Paul Johnson
Nonlinear Regression: Log transformations, Box-Cox transforms

1 The Problem: Licensing Older Drivers

In a current project, we are investigating the impact of state drivers licensing laws on highway fatalities among older drivers. This was prompted by a few highly publicized incidents of elderly drivers who lost control of their car and crashed into crowded markets (or such).

The data on crashes among older drivers is not available for all U.S. states for each year. Through the 1990s, only 17 states reported that data. However, for one year, 1998, the NHTSA did provide older driver fatality data for all states, along with other useful information about the older population.

In each year from 1994-2001, the NHTSA does report the number of older drivers who have driver's licenses. So I wondered if we could not just use the number who have licenses as an indicator for the effectiveness of state laws on drivers license tests. Presumably, the number who are licensed will be closely related to the danger that results on the road.

I can make a test of the usefulness of the licensing levels by examining the 1998 driver fatality data. Maybe one can argue that policy effectiveness depends only on "keeping the drivers licenses away from the seniors" so we could use the 1994-2001 data.

That purpose may or not be served, but it did lead to some interesting methodological observations about nonlinear regression. I put together a 2 variable dataset "fatalLicenses.txt" and that should be available in the POLS7070 web material. That data has the 1998 figures on the licensed elderly drivers (called *licenses* in the data) and fatalities among older drivers (*fatals*). I read it into R with this command:

```
dat <- read.table("fatalLicenses.txt",header=T)
```

Consider a plot of the data on older driver fatalities and the number of older licensed drivers, which is shown in Figure 1 .

2 Is $y = b_0 + b_1 x + e$?

Clearly there is a relationship. If you run a linear regression, these are the estimates in R:

Coefficients	OLS estimate	Std.Error		
Intercept	16.6*	6.6		
Older licenses	0.0003977*	0.0000282		
$R^2 = 0.80$	adj R^2=0.79	N=51	Residual Std.Error=33.7	
* $p \leq 0.05$				

The regression purist will say "there's heteroskedasticity in there, try logging the variables." Logging does make it look more homoskedastic. As you see in Figure 2, it does look much better.

Still, there are some shortcomings. To my eye, this scatter appears somewhat curved (concave downward). There's a case which appears to be an outlier at the coordinates (11,1), and that ought to be looked into.

Nevertheless, I could be happy if I were on a deserted island and had only this one scatterplot and could estimate just the one regression:

$$log(fatalities_i) = b_0 + b_1 older licenses_i + \varepsilon_i \qquad (1)$$

The parameter estimates of that regression are:

Coefficients	OLS estimate	Std.Error		
Intercept	-8.01*	0.85		
log(older licenses)	1.032*	0.07		
$R^2 = 0.80$	adj R^2=0.80	N=51	Residual Std.Error=0.53	
* $p \leq 0.05$				

Recall that the equation estimated as 1 represents an exponential relationship of this form

