Data transformation answers

Crispin Jordan

09/10/2020

Question 1 - Neurotoxin

Let's start by importing the data:

```
toxin <- read.table("Neurotoxin.csv", header = TRUE, sep = ",")
```

We need to familiarize ourselves with the data. Let's look at the entire dataset (as there are only a handful of datapoints):

```
toxin
```

```
##     Locality Resistance
## 1     Benton       0.29
## 2     Benton       0.77
## 3     Benton       0.96
## 4     Benton       0.64
## 5     Benton       0.70
## 6     Benton       0.99
## 7     Benton       0.34
## 8  Warrenton       0.17
## 9  Warrenton       0.28
## 10 Warrenton       0.20
## 11 Warrenton       0.20
## 12 Warrenton       0.37
```

Part (1)

We see that the columns are called, `Locality`, which notes the location of the two populations of snakes, and `Resistance`, which denotes the authors' measure of a snake's resistance to TTX.

Let's check whether **R** recognizes that `Locality` is a `Factor` (which is what we want):

```
str(toxin)
```

```
## 'data.frame':    12 obs. of  2 variables:
##  $ Locality  : chr  "Benton" "Benton" "Benton" "Benton" ...
##  $ Resistance: num  0.29 0.77 0.96 0.64 0.7 0.99 0.34 0.17 0.28 0.2 ...
```

Nope. We see that `Locality` is type `chr`. Let's change `Locality` to a `Factor`:

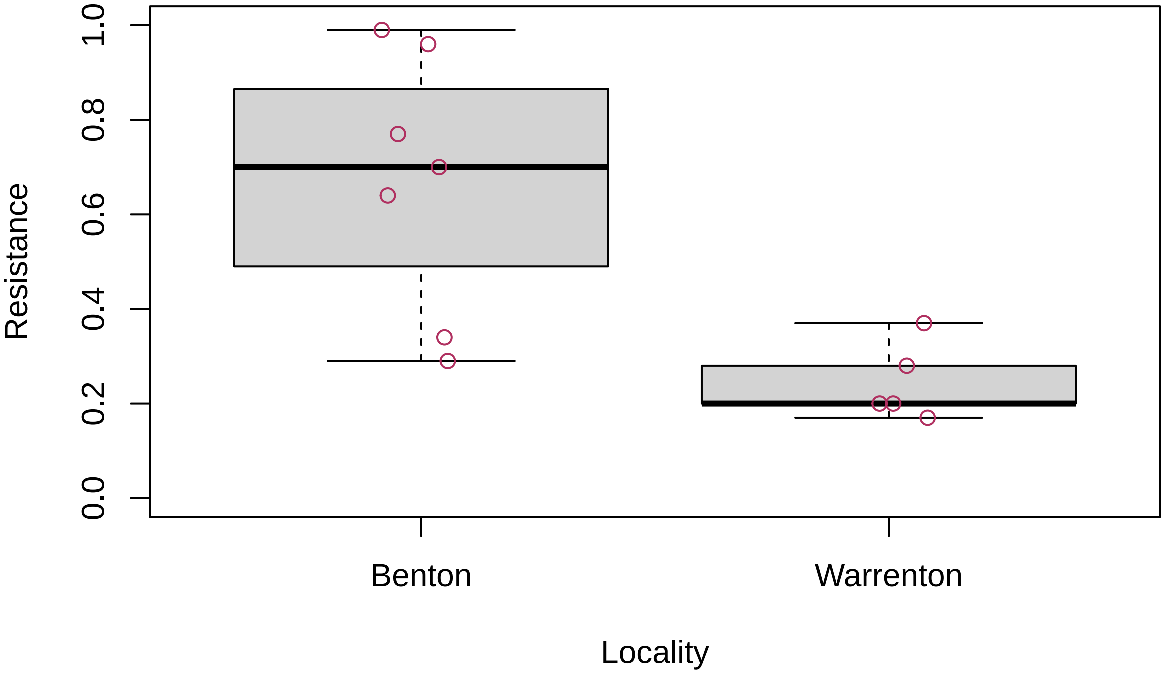
```
toxin$Locality <- factor(toxin$Locality)
str(toxin)
```

```
## 'data.frame':    12 obs. of  2 variables:
##  $ Locality  : Factor w/ 2 levels "Benton","Warrenton": 1 1 1 1 1 1 1 2 2 2 ...
##  $ Resistance: num  0.29 0.77 0.96 0.64 0.7 0.99 0.34 0.17 0.28 0.2 ...
```

Excellent! Now `Locality` is a `Factor` and `Resistance` is `num` (numeric). This is what we want.

Let's now visualize the data to help us become better acquainted with it. To make a plot, we need to decide which variable is the *dependent* variable? Our hypothesis is that populations of snakes that eat the newts will have evolved resistance to TTX. Therefore, we expect the variable `Resistance` to depend on `Locality`; `Resistance` is the dependent variable.

```
boxplot(Resistance ~ Locality, data = toxin, ylim = c(0,1))
stripchart(Resistance ~ Locality, data = toxin, add = TRUE, vertical = TRUE, pch = 21,
col = "maroon", method = "jitter")
```



What do we notice from this output?

