ENTMLGY 6702 Entomological Techniques and Data Analysis

Transformations and Curvilinear Models

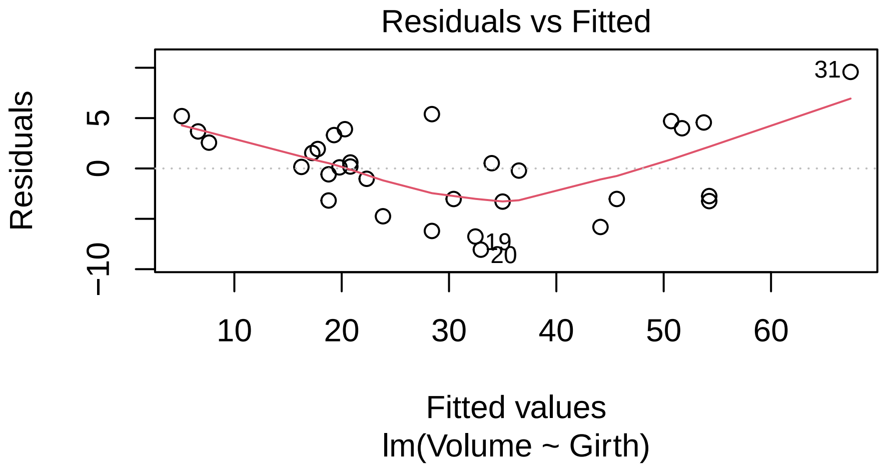
We will use the `trees` data from the `datasets` package for this tutorial. Fit a simple linear regression:

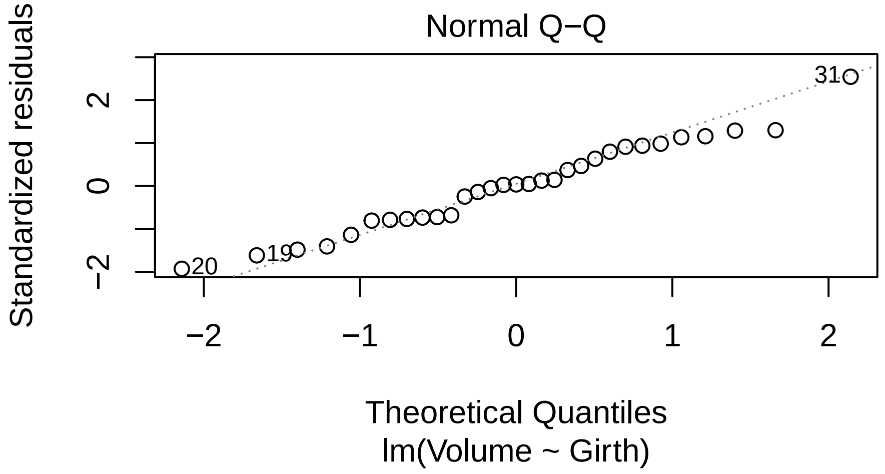
```
fit_trees_1 <- lm(Volume~Girth, data=trees)
summary(fit_trees_1)
```

```
##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## 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 ***
## Girth         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
```

And then take a look at the residuals again, which as you might recall from the simple linear regression tutorial, do not look so great: whereas I'd be okay with that qq-plot, there is some curvature in the residual plot.

```
plot(fit_trees_1, which=c(1,2))
```





We typically deal with residuals that violate assumptions by transforming the response variable and/or predictor and/or by fitting "higher order" (e.g., squared or quadratic) terms. We cover these approaches below.