$$fatalities_i = e^{b_0} (older licenses_i)^{b_1} * e^{\varepsilon_i}$$

Figure 1: Older Drivers: Fatalities and Licenses

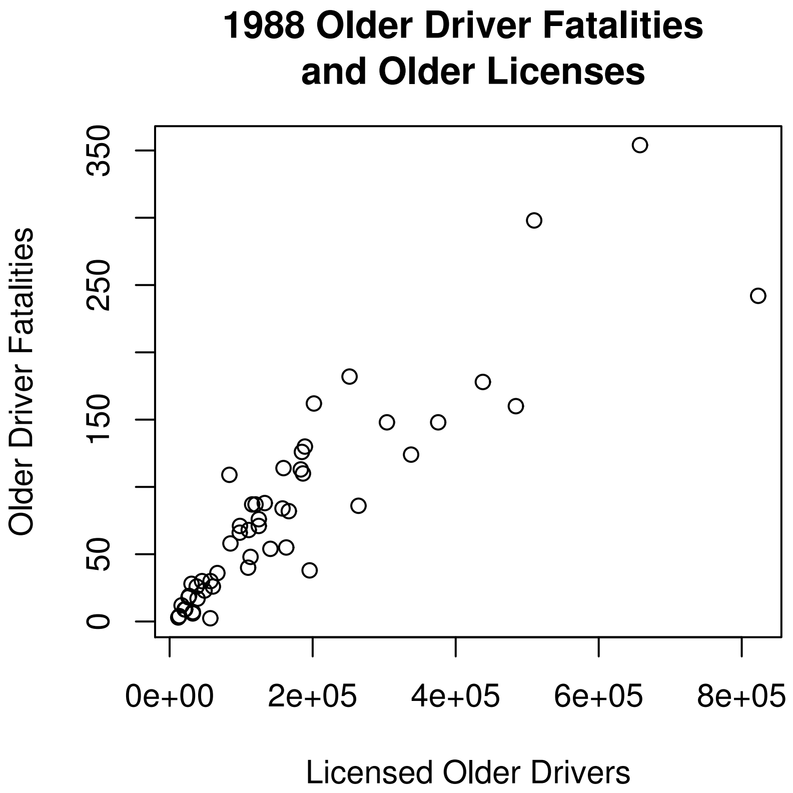
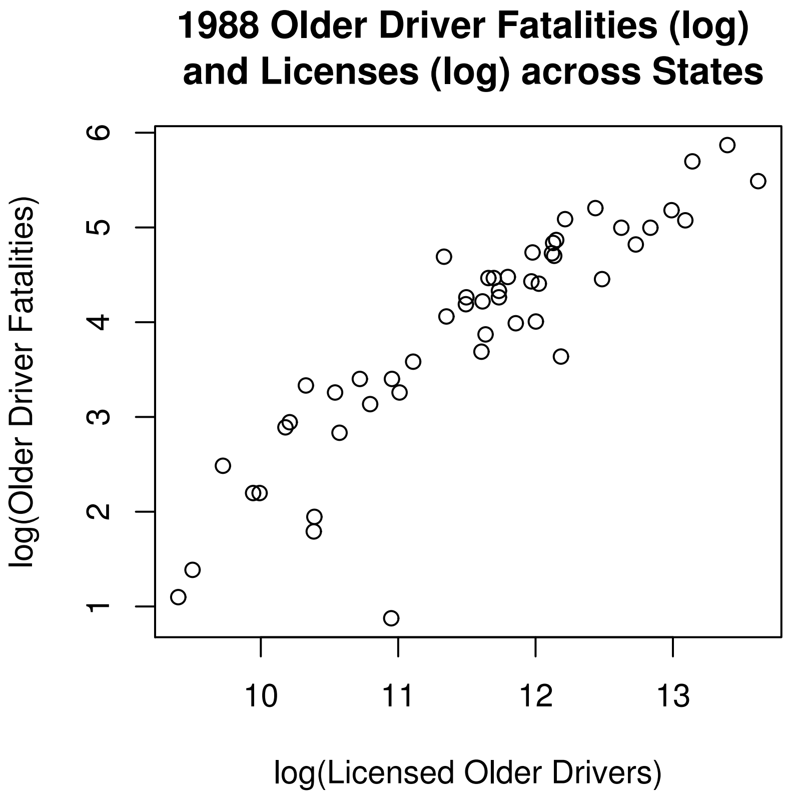


Figure 2: Older Drivers: log-log model



Note that $E(e^{\varepsilon_i}) = 1$. The key parameter is b_1. That is important because it represents elasticity. If $b_1 > 1$, it means that a one percent increase in the number of licenses causes a greater than one percentage point increase in fatalities. And that means that if more older drivers are licensed, then more dangerous drivers must be getting licenses. As a result, the licensing rates might be a good indicator of the policy impact of various proposals to deal with the dangers posed by elderly drivers.

3 Concentrate on the left hand side

There is one problem with the functional form that we can learn something about. It is customary to add a constant to a variable before logging it. That way, in case the observed value were 0, then the log of $(0+\alpha)$could still be calculated and the model would not blow up. But what should αbe?