1. The breadth of the boxplots seems to differ between the localities, suggesting that the data will not meet the assumption of equal variance.
2. The boxplot for the Benton population is nicely symmetrical, suggesting that the data are normally distributed; this is less obvious for the Warrenton population.
3. No outliers.
4. The distributions for the two populations overlap very little, suggesting that we can expect strong evidence that the population differ.

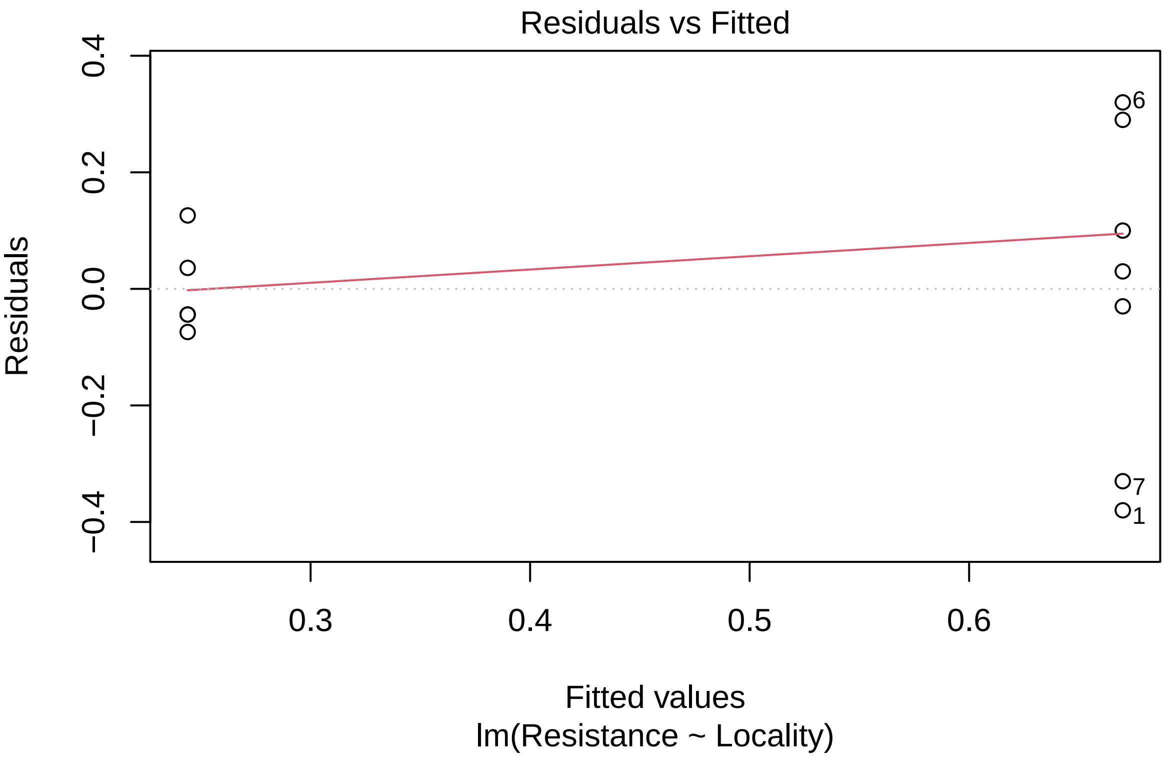
Part (2)

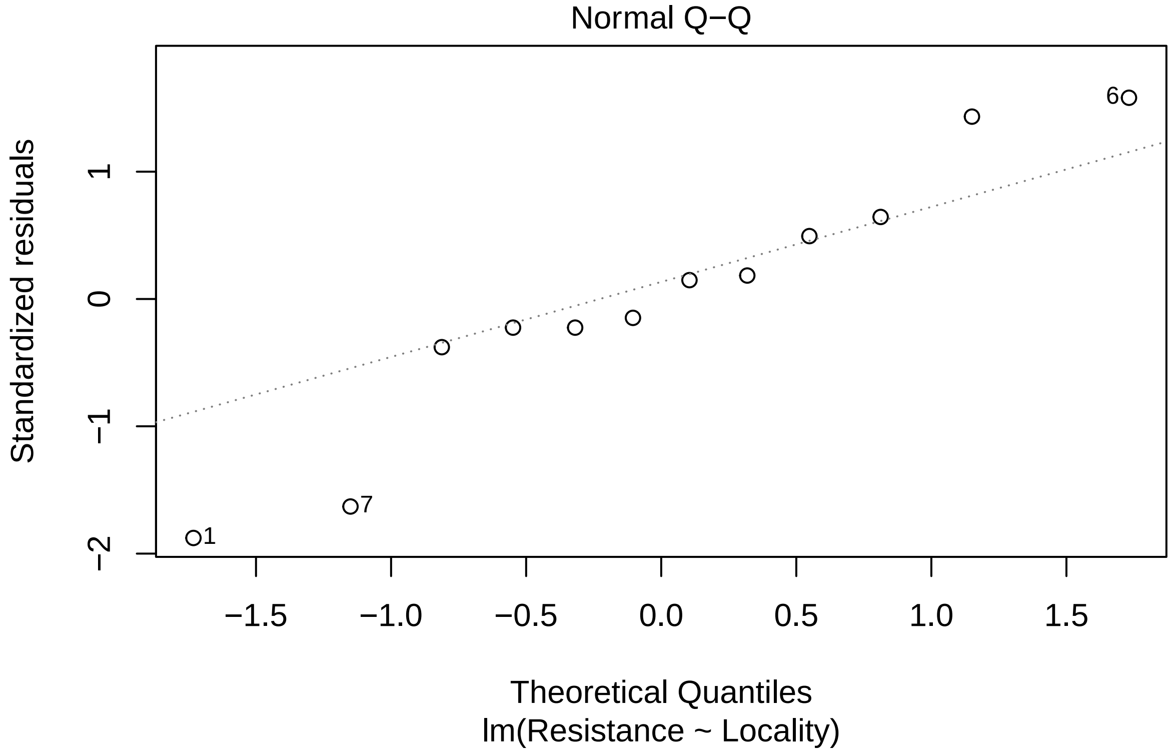
Let's model these data to test the null hypothesis that the mean `Resistance` does not differ between `Locality` levels. We decided, above, that `Resistance` is the dependnent variable. Therefore we can use the model, below. (Please note that we could also use a t-test for this analysis, but our current goal is to practice general linear models…)