1 Transformations

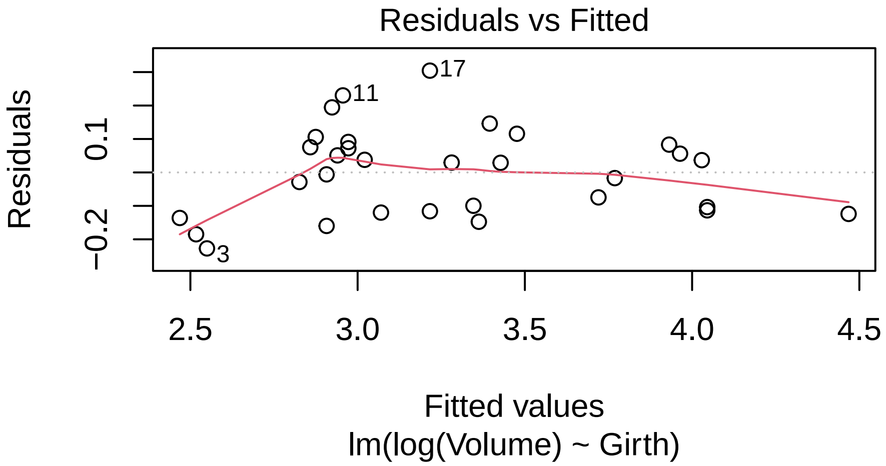
A log or square-root transformation of the response variable can often help with correcting issues of normality and heteroscedasticity. Here, we will just try log-transforming.

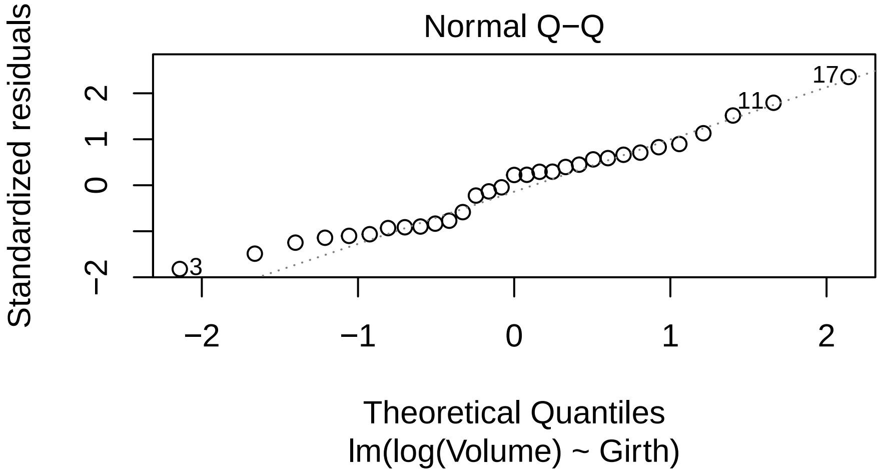
```
fit_trees_log_1 <- lm(log(Volume)~Girth, data=trees)
summary(fit_trees_log_1)
```

```
##
## Call:
## lm(formula = log(Volume) ~ Girth, data = trees)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22719 -0.11468  0.02889  0.07930  0.30436
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.118997   0.104021   10.76 1.23e-11 ***
## Girth       0.162566   0.007647   21.26  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1314 on 29 degrees of freedom
## Multiple R-squared:  0.9397, Adjusted R-squared:  0.9376
## F-statistic:   452 on 1 and 29 DF,  p-value: < 2.2e-16
```

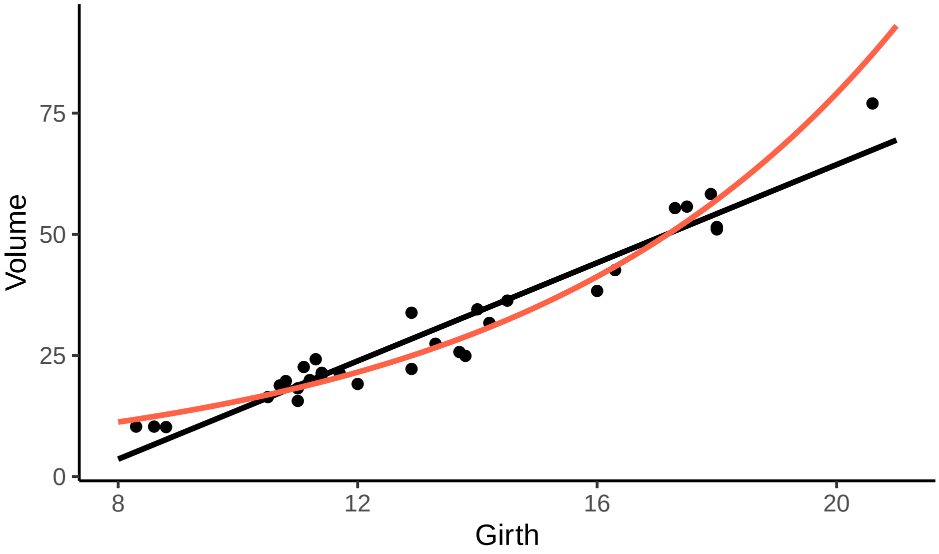
```
plot(fit_trees_log_1, which=c(1,2))
```





