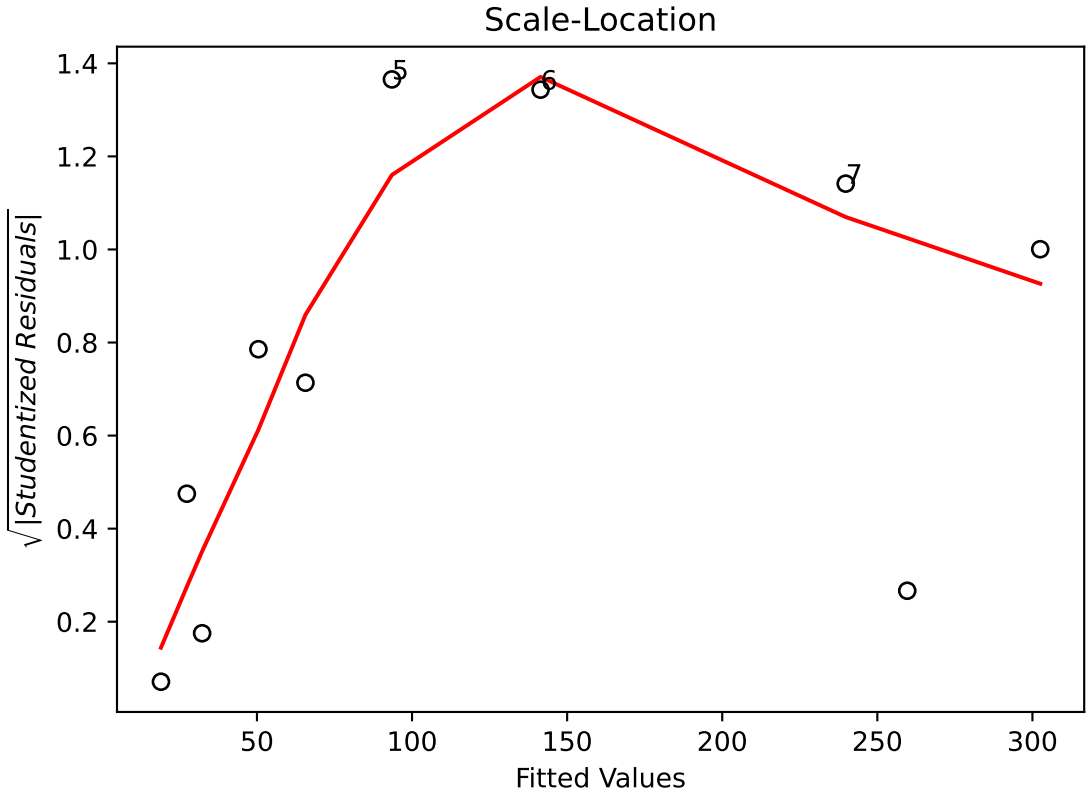
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                                 OLS Regression Results
## ==============================================================================
## Dep. Variable:               board_ft   R-squared (uncentered):                   0.996
## Model:                            OLS   Adj. R-squared (uncentered):              0.996
## Method:                 Least Squares   F-statistic:                              2247.
## Date:                Mon, 20 Sep 2021   Prob (F-statistic):                    4.14e-12
## Time:                        17:39:24   Log-Likelihood:                         -37.305
## No. Observations:                  10   AIC:                                      76.61
## Df Residuals:                       9   BIC:                                      76.91
## Df Model:                           1
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## trans          0.5772      0.012     47.403      0.000       0.550       0.605
```

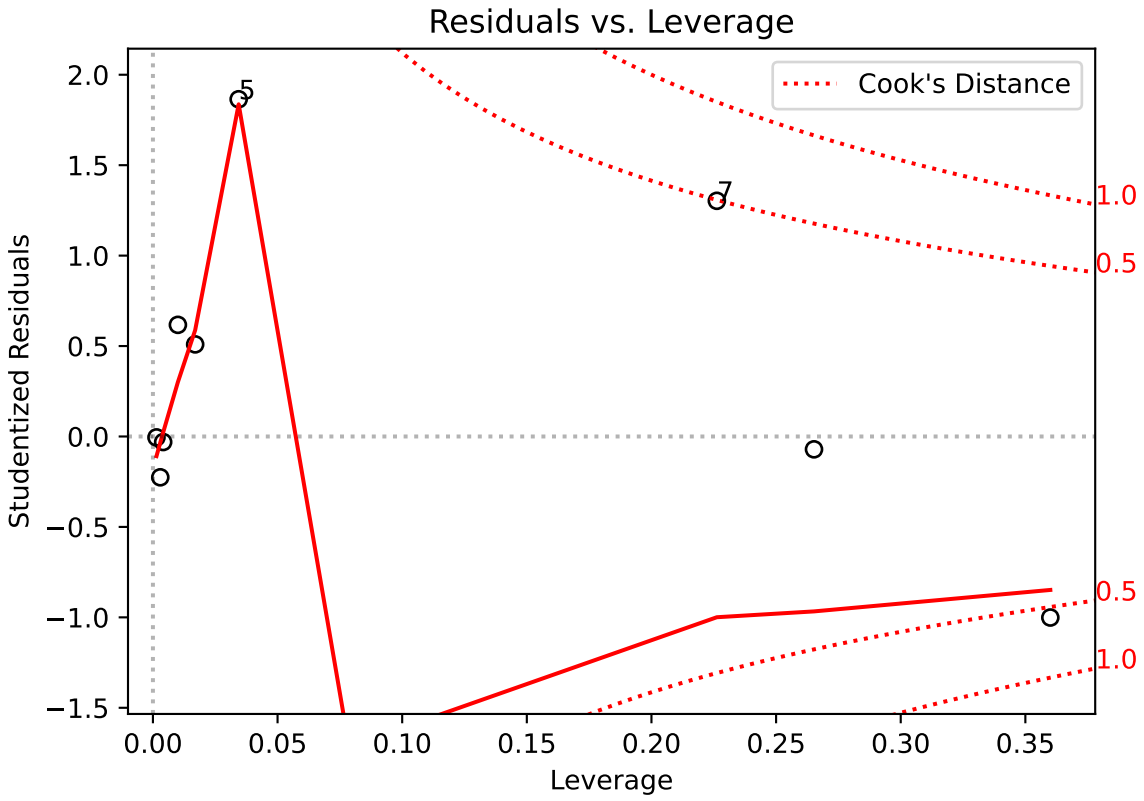
```
## ==============================================================================
## Omnibus:                        0.544   Durbin-Watson:                   2.808
## Prob(Omnibus):                  0.762   Jarque-Bera (JB):                0.035