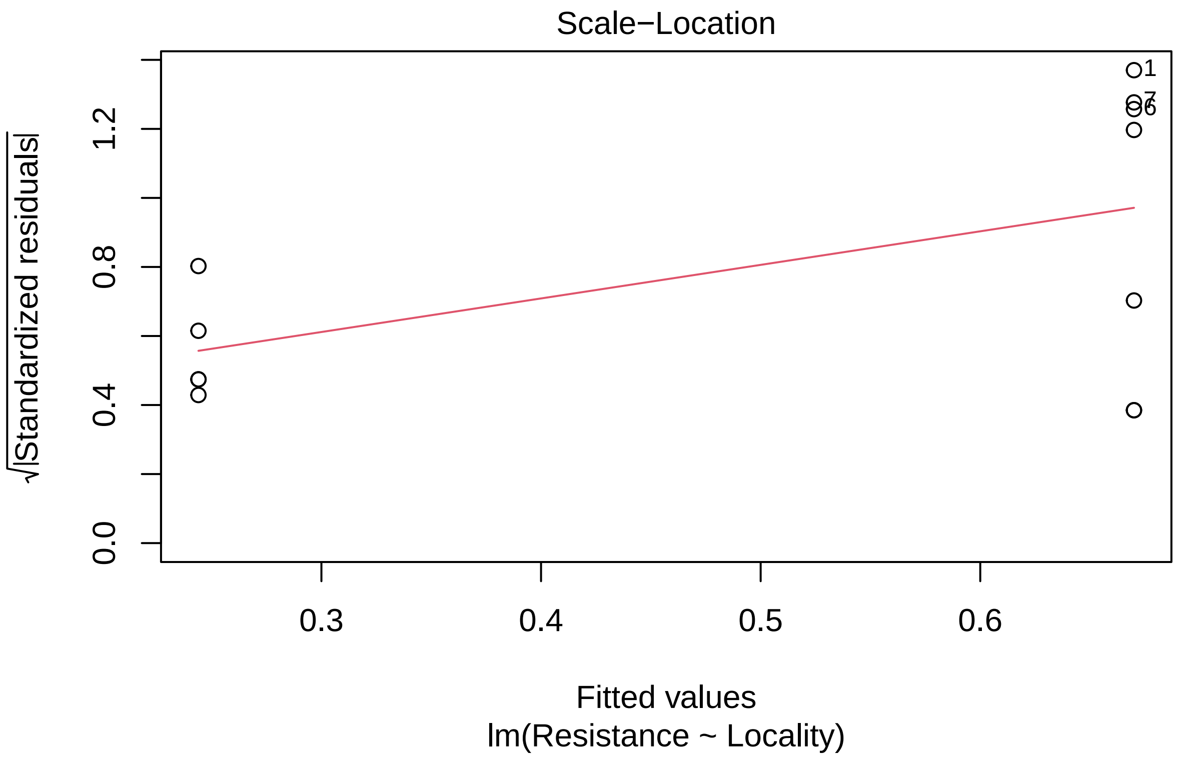
```
toxin.lm <- lm(Resistance ~ Locality, data = toxin)
```