The normality assumption looks okay, and the residual plot, in my opinion, is close enough. If we wanted to plot the line from our model with a log-transformation onto the graph, we would need to back-transform our predictions: our model actually fit the regression line to a log-transformed version of `Volume` as a function of `Girth`.

```r
new_data <- data.frame(Girth = seq(8, 21, 0.01))
new_data$Predicted_trees_lm  <- predict(fit_trees_1, newdata=new_data)
new_data$Predicted_trees_lm_log  <- exp(predict(fit_trees_log_1, newdata=new_data))
ggplot(data=trees, mapping=aes(x=Girth, y=Volume))+
        geom_point()+theme_classic()+
        geom_line(data=new_data, aes(x=Girth, y=Predicted_trees_lm), linewidth=1)+
        geom_line(data=new_data, aes(x=Girth, y=Predicted_trees_lm_log),
                  color="tomato1", linewidth=1)
```



2 Curvilinear fits

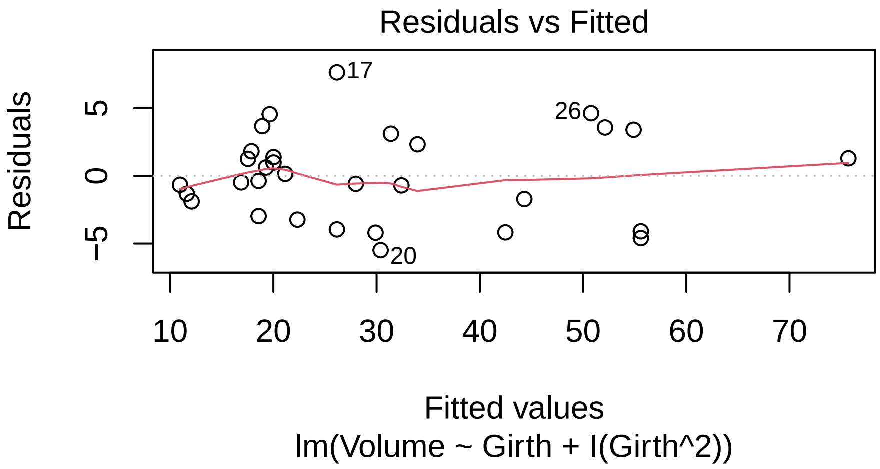
If on initial plotting or in the residual plots you notice some curvature, a curvilinear model might provide a better fit. This can be achieved by fitting a quadratic or squared term to the model. Note: when one includes higher order terms (x^2, x^3, x^4), it is convention to include all lower order terms in the model. For example, you would rarely ever fit $Y \sim X^3$, as the appropriate model would be $Y \sim X + X^2 + X^3$.

