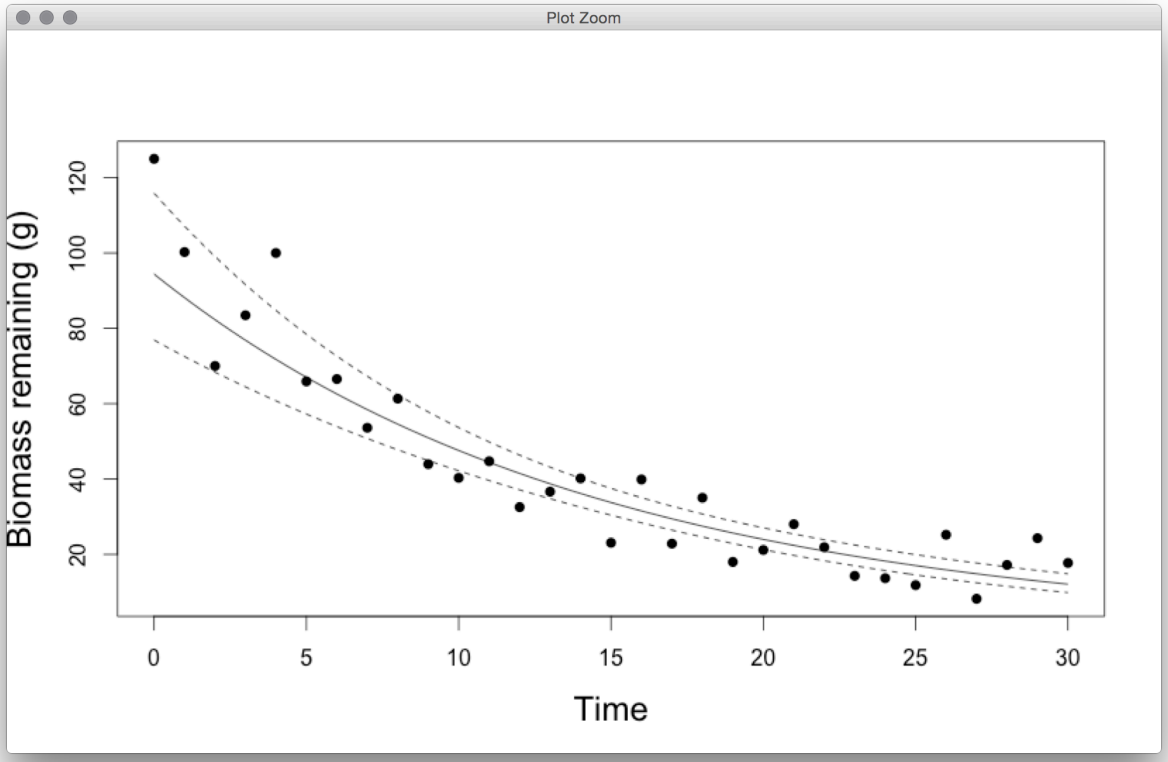
Advancing in R



Module 9: Nonlinear models

Introduction

In this practical, we will exploit R's ability to estimate coefficients for any number of nonlinear equations. We will start with the simplest approach, involving transformations, and proceed to more flexible and potentially more realistic methods.

This practical assumes that you are familiar with the general syntax of coding in R, with making basic plots, with the theory and practice of linear and multiple regression, with the general version of the linear model, with nonlinear functions and nonlinear least squares, and that you have reviewed material covered in the first six practical exercises. If you need to, review this material before attending the practical session.

Learning outcomes

Know how to recognize nonlinear relationships in both plots of data and diagnostics

Know how to fit nonlinear relationships using transformations, polynomials, and mechanistic models

Know how to generate orthogonal polynomial terms

Know how to use self-starting functions to fit mechanistic nonlinear models

Know how to specify starting parameter values for mechanistic nonlinear models

Some useful commands

Below is a list of some (but not all) of the useful commands you may need to run at some stage of today's practical.

exp()

I() ←(NB: capital "i")

log()

nls()

nlsResiduals()

poly()

predict()

sqrt()

Data quality control and exploration

Begin by loading the RStudio package. Open a new project and a new Rscript, and save each of these using meaningful and descriptive filenames. Include the usual annotation at the start of your script that identifies the purpose of the script and its author, as well as commands that clear the workspace.

Find the data file called “decay.csv” in the folder for this exercise, and read it into a well-named data frame. Verify that it has loaded properly, and examine the structure of the object using the str() command.