In Venables & Ripley, MASS 4ed, p. 171-172, they discuss a function (from the MASS package) called "logtrans". logtrans uses maximum likelihood to plot the best choice of a constant alpha in a regression equation like

$$log(fatalities_i + \alpha) = b_0 + b_1 log(older\,licenses_i) + \varepsilon_i$$

Actually, logtrans works to figure the best αwhether you log the right hand side or not. (I made the mistake of not logging it in my first go-round with this exercise and got a weird result–see below).

The graphical output from this R command:

```
vals <- logtrans(fatals ~ log(licenses), data=dat, alpha=seq(1,1000,len=200))
```

is shown in Figure 3:

Note that the "best" alpha is clearly a really small number, certainly not more than 50. The point estimate of the best alpha is 21.08.

If we make the dependent variable the log(fatalities+21.0), the scatterplot looks just about right. Please consider Figure 4. In my opinion, this scatterplot is slightly more linear in appearance than Figure 2, but either one is nice looking, compared to the one we started with.

4 Box-Cox

If you stop to think about it, the log(y) is a pretty arbitrary tranformation. There is a whole range of similar functions that might do as well.

In 1964, Box and Cox considered the log as one possible transformation in a framework like this:

$$y^{(\lambda)} = \begin{array}{ll} \frac{y^{\lambda}-1}{\lambda} & if\,\lambda \neq 0 \\ log(y) & if\,\lambda = 0 \end{array}$$

In this framework, the parameter λranges across a scale. If λ happens to be 0, then the output is the log of y. But for other λ,then different functions emerge. For example, if it turns out that $\lambda = 1$,then you have a linear transformation. If you set $\lambda = 0.5$, (recall that $y^{0.5}$is just $\sqrt{y}$), you have a square root representation that looks similar to a log, but it is not exactly the same. On the other hand, if you put $\lambda = 2$, then you are essentially squaring y. If you set $\lambda = -1$, then the relationship is a reciprocal.

You might wonder why they do it this way. There's a mathematical argument which shows that as λgets closer and closer to 0, the formula

$$(y^{\lambda} - 1)/\lambda$$

converges to $log(y)$. I know it "seems" as though this formula would become undefined because 0 would be in the denominator, but it just isn't so! (See Kmenta, Elements of Econometrics 2ed, p. 518).

Figure 3: Logtrans output for (log(fatalities+α)$\sim$log(licenses)

