S631 HW11

Erik Parker

November 27th, 2017

1. ALR 8.2: Reconsider the stopping distance data used in problem 7.6.

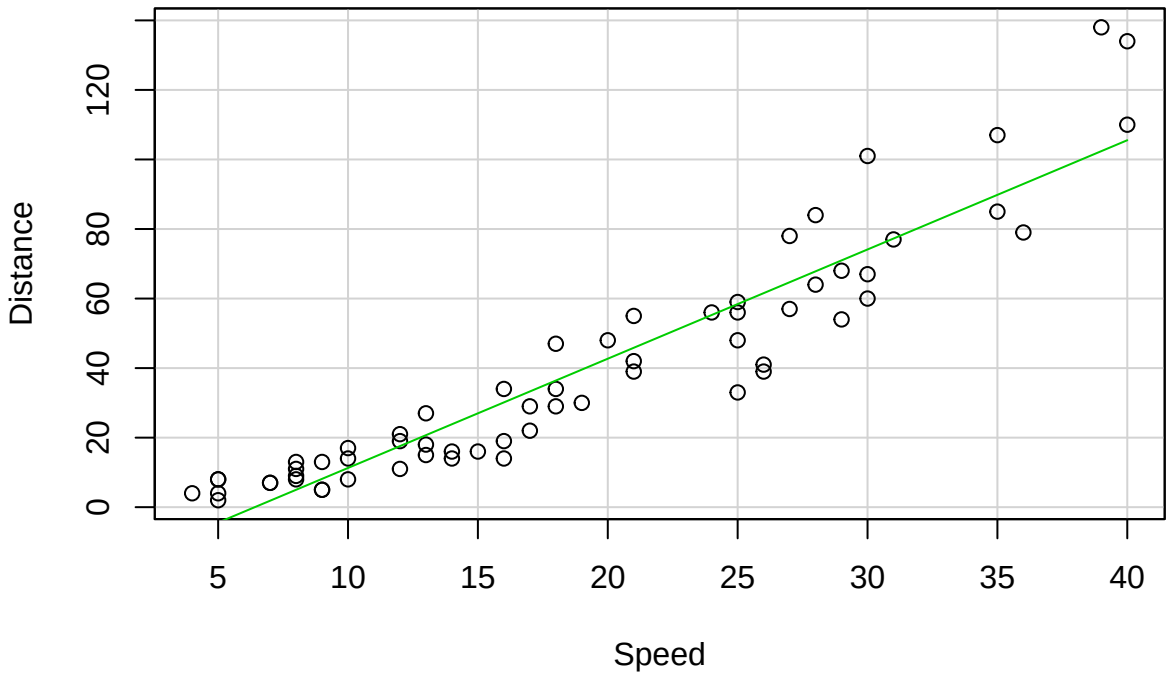
```
rm(list = ls())

library(alr4)

stopping <- stopping
```

8.2.1. Using *Speed* as the only regressor, find an appropriate transformation for *Distance* that can lienarize this regression.

```
summary(stopping)
```

```
##      Speed          Distance
##  Min.   : 4.00   Min.   :  2.00
##  1st Qu.:10.00   1st Qu.: 13.25
##  Median :17.50   Median : 29.50
##  Mean   :18.92   Mean   : 39.31
##  3rd Qu.:26.75   3rd Qu.: 56.75
##  Max.   :40.00   Max.   :138.00
```

```
scatterplot(Distance ~ Speed, data = stopping, boxplots = FALSE, smooth = FALSE)
```