## Skew:                          -0.123   Prob(JB):                        0.983
## Kurtosis:                       2.851   Cond. No.                         1.00
## ==============================================================================
##
## Notes:
## [1] R² is computed without centering (uncentered) since the model does not contain a constant.
## [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
## """
##
## /Users/bridenhour/anaconda3/envs/r-reticulate/lib/python3.9/site-packages/scipy/stats/stats.py:1541:
##   warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
OLSplot.ResidFitted(results3)
plt.show()
plt.close()
OLSplot.qqplot(results3)
plt.show()
plt.close()
OLSplot.ScaleLocation(results3)
plt.show()
plt.close()
OLSplot.Leverage(results3)
plt.show()
plt.close()
```


Residuals vs. Fitted
Residuals
Fitted Values
20
15
10
5
0
−5
−10
−15
−20
50
100
150
200
250
300
5
7
6

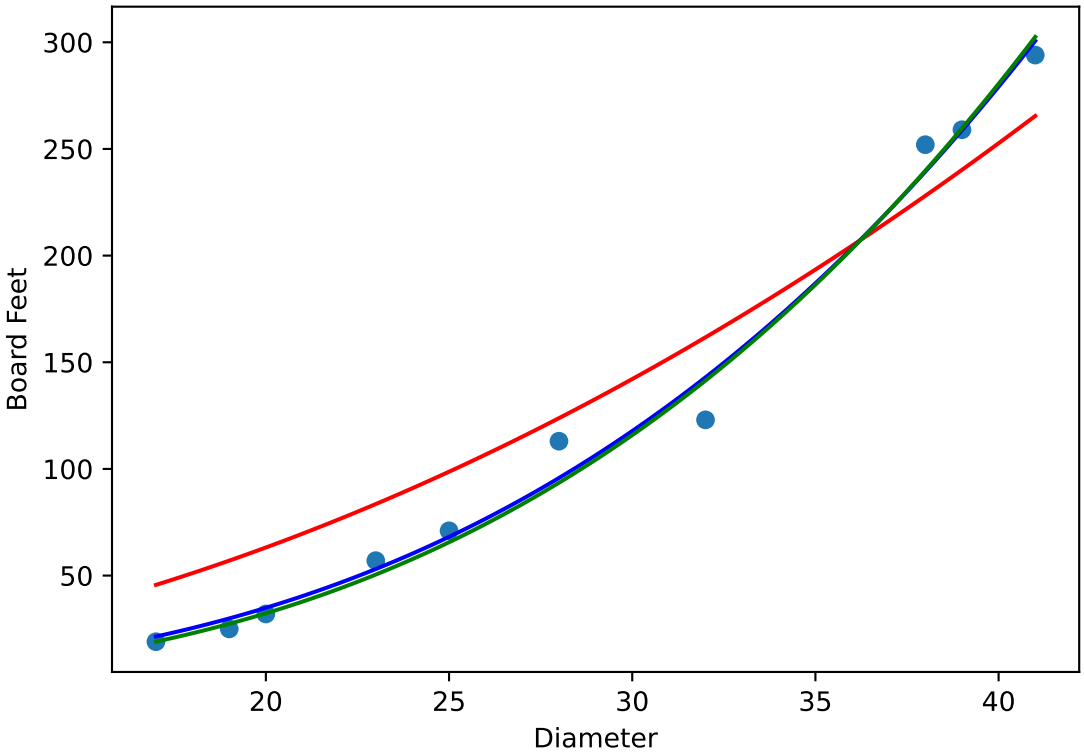

Normal Q-Q
Standardized Residuals
Theoretical Quantiles
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
5
7
6


Scale-Location
√|Studentized Residuals|
1.4
1.2
1.0
0.8
0.6
0.4
0.2
5
6
7
50
100
150
200
250
300
Fitted Values



Model Comparison

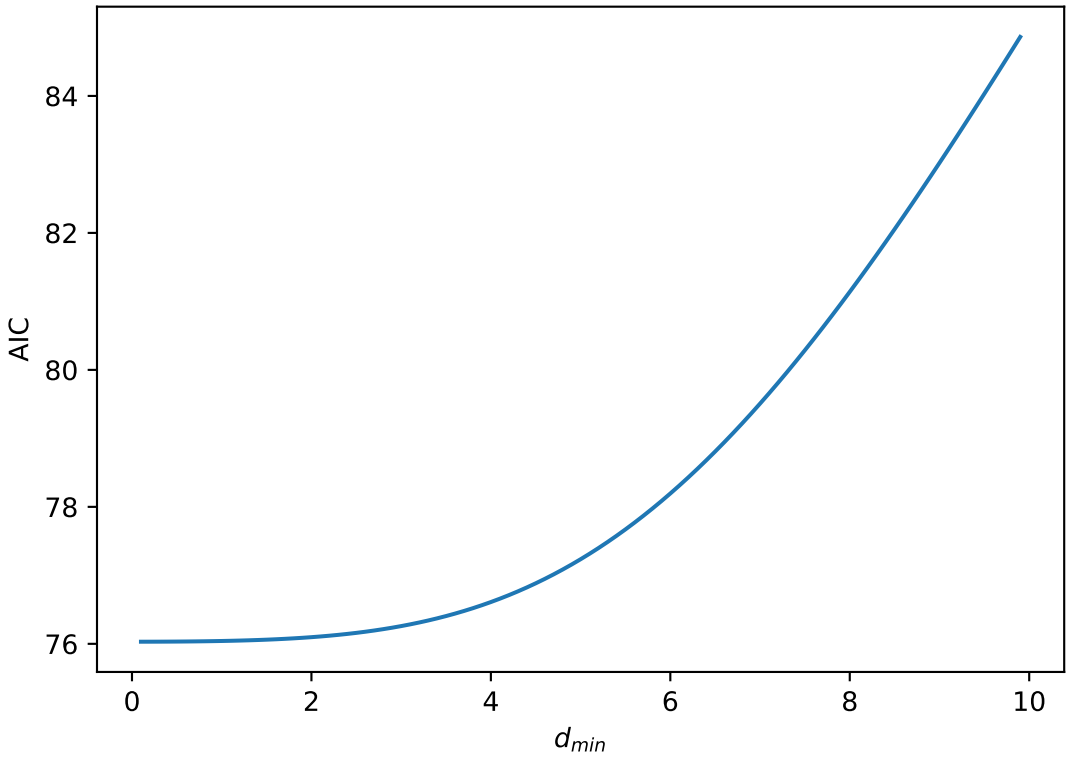
```
plt.scatter(lumber.diameter, lumber.board_ft)
plt.xlabel('Diameter')
plt.ylabel('Board Feet')