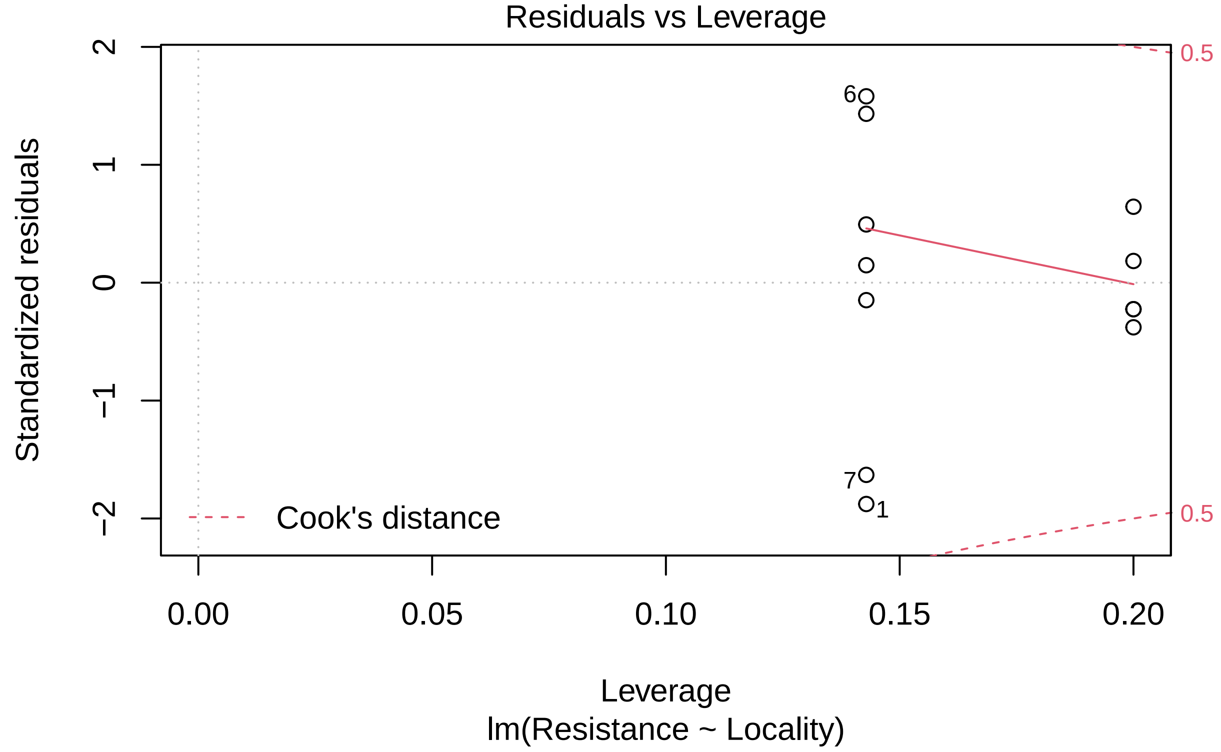
Before looking at the results, we should check the assumptions:

```
plot(toxin.lm)
```









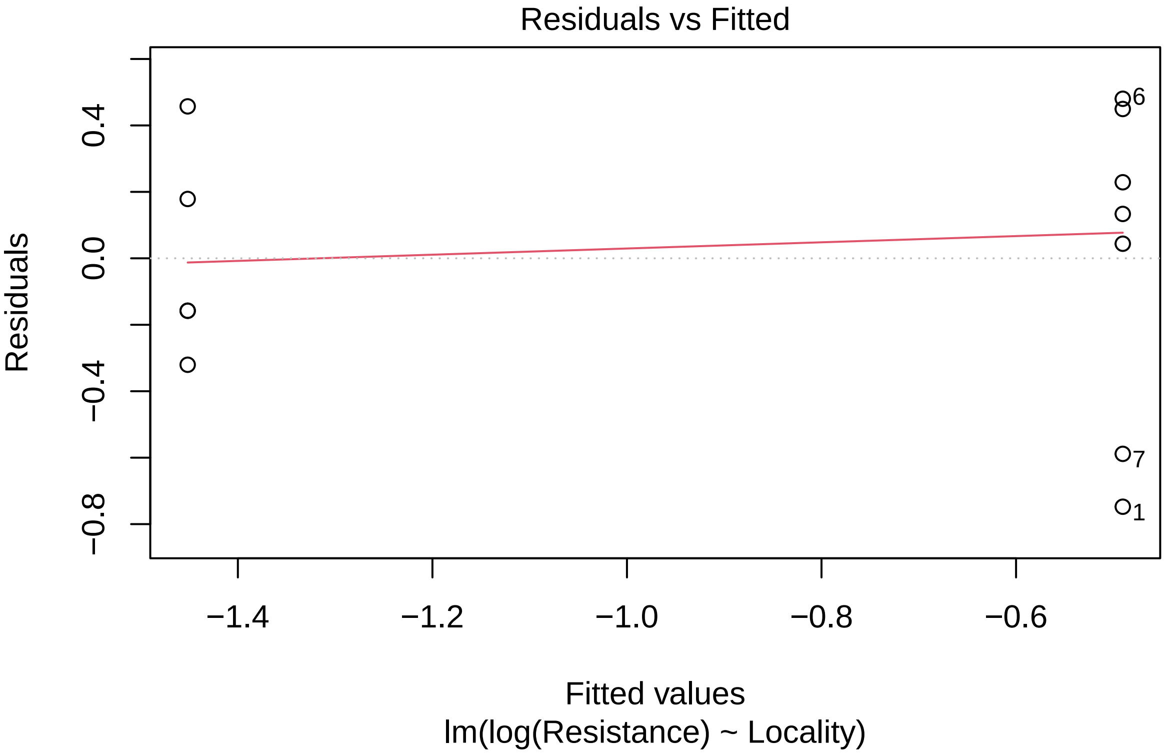
The first plot allows us to test the assumption of equal variance. When looking at the boxplot, we predicted that the data would not meet the assumption of equal variance; this is borne out in the first residual plot. Notice that the spread among the residual points is much less on the left of the plot than for the right; this suggests that our data do not meet the assumption of equal variance. The second plot also suggests a problem with normality. But, unequal variance is a more important assumption than normality, so we'll focus on equal variance, for now. The third plot also indicates that unequal variance (and, in general does so more clearly than the first plot does). For the third plot, notice the red line is not flat, indicating that variance is unequal between levels of `Locality` (the residuals around the line are also more spread out on the right than the left).