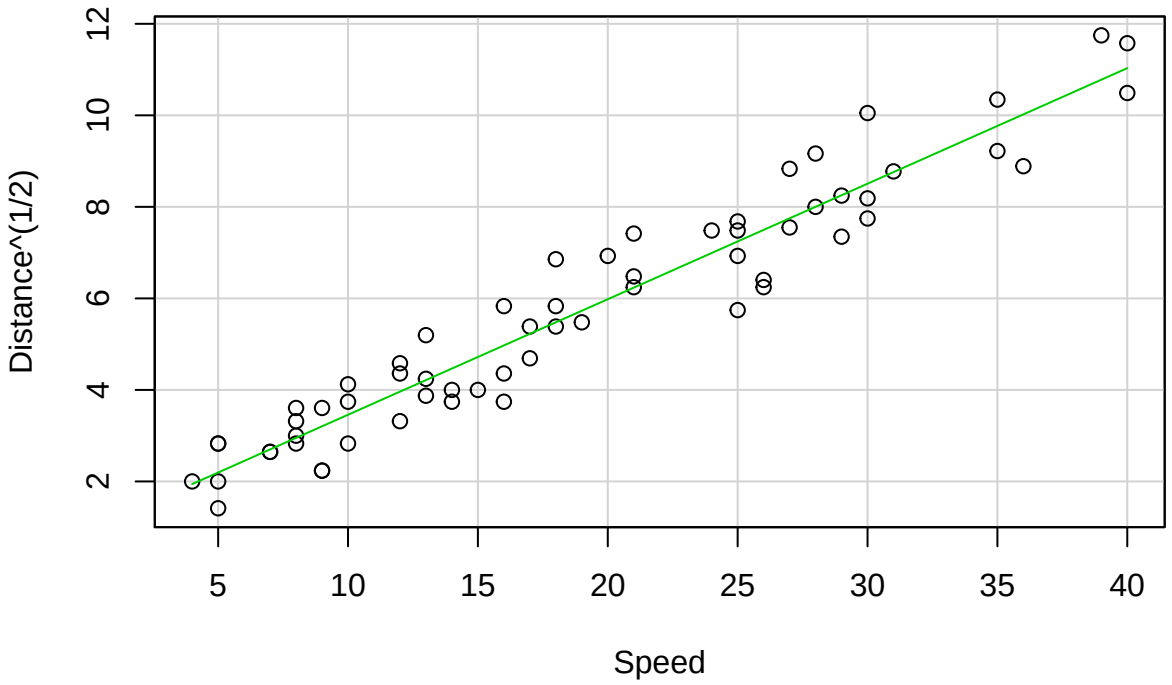



```
summary(powerTransform(lm(Distance ~ Speed, data = stopping)))
```

```
## bcPower Transformation to Normality
```

```
##    Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
## Y1    0.4163         0.5       0.2885       0.5441
##
## Likelihood ratio tests about transformation parameters
##                             LRT df         pval
## LR test, lambda = (0) 33.67953  1 6.498075e-09
## LR test, lambda = (1) 60.09488  1 8.992806e-15
```

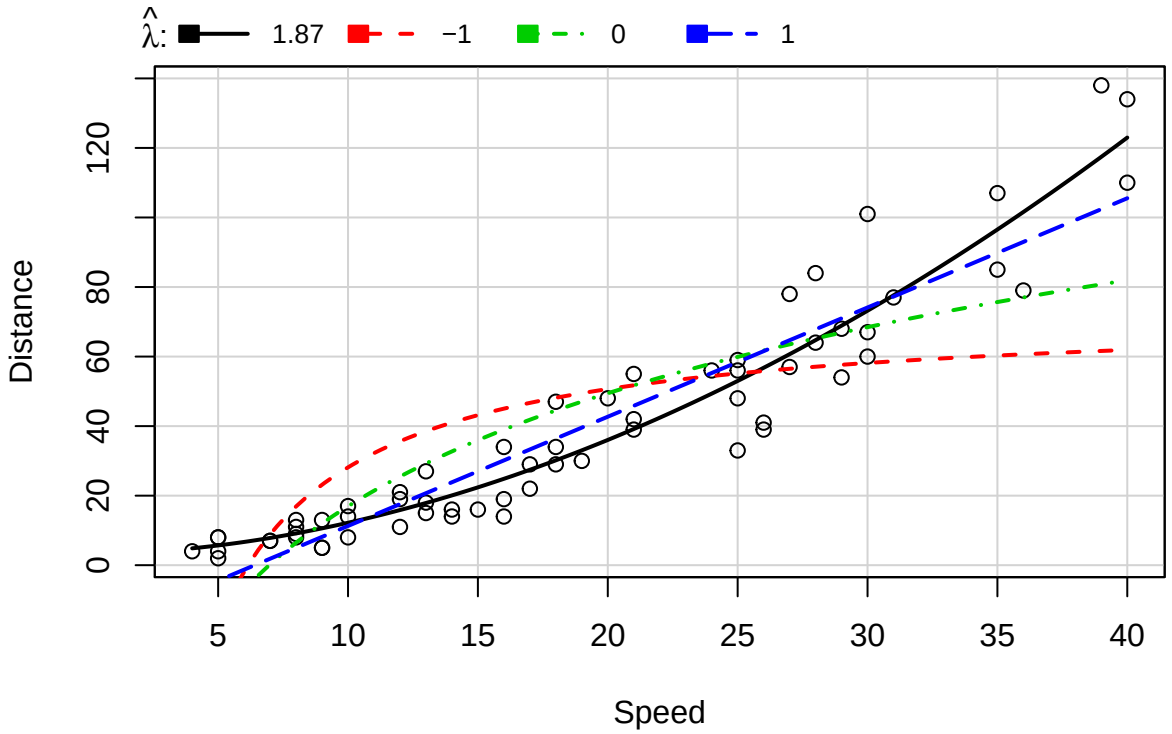
```
scatterplot(Distance^(1/2) ~ Speed, data = stopping, boxplots = FALSE, smooth = FALSE)
```