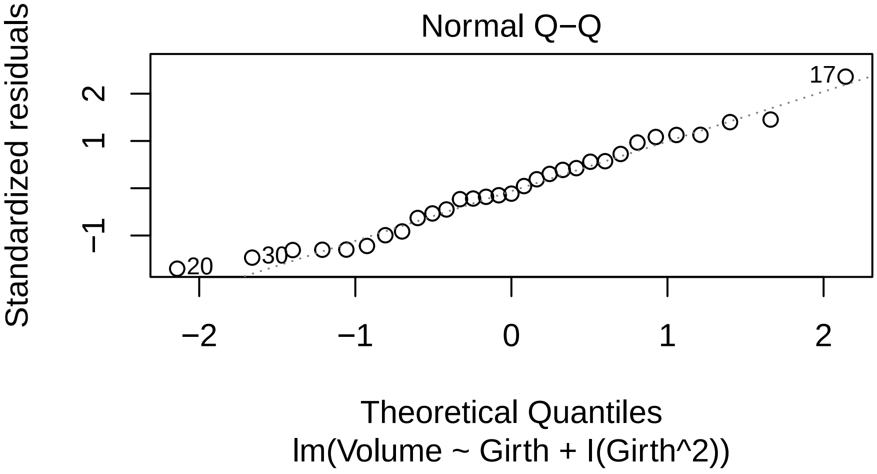
I fit polynomials as follows. The I() syntax is required and lets R know that you are performing a transformation on that variable (here, it is getting squared).

```
fit_trees_poly <- lm(Volume ~ Girth + I(Girth^2), data=trees)
summary(fit_trees_poly)
```

```
##
## Call:
## lm(formula = Volume ~ Girth + I(Girth^2), data = trees)
##
## 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
## Girth       -2.09214    1.64734  -1.270 0.214534
## I(Girth^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
```

```
plot(fit_trees_poly, which=c(1,2))
```

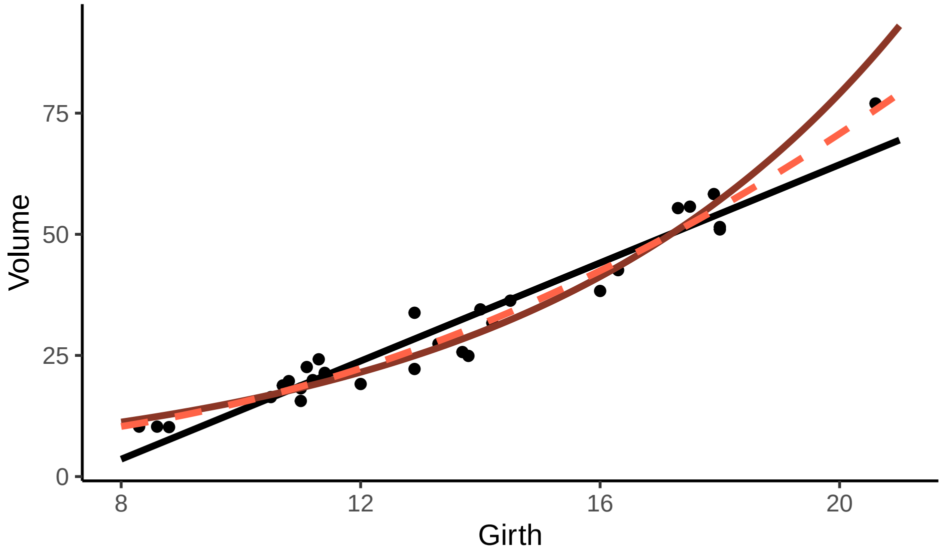




Here is a plot comparing our three fits. The straightline fit is in black, the model with a log transformation is solid tomato, and the model with a second order term is dashed/light tomato.

```
new_data <- data.frame(Girth = seq(8, 21, 0.01))
new_data$Pred_lm  <- predict(fit_trees_1, newdata=new_data)
new_data$Pred_log  <- exp(predict(fit_trees_log_1, newdata=new_data))
new_data$Pred_poly  <- predict(fit_trees_poly, newdata=new_data)

ggplot(data=trees, mapping=aes(x=Girth, y=Volume))+
 geom_point()+theme_classic()+
 geom_line(data=new_data, aes(x=Girth, y=Pred_lm), linewidth=1.2)+
 geom_line(data=new_data, aes(x=Girth, y=Pred_log),   color="tomato4", linewidth=1.2)+
 geom_line(data=new_data, aes(x=Girth, y=Pred_poly),  linetype="dashed", linewidth=1.2,
           color="tomato1")
```



2.1 Interpreting curvilinear fits

When you transform either your response or predictor, the interpretation of the model changes. For example, following a log-transformation, you can make broad statements about the nature of the correlation (positive vs. negative), but a one unit change in X is now a β_1 unit change in $log(Y)$. Be cognizant of these changes as you write up results for publication.