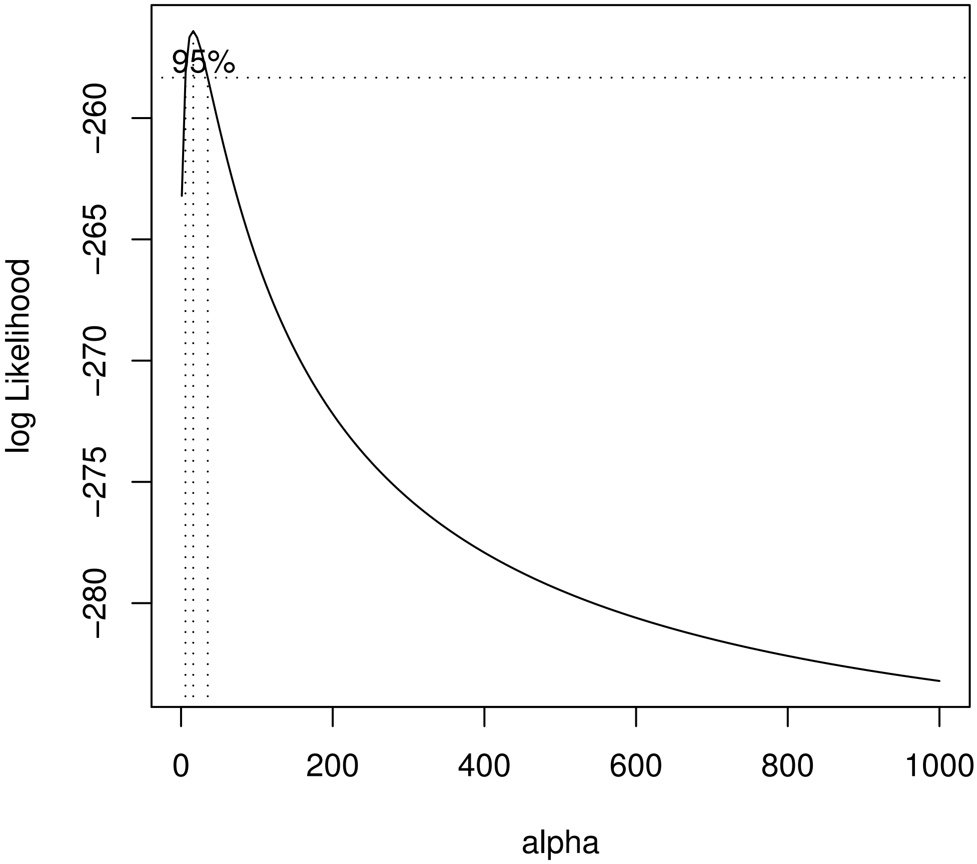


Figure 4: Plot of log(fatalities+21) against log(licenses)

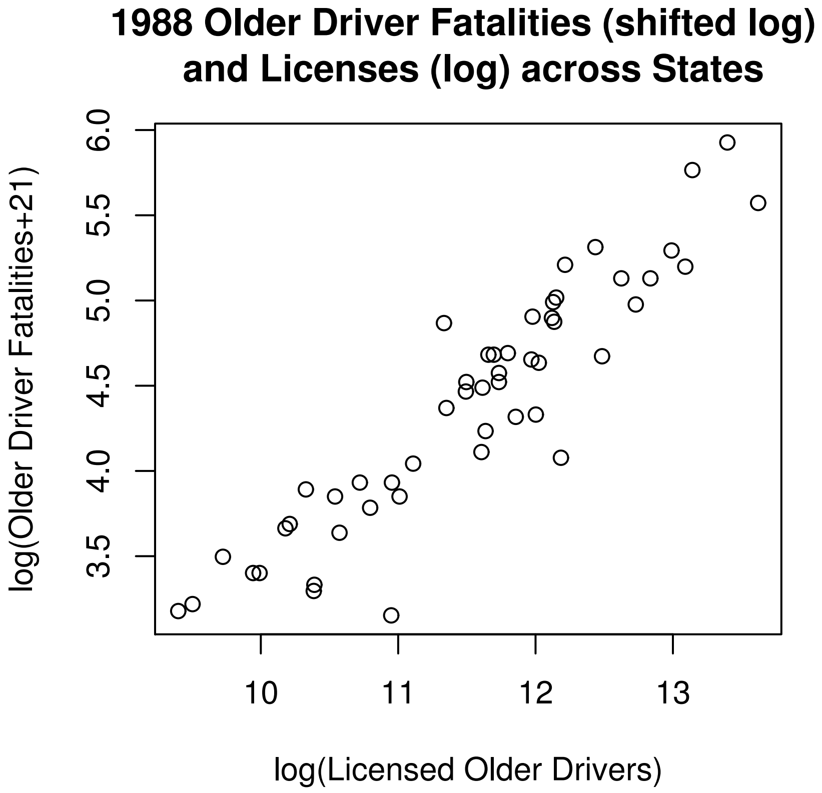
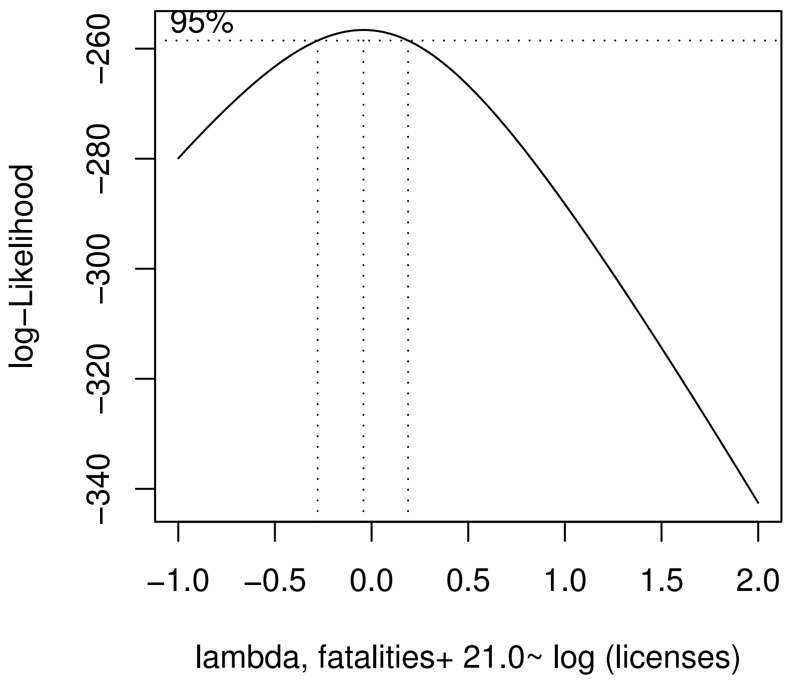