So, the best transformation for *Distance* seems to be the 1/2 power, square root, transformation.

8.2.2. Using *Distance* as the response, transform the predictor *Speed* using a power transformation with each $\lambda \in -1, 0, 1$ and show that none of these transformations is adequate.

```
with(stopping, invTranPlot(Speed, Distance))
```




```
##      lambda       RSS
## 1  1.868443  5823.372
## 2 -1.000000 34951.108
## 3  0.000000 18844.172
## 4  1.000000  8310.166
```

Using the invTranPlot() command, we are able to generate the plot above which shows the $\hat{\lambda}$ which best minimizes the RSS_λ for our mean function outlined in ALR 8.4, in addition to other potential values for λ, here -1, 0, and 1. We can see from this plot quite clearly that none of those three values of λ for the varaible *Speed* adaquately transforms and linearizes the mean function. Specifically, we see that if we use the untransformed regressor, *Distance*, the most appropriate transformation for *Speed* is 2.

8.2.3. Show that using $\lambda = 2$ does match the data well. This suggests using a quadratic polynomial for regressors, including both *Speed* and $Speed^2$.

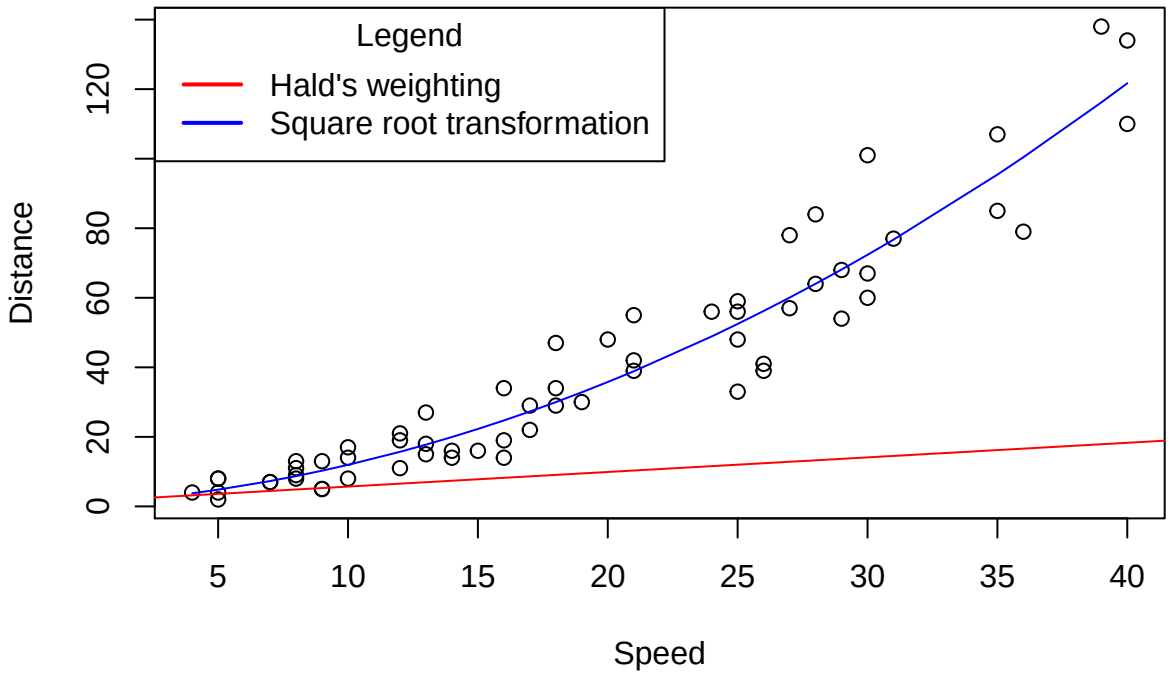
```
m1 <- lm(Distance ~ Speed, data = stopping)

m2 <- lm(Distance ~ Speed + I(Speed^2), data = stopping)

anova(m1, m2)
```

```
## Analysis of Variance Table
##
## Model 1: Distance ~ Speed
## Model 2: Distance ~ Speed + I(Speed^2)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     60 8310.2
## 2     59 5814.1  1      2496 25.329 4.835e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Above in section 8.2.2, we can see from the transformation plot that $\hat{\lambda} = 1.87$, meaning the nearest rounded value is $\lambda = 2$. This provides some support for using a quadratic polynomial for the regressor, *Speed*. Further support for this can be seen by the anova performed between the two models, m1 and m2, where m2 contains the $Speed^2$ term. This anova finds that we can reject the null hypothesis that the reduced mean function, without the quadratic polynomial of *Speed*, is sufficient to explain the response, and so we can say that the quadratic transformation is useful.

8.2.4. Hald (1960) suggested on the basis of a theoretical argument using a quadratic mean function for *Distance* given *Speed*, with $Var(Distance|Speed) = \sigma^2 Speed^2$. Draw the plot of *Distance* versus *Speed*, and add a line on the plot of the fitted curve from Hald's model. Then obtain the fitted values from the fit of the transformed *Distance* on *Speed*, using the transformation you found in Problem 8.2.1. Transform these fitted values to the *Distance* scale (for example, if you fit the regression sqrt(*Distance*) ~ *Speed*, then the fitted values would be in square-root scale and you would square them to get the original *Distance* scale). Add to your plot the line corresponding to these transformed fitted values. Compare the fit of the two models.

```
m3 <- lm(Distance ~ Speed + I(Speed^2), data = stopping, weights = 1/(Speed^2))

m4 <- lm(sqrt(Distance) ~ Speed, data = stopping)