This dataset is from a study of the mass of organic material remaining (MASSLEFT) as a function of TIME, i.e., it is a study of the rate of organic decay.

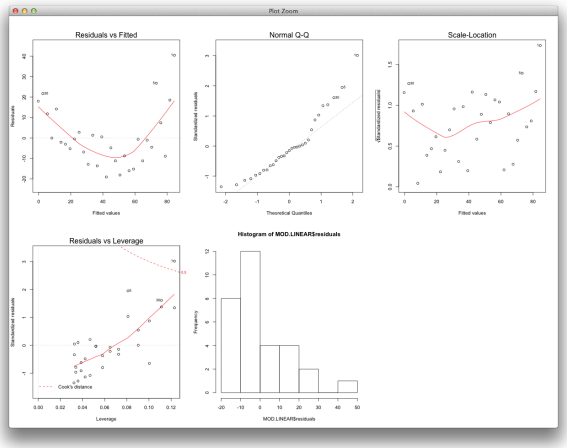
Use plotting commands to exercise data quality control. Check for outliers. If necessary, clean the data and recode any variables that need to be read differently than they currently are.

Examine the distributions of the variables. You may notice that the response variable is markedly skewed. Before we explore transformations, let’s plot the raw data – the nature of the relationship may affect our decisions concerning transformations of the response. What kind of plot should you use to explore the covariance between TIME and MASSLEFT?

Having plotted the data, what do you think about the likely relationship between the two variables? Is there any theoretical reason to expect that organic decay should proceed in a nonlinear manner? If so, what kind of function might best summarize the relationship?

Fitting a linear model: how diagnostics can spot curves even when you can’t

You will no doubt be strongly suspicious of a simple linear model based on what you have seen so far, but it’s instructive to proceed anyway, because sometimes the curve will not be so obvious, and it helps if you have experience with diagnostics from poorly fitting models. Build a linear model using the untransformed data, and examine the diagnostic plots. What is remarkable about the plot of residuals versus fitted values?



To help you interpret this, plot the data on a raw scale, and add the fitted linear function. Can you now understand what the "u-shape" of the first diagnostic plot tells you about the model fit?

Nonlinear functions part 1: transforming data to produce linear relationships

Now that it's clear that a straight line through the raw data won't work, try some of the common transformations (e.g., exponential, log, root or power transformations). You should examine two sets of plots: histograms that might provide hints on error distributions, and scatterplots that show the relationships between TIME and the transformed responses. Which of the transforms is best? Does it accord well with your theoretical reasoning about the biological relationship, above?

Build a linear model using the transformed response. You should nest the transformation within the model rather than creating a new vector, because sometimes it's handy to have R remember where the original data came from. Examine the diagnostic plots. Are they better? Remember that the ultimate goal is a good model fit that summarizes the relationship, not necessarily impeccable diagnostic plots….

Assuming you are satisfied with the diagnostics, examine the model summary. How many parameters are there? Which of these represents the test of the relationship? How would you interpret the coefficient if you had to explain the curvilinear relationship?

Illustrate your model by making two separate publication quality plots. In the first, plot TIME against the transformed response, and include the fitted line and confidence region (using the predict() and matlines() commands). This kind of presentation most accurately represents the model, but it has a serious drawback in terms of interpretability. Therefore, also plot the data on a raw scale. How could you now illustrate the line of best fit? Hint: you must back-transform the predicted values (for both the fit and upper and lower bounds of the CI) before plotting them. Which transformation is the inverse of the log transform?

Write a single sentence of results text that summarizes the relationship, citing the key statistical terms and the figure as appropriate.

Modelling growth in deer using polynomial regression

Often there are no transformations that will linearize a relationship; in these cases we need different methods to produce curves in our models. The first of two approaches I will introduce involves adding higher order coefficients for the predictor variable. Because the polynomial terms can be interpreted in the same way as any predictor (i.e., the method used to fit coefficients is the same), all of the linear regression methods we have learned so far can be deployed for polynomial regression. This will not be true when we explore nonlinear least squares later on.

Find the data file called "jaws.csv" in the folder for this practical, and read it into a well-named data frame. Verify that it has loaded properly, and examine the structure of the object using the str() command.

This dataset is from a study of jaw length (BONE) as a function of AGE in a population of deer, i.e., it is a study of growth. Use plotting commands to exercise data quality control. Check for outliers. If

necessary, clean the data and recode any variables that need to be read differently than they currently are.