Figure 5: boxcox(fatalities~log(licenses+21))



In the original Box-Cox approach, as far as I understand it, one was supposed to apply this λ- transformation to all the variables in the model. So you would think about a theoretical model like:

$$\frac{y_i^\lambda - 1}{\lambda} = b_0 + b_1 \frac{x_i^\lambda - 1}{\lambda} + e_i \tag{2}$$

I do not know the detailed history of this approach, but I gather that, pretty early on, people started to observe that there is no reason to apply the same transformation across the whole model. One might, for example, want something like this:

$$\frac{y_i^\lambda - 1}{\lambda} = b_0 + b_1 x_i + e_i$$

or

$$\frac{y_i^\lambda - 1}{\lambda} = b_0 + b_1 log(x_i) + e_i \tag{3}$$

In Venables and Ripley, 4ed, p. 171, the approach being described is that LHS sort. The boxcox method in the MASS package lets you estimate the best lambda value for a given dependent variable against a set of independent variables that the user can transform in any desired way.

Suppose we want to reconsider the idea that the dependent variable should be log(fatalities+21.0). In Figure 5, the result of the boxcox command is presented:

```
boxcox(fatals+21.0 ~ log(licenses), data=dat, plotit-T, xlab=”lambda, fatalities + 21.0~ log(licenses)”)
```

That result indicates that the appropriate λvalue is 0, or very close to it. Hence there’s no reason to reject the idea that the dependent variable should be log(y+21.0).

5 Estimated model

I don't feel entirely comfortable just throwing together variables in this way. Usually, if there is some theory about how variables are related, then one can feel a bit more enthusiastic about specification.

Given the information at hand–which is all I have–I believe the conclusion should be that the best specification is:

$$log(fatalities_i + 21.0) = b_0 + b_1 log(olderlicenses_i) + \varepsilon_i$$

and so we estimate this model in R with the command:

```
lm(log(fatals+21)~log(licenses),data=dat)
```

and the parameter estimates are:

Coefficients	OLS estimate	Std.Error		
Intercept	-2.97*	0.43		
log(older licenses)	0.68*	0.037		
$R^2 = 0.85$	adj R^2=0.85	N=51	Residual Std.Error=0.27	
* $p \leq 0.05$				

Well, note the residual standard error got smaller, the $R^2$is pretty good. We could be really happy except that the estimate of $b_1 = 0.68$, which does not help very much with the elasticity argument that I wanted to make when this all started. The estimated model implies the "truth" is something like:

$$\widehat{fatalities}_i = -21 + e^{-2.97} olderlicenses_i^{0.68}$$

All of this left hand side analysis is carried out on the assumption that the right hand side is correctly specified. As you will see in the next section, the conclusions you make about the parameters α and $\lambda$are vitally dependent on what is put in on the RHS of the equation. If you look closely at the likelihood values displayed Box-Cox plots above and below, I believe you will conclude that, between $olderlicenses_i$and $log(olderlicenses_i)$, the best fitting choice is indeed $log(olderlicenses_i)$. But that does not mean that some other transformation might not be better.