plot(Distance ~ Speed, data = stopping)
abline(m3, col = "red")
lines(stopping$Speed, fitted(m4)^2, col = "blue", pch = 20)
legend("topleft", title = "Legend", c("Hald's weighting", "Square root transformation"),
    lty = c(1, 1), lwd = c(2, 2), col = c("red", "blue"))
```



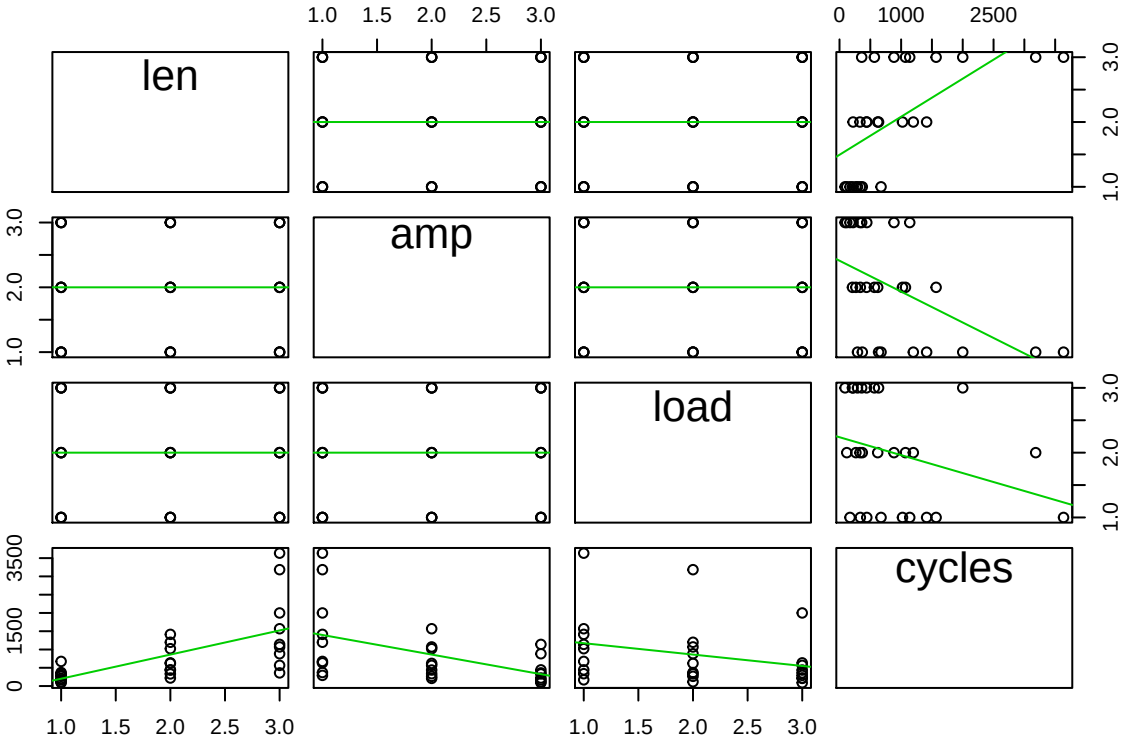
From the above plot, it is clear that the earlier identified square root transformation for the response, *Distance*, is far superior in terms of improving the fit of the mean function to the data than is Hald's suggestion of weighting by $Speed^2$.

2. ALR 8.6: These data were introduced in Section 5.2. For this problem, we will start with *cycles*, rather than its logarithm, as the response. Remember that you may need to declare *len*, *amp*, and *load* as factors.