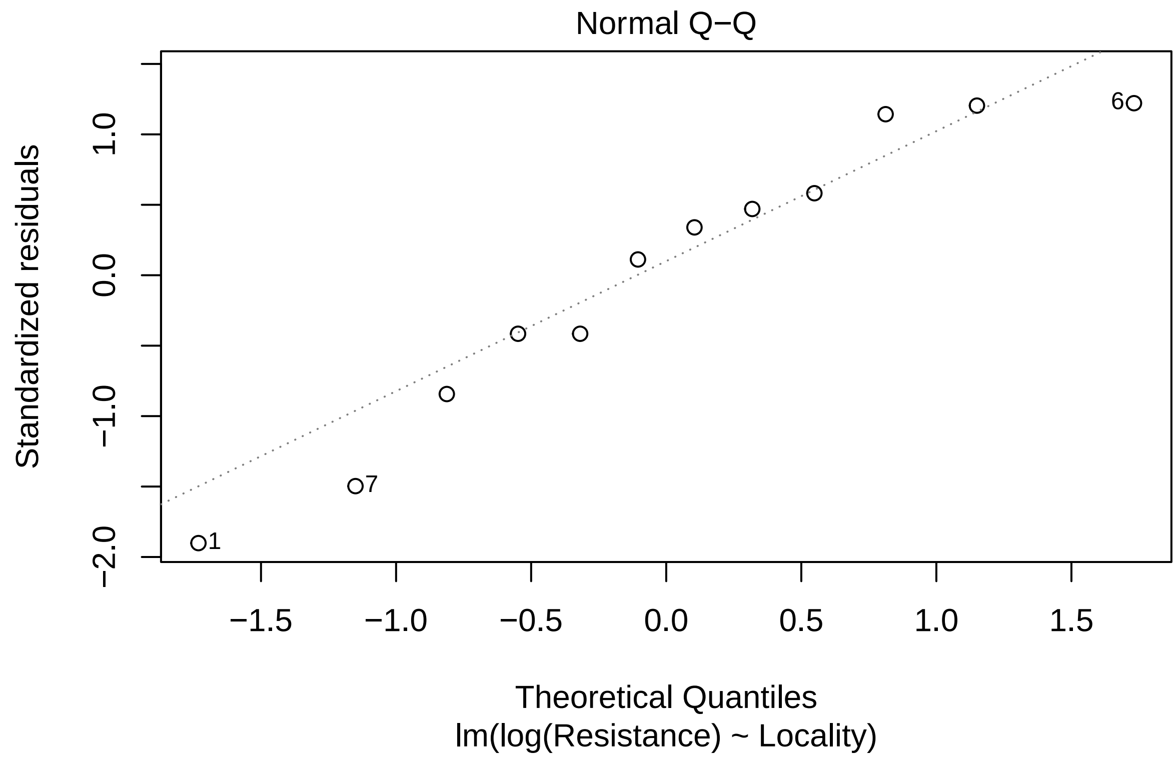
Let's try transforming the data. A `log()` transformation can be useful when variance increases with the mean, which is what we see in these data (see the first residual plot, above). We can implement a `log()` transformation like this:

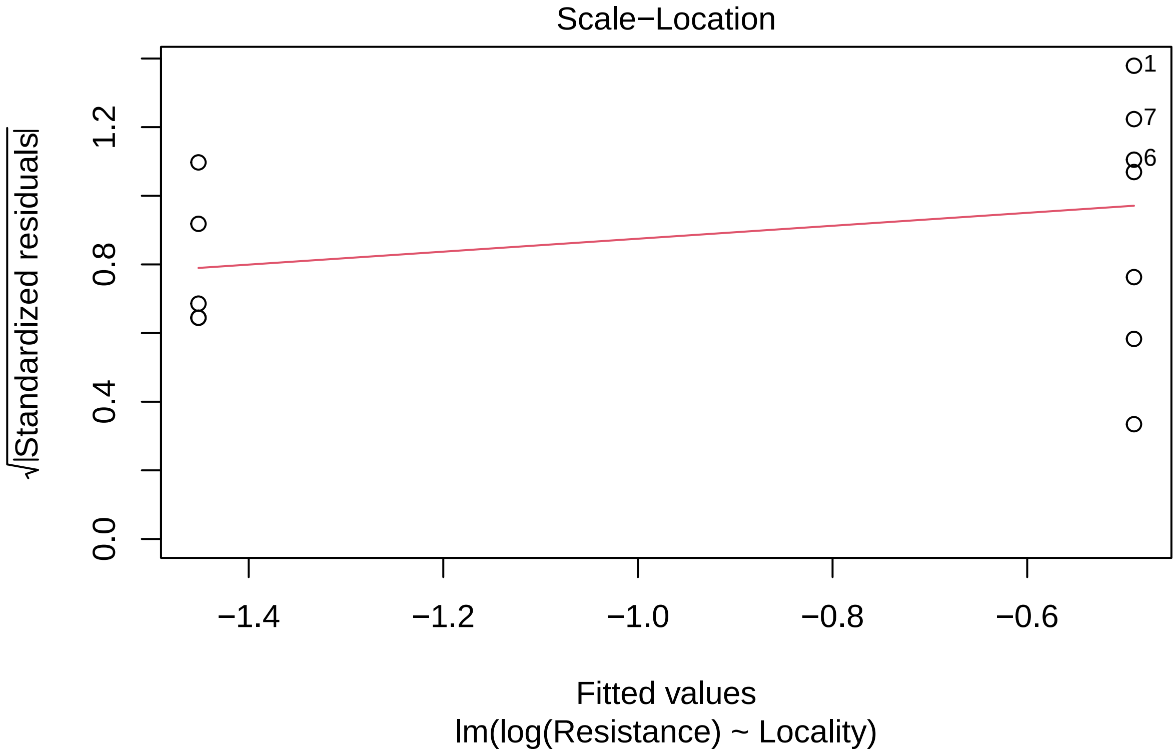
```
toxin.lm.log <- lm(log(Resistance) ~ Locality, data = toxin)
```

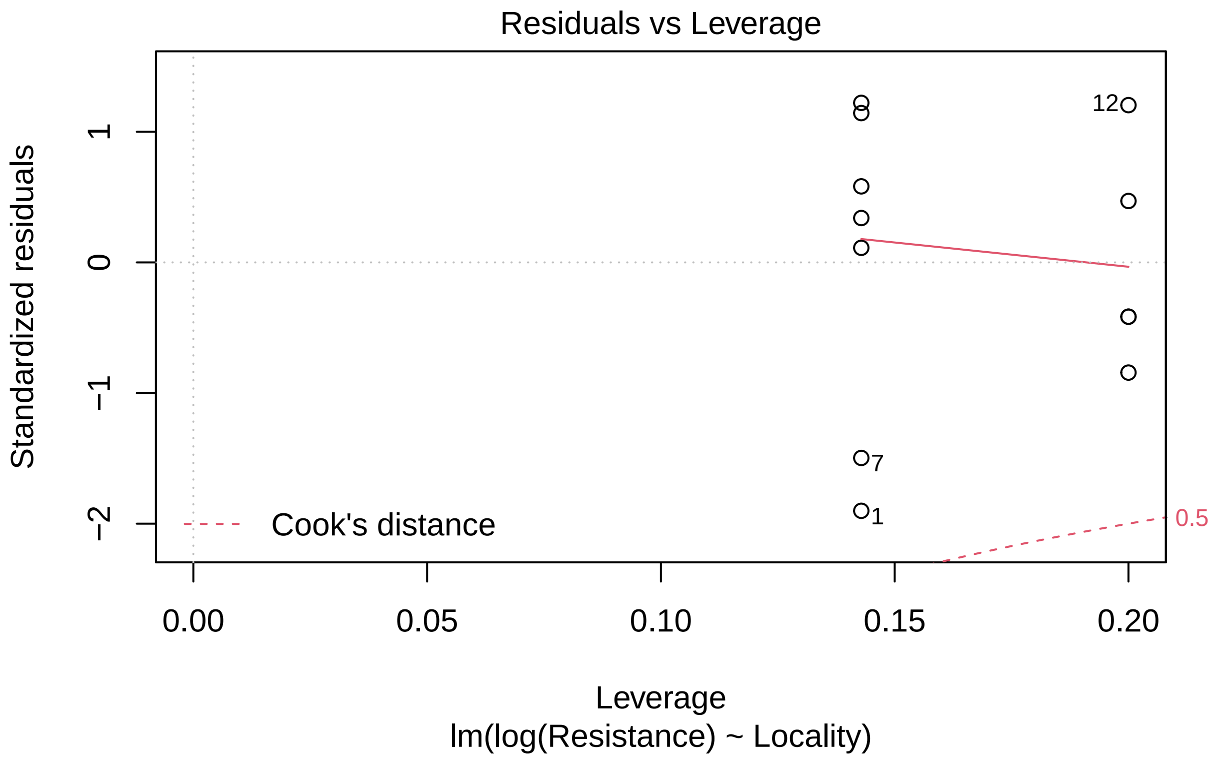
Now, let's check the assumptions again:

```
plot(toxin.lm.log)
```