X = pd.DataFrame(data = {'diameter':np.linspace(17,41,100)})
X['trans'] = X.diameter**3/(2*dm**3) - 3*X.diameter/(2*dm) + 1

plt.plot(X.diameter, results1.predict(X), color = 'r')
plt.plot(X.diameter, results2.predict(X), color = 'b')
plt.plot(X.diameter, results3.predict(X), color = 'g')
plt.show()
plt.close()
```




```
print('AIC for model 1:', results1.aic)
print('AIC for model 2:', results2.aic)
print('AIC for model 3:', results3.aic)
```

```
## AIC for model 1: 96.62575918487445
## AIC for model 2: 76.03074910319467
## AIC for model 3: 76.61007361321796
```

```
AICvals = np.empty(0)
dVals = np.arange(0.1,10,0.1)
for dmin in dVals:
  lumber['trans'] = lumber.diameter**3/(2*dmin**3) - 3*lumber.diameter/(2*dmin) + 1
  mm = smf.ols('board_ft ~ trans - 1', data = lumber)
  AICvals = np.append(AICvals, (mm.fit()).aic)

plt.plot(dVals, AICvals)
plt.xlabel('$d_{min}$')
plt.ylabel('AIC')
plt.show()
plt.close()
```


AIC
84
82
80
78
76
0
2
4
6
8
10
d_{min}