```
wool <- Wool

wool$len <- as.factor(wool$len)
wool$amp <- as.factor(wool$amp)
wool$load <- as.factor(wool$load)
```

8.6.1. Draw the scatterplot matrix for these data and summarize the information in this plot.

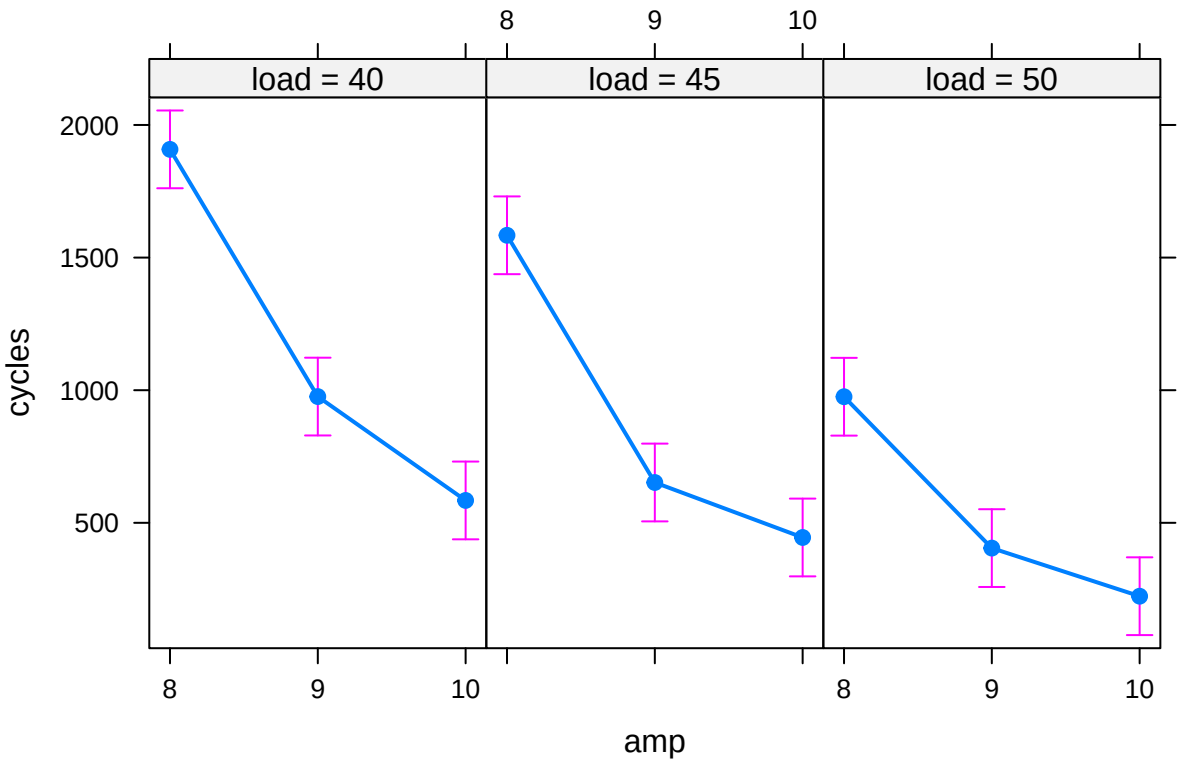
```
scatterplotMatrix(wool, smooth = FALSE, diagonal = "none")
```



In this plot, the most meaningful information is contained on the bottom row, where cycles is on the y-axis and all the factors are on the x-axis. From left to right we can first see that as the length of the wool specimen increases, so too does the number of cycles required until its failure. Next, as the amplitude of the loading cycle increased, the number of cycles needed until the failure of the sample decreased. Finally, as the load weight increases, the number of cycles until failure decreases.

8.6.2. View all three predictors as factors with three levels, and without transforming *cycles*, fit the second-order mean function with regressors for all main effects and all two-factor interactions. Summarize results of the *amp* by *load* interaction with an effects plot.

```
m1 <- lm(cycles ~ len + amp + load + len:amp + len:load + amp:load, data = wool)

plot(Effect(c("amp", "load"), m1))
```



From the effects plot of the interaction between *amp* and *load* on *cycles*, we can clearly see that in general as both *load* and *amp* increase individually (holding the other constant), the number of cycles until the failure of the specimen decreases. Furthermore though, with this plot we can also visualize the interaction between the two regressors - specifically, the magnitude of the decrease in *cycles* seen as *amp* increases is lower at increasing values for *load*.

8.6.3. Fit the first-order mean function consisting only of the main effects. From Problem 8.6.2, this mean function is not adequate for these data based on using *cycles* as the response because the tests for each of the two-factor interactions indicate that these are likely to be nonzero. Use the Box–Cox method to select a transformation for *cycles* based on the first-order mean function.

```
m2 <- lm(cycles ~ len + amp + load, data = wool)