Normal Q−Q
Standardized residuals
−2.0
−1.0
0.0
1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
1
7
6
Theoretical Quantiles
lm(log(Resistance) ~ Locality)


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
−1.4
−1.2
−1.0
−0.8
−0.6
1
7
6
Fitted values
lm(log(Resistance) ~ Locality)



Part (3)

The first plot shows that log-transforming the `Resistance` data helps to meet the assumption of equal variance; the spread among the residuals on the left is now comparable to that on the right. (Although I have not shown the results here, I also tried a square-root transformation, which gave less satisfactory results). The data also appear generally normally distributed. The third plot suggests that variance is relatively equal between levels of `Locality`. *(Note, however, that the assumptions of equal variance and normality are both difficult to asses with such a small sample size, generally.)* Therefore, the data sufficiently satisfy the assumptions of equal variance and normality. We also know from the original publication that the data are randomly selected and independent. Therefore, we can check our results.

NOTE: if the data had not sufficiently met the assumptions, we would have to try a different transformation. If we failed to find a useful transformation, we would need to consider alternative approaches to analyze the data, such as a randomization test.

Part (4)

Part (4) section a.

We're asked to obtain the degrees of freedom (d.f.) for our analysis. Here's one way of obtaining the d.f.:

```
anova(toxin.lm.log)
```

```
## Analysis of Variance Table
##
## Response: log(Resistance)
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Locality   1 2.6955 2.69552  14.945 0.00313 **
## Residuals 10 1.8036 0.18036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are two types of d.f. to report:

1) d.f. for the among treatment variation (i.e., for the effect of `Locality`), which we see equals 1 (you will recall that d.f. for a treatment equals the number of treatments minus 1).

2) d.f., associated with the residual ('error') variation, which equals 10.

Therefore, we would report our d.f. as, `d.f., = 1,10`. (Notice that we list the residual d.f. second.) As you'll see next, we report the d.f. with the F-value.

Part (4) section b.

We obtain the F-value from the same output. We see that `F = 14.945`. When we report `F` value for our analysis, we include the d.f. information. For example, we can write, $F_{1,10} = 14.945$. Alternatively, we could write, $F(1, 10) = 14.945$.

Part (4) section c.

We can obtain our p-value from the same output, above; the `p = 0.00313`. As the p-value is around 0.005, suggesting that we have strong evidence (Benjamin et al. (2018) https://www.nature.com/articles/s41562-017-0189-z.pdf?origin=ppub) that log of `Resistance` differs between the two levels of `Locality`.

Part (4) section d.

We need our estimate of effect size. In this example, because our experiment has only two treatments (or rather, because there is only one effect size that interests us), we can obtain the effect size in at least two ways. First, we can examine the output of the `summary()` for our object, `toxin.lm.log`, which holds the output of our model when we used log-transformed data:

```
summary(toxin.lm.log)
```

```
##
## Call:
## lm(formula = log(Resistance) ~ Locality, data = toxin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74761 -0.19846  0.08879  0.28404  0.48022
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.4903     0.1605  -3.054  0.01216 *
## LocalityWarrenton  -0.9613     0.2487  -3.866  0.00313 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4247 on 10 degrees of freedom
## Multiple R-squared:  0.5991, Adjusted R-squared:  0.559
## F-statistic: 14.95 on 1 and 10 DF,  p-value: 0.00313
```

To interpret this output, we must understand the `(Intercept)`. Recall that the independent variable, `Locality`, has only two levels: `Benton` and `Warrenton`. Here the `(Intercept)` refers to `Benton` (can you see why?); therefore, the `Estimate` of the `(Intercept)` (i.e., the mean value of `Benton`) equals -0.4903. Now, stop and ask yourself, "Does this estimate make biological sense?" Remember that the dependent variable, `Resistance`, referred to a (standardized) quantity of toxin that cause a snake's mobility to drop by 50%: is it possible to have a *negative* amount of toxin (see -0.4903)? Clearly this is impossible. But, this value makes sense when we remember that the model used log-transformed values of `Resistance` (the log of a number less than 1 equals a negative number).

Anyway, our goal here was to find the effect size and the SE for the effect size. Where do we find this? The effect size will be the `Estimate` value for the second row: recall that this value equals the difference between the mean of the treatment represented on the second row (i.e., Warrenton) and the (Intercept) (i.e., the mean Benton). Therefore, this Estimate, which equals -0.9613, represents the difference between the mean values of Resistance for the two Localities (based on log-transformed data); this is the 'effect size' of Locality. The negative sign indicates that the mean Resistance for Warrenton is less than for Benton; when reporting an effect size, we can drop the negative sign. This same line of output provides an estimate of the standard error for this effect size, being 0.2487. Therefore, we can report the effect size and its SE as 0.9613 (we dropped the negative sign) and 0.2487, respectively, and we remember to note that these values are on the log-scale.

Alternatively, we could use emmeans() to obtain the effect size and its SE:

```
library(emmeans)
toxin.log.emmeans <- emmeans(toxin.lm.log, "Locality")
toxin.log.emmeans
```

```
##  Locality  emmean    SE df lower.CL upper.CL
##  Benton     -0.49 0.161 10   -0.848   -0.133
##  Warrenton  -1.45 0.190 10   -1.875   -1.028
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Please note that, as a step towards obtaining the effect size, we can report the mean Resistance and SE of each level of Locality with this output above, remembering to report that these values are on the log-scale.