6 Watch out for this pitfall

As a concluding note, I want to confess that I made a couple of mistakes in my first cut at this and I expect other people will make them as well. The mistakes I made were all due to careless specification on the right hand side of the equation in the commands logtrans and boxcox.

If you forget to log the right hand side in the logtrans calculation, using the command:

```
vals <- logtrans(fatals ~ log(licenses), data=dat, alpha=seq(1,1000,len=200))
```

then the estimate of alpha that is 191.0. If you use that alpha as your working transformation for $fatalities_i$, then you produce a scatterplot for the log relationship which is clearly nonlinear (and mistaken). Consider Figure 6:

I made the same right-hand-side mistake when it came to the Box-Cox transformation. I had in mind the sort of model described in the Kmenta book (see equation 2). I mistakenly assumed it would estimate the λcoefficient on both sides when the following command was used:

```
boxcox((fatals) ~ licenses, data=dat, plotit=T)
```

But that does not estimate both sides, of course, it only estimates the LHS transformation. The boxcox procedure proceeds on the assumption we want the right hand side to include licenses, not log(licenses), and the mistaken estimate is $\lambda = 0.5$.

That's giving a best estimate of λthat is pretty far from zero. It could "fool" us to proceeding on the hypothesis that the dependent variable should be $\sqrt{fatalities}$.

If you run it again with the transformed value of $y| + \alpha$ with this command:

```
boxcox((fatals+21) ~ licenses, data=dat, plotit=T,xlab="lambda when alpha=21.0")
```

you get a similar result.

If you follow along with the idea that the λ coefficient should be 0.5, then you will arrive at some really peculiar regression results and a bad looking scatterplot.

Figure 6: Mistaken logtrans fitting log(fatalities + α)$\sim$licenses

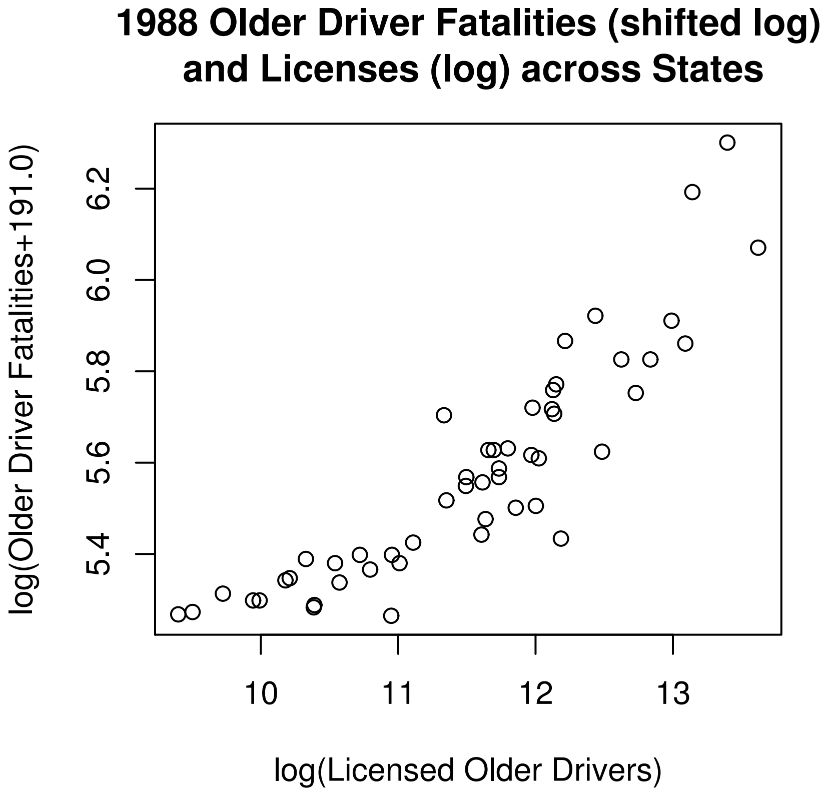


Figure 7: boxcox(fatalities∼licenses)