summary(powerTransform(m2))
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
## Y1   -0.1005           0      -0.2249       0.0239
##
## Likelihood ratio tests about transformation parameters
##                             LRT df      pval
## LR test, lambda = (0)  2.38372  1 0.1226053
## LR test, lambda = (1) 83.89818  1 0.0000000
```

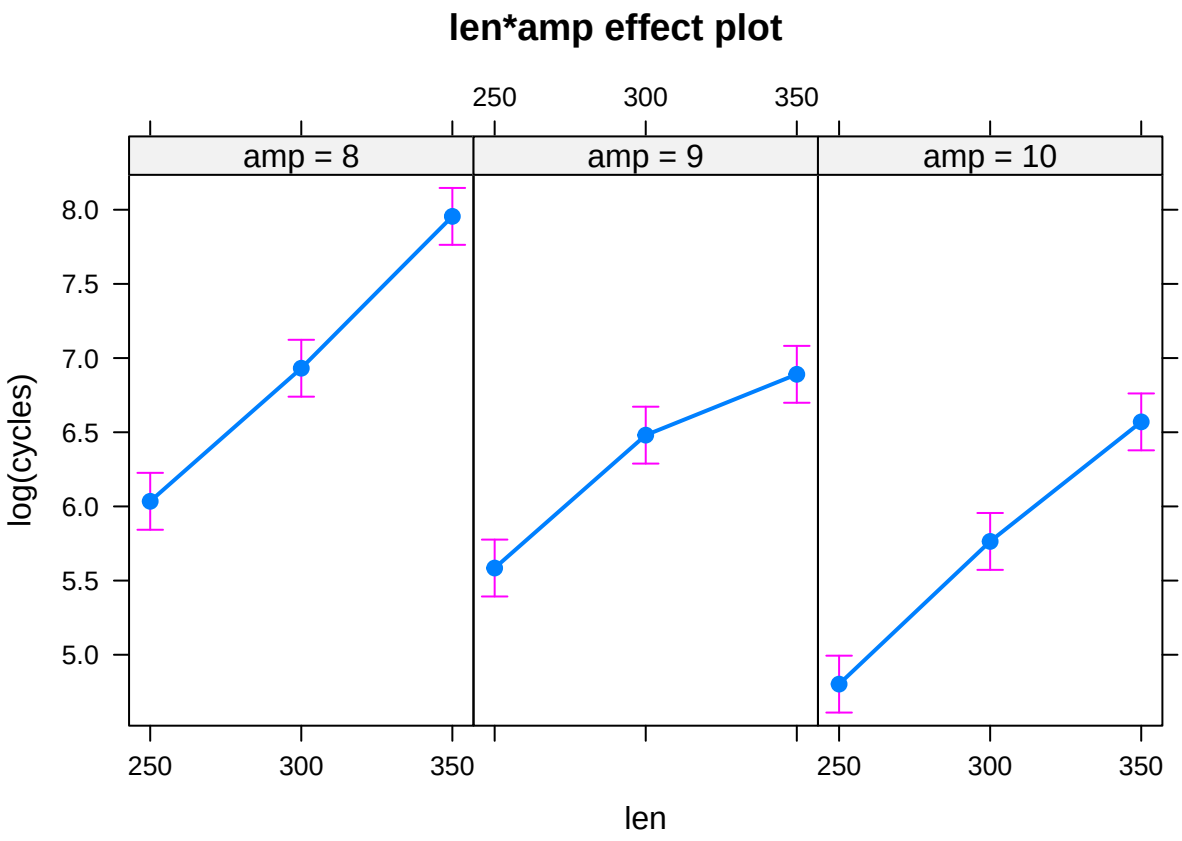
Using the Box-Cox method, we see that the best transformation of *cycles* for the first-order mean function is the 0 power, the log() transformation.