To obtain the effect size and its SE, we use the pairs() function:

```
pairs(toxin.log.emmeans)
```

```
##  contrast           estimate    SE df t.ratio p.value
##  Benton - Warrenton    0.961 0.249 10 3.866   0.0031
##
## Results are given on the log (not the response) scale.
```

Here, we see that the effect size and its SE are identical to what we obtained from the output of summary(toxin.lm.log). Again, we can, and should report this effect size result and note that it is on the log-scale. The SE of the effect size is essential to allow others to include your results in a meta-analysis.

Note that is is also worthwhile to report a 95% CI for the effect size. The 95% CIs for the effect size allow us to infer a range of reasonable effect sizes for our data. This is easily obtained using the confint() functions in the emmeans library:

```
confint(pairs(toxin.log.emmeans))
```

```
##  contrast           estimate    SE df lower.CL upper.CL
##  Benton - Warrenton    0.961 0.249 10    0.407     1.52
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Part (4) section e.

Instead of reporting results on the log-scale, we can back-transform the data to the original data scale. This can be done by adding the type = "response" option to the emmeans() function, as done here:

```
toxin.back.emmeans <- emmeans(toxin.lm.log, "Locality", type = "response")
toxin.back.emmeans
```

```
##  Locality  response     SE df lower.CL upper.CL
##  Benton       0.612 0.0983 10    0.428    0.876
##  Warrenton    0.234 0.0445 10    0.153    0.358
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

The means, SE's and CI's, above are back-transformed to the original data scale. However, if you report these values you must indicate that these are *generalized* means, not arithmetic means; in the case of a log-transformation, the mean values will be geometric means.

Using the output from `emmeans()`, we can obtain back-transformed effect sizes (and their SE's) like this:

```
pairs(toxin.back.emmeans)
```

```
##  contrast           ratio   SE df null t.ratio p.value
##  Benton / Warrenton  2.62 0.65 10    1 3.866   0.0031
##
## Tests are performed on the log scale
```

And, we can obtain 95% CI's for the effect sizes like this:

```
confint(pairs(toxin.back.emmeans))
```

```
##  contrast           ratio   SE df lower.CL upper.CL
##  Benton / Warrenton  2.62 0.65 10      1.5     4.55
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
```

It is often more intuitive to interpret results on the back-transformed scale. However, one draw-back of the approach we used to back-transform is that means (and their SE's, and effect sizes and their SE's and 95% CI's) are all based on generalized means, which is often not a scale that we often expect intuitively (unless you're very well versed in data analysis!). For this reason, if you choose to report back-transformed values, always clearly report how they were back-transformed and how this affects the reader's interpretation. (Note that methods do exist to obtain back-transformed arithemetic mean values, but we do not deal with them here).

Part (4) section f.

Interpreting your statistical results in terms of biology is critical, as this allows you to communicate your insights into biology.

So, what do these results mean, biologically? We'll focus on the back-transformed effect size and its 95% CI's in our interpretation, but noting that the scale might not reflect exactly what we'd like.

The (back-transformed) effect size is a ratio, with `Benton / Warrenton`; the mean of this ratio equals 2.62. This mean is our best estimate of the effect size, and it implies that the mean `Resistance` value for `Benton` is 2.62 times greater than that for `Warrenton`. Depending on your point of view, that may be viewed as a reasonably large difference in `Resistance`. Moreover, the 95% CI's for the effect size indicate that it would be reasonable to say that `Resistance` in `Benton` snakes might be as little as 1.5 times that as `Warrenton`, or as large as 4.55 times that of `Warrenton`.

Note, however, that this interpretation does not tell us whether the *absolute* level of `Resistance` is high or low in each `Locality`. This is because our interpretation has been based on relative differences (a ratio); note however, that even two small numbers (e.g., 0.0000002 and 0.0000001) can create a large ratio (e.g., 0.0000002 / 0.0000001 = 2). Therefore, to know whether the mean levels of `Resistance` in `Benton` and `Warrenton` (0.612 and 0.234, respectively) are high vs. low in a biological sense will require knowledge about neurotoxicity that lies beyond my expertise!

As a final note, remember that we should report the following in a report or paper:

- plot the data (showing individual value if possible)

- means and SE's or 95% CI's for groups
- name of the test used
- test statistic (F-value in our case)
- d.f.
- p-value for the overall test
- p-values, d.f., and test statistic (t.ratio, above) for post-hoc tests
- effect size(s) and its SE (essential for other to use your data in a meta-analysis)
- effect size(s) and its 95% CI to aid interpretation of the results: is the effect size interesting?