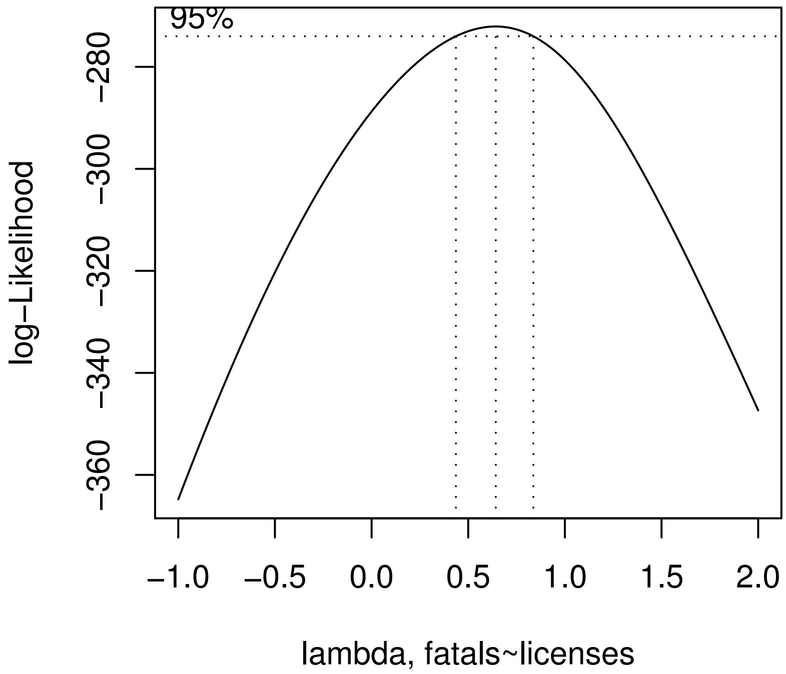
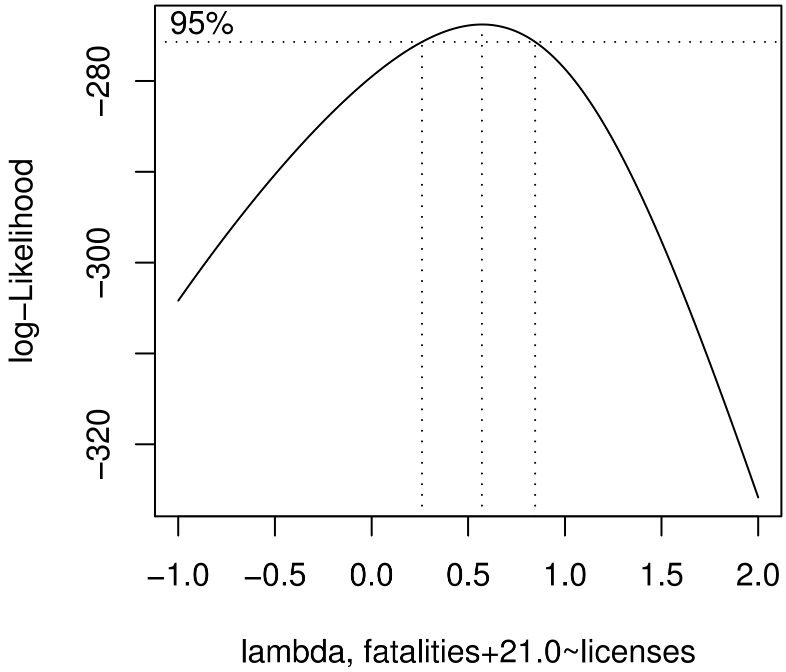


Figure 8: boxcox(fatalities+21∼licenses)


log−Likelihood
lambda, fatalities+21.0~licenses