8.6.4. In the transformed scale, fit both the first-order model and the second-order model, and compute an F-test comparing these two models. This is a nonstandard test because it is simultaneously testing all interactions to be equal to zero. Then provide an effects plot for the *len* by *amp* interaction. This will of course be three parallel lines. Then redraw this effects plot with *cycles* rather than *log(cycles)* on the horizontal axis, and compare with the effects plot you drew in Problem 8.6.2.

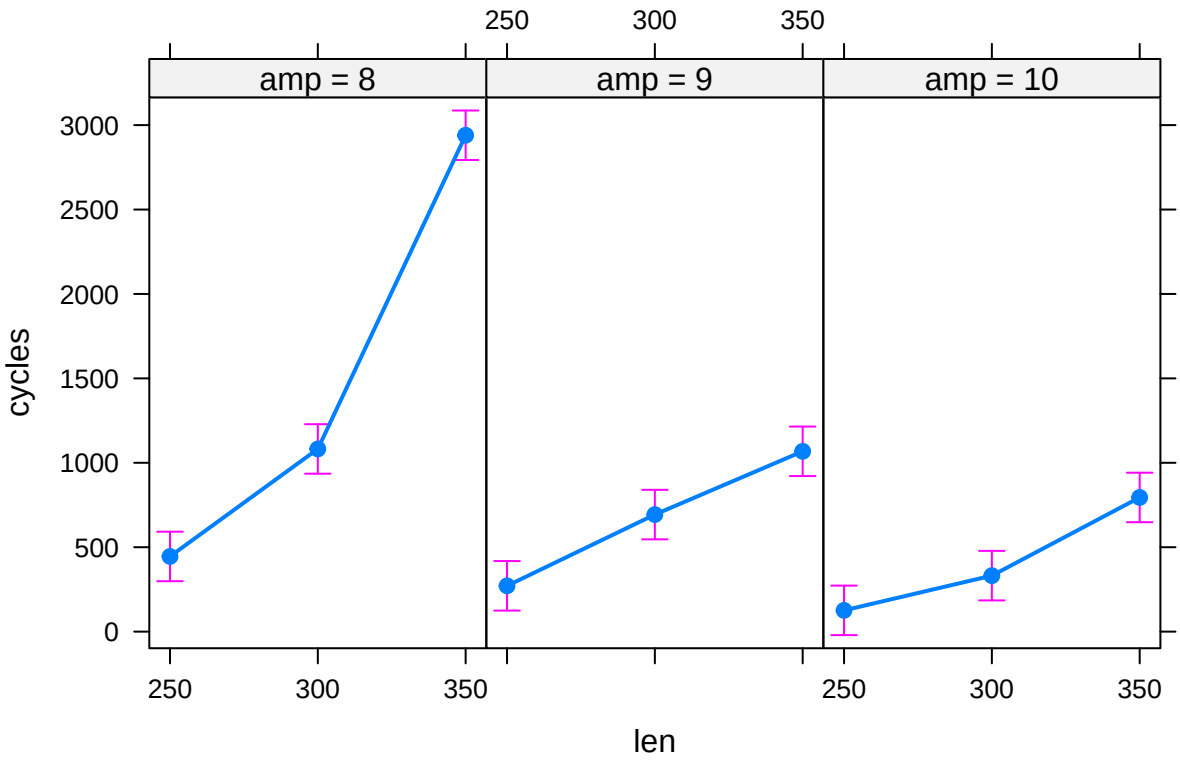
```
m1l <- lm(log(cycles) ~ len + amp + load + len:amp + len:load + amp:load, data = wool)

m2l <- lm(log(cycles) ~ len + amp + load, data = wool)

anova(m2l, m1l)
```

```
## Analysis of Variance Table
##
## Model 1: log(cycles) ~ len + amp + load
## Model 2: log(cycles) ~ len + amp + load + len:amp + len:load + amp:load
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1     20 0.71742
## 2      8 0.16591 12   0.55151 2.216 0.1325
```

```
plot(Effect(c("len", "amp"), m1l))
```




```
plot(Effect(c("len", "amp"), m1))
```



First, we can see from the p-value of 0.1325 from the F-test that we can't reject the null hypothesis that the reduced model not containing any interactions sufficiently explains the variation seen response, so the second order interactions do not need to be added to the mean function.

Next, when we compare the effect plot of the interaction between *len* and *amp* on untransformed *cycles* to the earlier plot of *amp* and *load* on *cycles*, we can see that there are some clear differences. Compared to the earlier plot, we can see that this new one more clearly uncovers an interaction between the regressors, *len* and *amp*. This is most clear when we examine the plots of *len* on *cycles* for $amp = 8$ and either 9 or 10. When *amp* is 9 or 10, there is an essentially linear increase in the number of cycles until failure as the length of the specimen increases; but at an *amp* of 8, we see that when the length of the specimen is 350mm the number of cycles until its failure is much higher than for the other values of *len*. This particular outlying point then makes it easy to see that there is an interaction between the regressors which needs to be accounted for.