Examine the distributions of the variables. Once again the response variable is skewed, but this time the skew is negative (with a longer tail to the left). Are any of the usual transformations useful for this kind of skew? Try them and see. Sometimes negative skew is improved by power transformations (by raising the data to a second or higher power). Does that help in this case?

Plot the relationship between jaw length and AGE. How would you describe it? What biological intuition is represented by this curve?

Although we can be fairly sure that it will be inadequate, build a linear model using the raw data anyway, and examine the diagnostic plots. You should see that curves produce rather predictable deviations from the expected lack of pattern, especially for the plot of fitted versus residual values.

Let's build a second-order polynomial regression. We could do this manually by creating a vector of squared predictor terms, but then we would run a very high risk of multicollinearity (obviously a predictor will covary strongly with its square). Conveniently, R has an inbuilt function that produces *orthogonal* polynomials, meaning that the squared term is built in such a way as to be uncorrelated with the original term (by mean-centering the quadratic term). The syntax for a second order polynomial regression is as follows:

```
> MOD.POLY.D<-lm(BONE~poly(AGE,2),data=DEER)
```

Note that it would be relatively simple to change the order of the polynomial, by changing the 2 to 3 or higher if you wanted to. Let's stick with a second order for now. Examine the diagnostic plots for this model. Are they better than those for the linear regression? Check out the model summary. Can you interpret all three coefficients? What does the sign of the second-order term imply about the convexity or concavity of the curve? Use an F-test to verify that the polynomial regression is superior to the linear form.

Now plot the data, the fitted line, and the confidence interval around it using the predict command. You will need to specify a new vector for only the single predictor variable provided you have nested the poly function within the model call. Examine the fitted line, and ask yourself if the function is biologically justifiable – is it really probably that jaw length decreases in very old deer, the model fit notwithstanding?

Nonlinear least squares regression

You will probably conclude that the curve from the polynomial regression is a good fit for the data, but does not well represent the biological reality of growth. Consequently, we might like to use a different function that better approximates how growth is known to work: the asymptotic exponential function:

$$y = a - b * e^{-c*x}$$

where x is the predictor, y is the response, and a,b, and c are parameters (analagous to the slope, m, and intercept, c, in y = m*x + c).

Because nonlinear least squares (NLS) regression finds coefficients using iterations rather than analytically, we need to provide starting values for the coefficients. This can be challenging unless you understand the particulars of the equation you are using. In our case, a represents the asymptotic value of the function, which we can guess is around 120 by consulting a scatterplot. We can deduce a starting value for b by setting age to zero, in which case the value of b is derived by rearranging the equation b = a – intercept. If the intercept is around 10, then we can guess that b is around 110. To guess parameter c, we must inspect the plot at the steepest part of the curve (in these data, when AGE is around 5, and bone is about 40). We can guess the value for c by rearranging the equation again given the values of a and b specified: c = -log ((a – y)/b)/x. When AGE is 5, BONE is 40 or so, and therefore

c = -log ((120 – 40)/110)/5 = 0.063. A bit painful, but we now have three starting values. Use them to build a NLS model.

```
MOD.ASYMP<-nls(BONE~a-b*exp(-c*AGE),
data=DEER,start=list(a=120,b=110,c=0.063))
```

Sometimes your starting values don't provide a solution because the iterative procedure used by NLS cannot produce stable parameter estimates. In those cases, re-examine your choices, and try different values. For a few specific nonlinear functions, R also has inbuilt functions that will produce a regression without starting values. These so-called "self-starting" models simplify the procedure of doing nonlinear least squares, but you should always check the sensitivity of your estimates to a range of starting values, even if you use the self starting functions to find your first guesses. The syntax for using the self-starting function is below, with the function SSasymp the key ingredient that specifies both the form of the equation and the starting values for a, b, and c:

```
MOD.ASYMP.SS<-nls(BONE~SSasymp(AGE,a,b,c),data=DEER)
```

To examine diagnostic plots for NLS regressions, we'll need to (install if necessary and) load a special package, nlsTools. Do so now. This package contains a function, nlsResiduals, for working with the residuals of the NLS regression, and you can nest that function into a plot() command to view a selection of four diagnostic plots.

```
plot(nlsResiduals(MOD.ASYMP))
```

This is a slightly different selection of plots from those we have been using, but they illustrate the same fundamental properties of the regression, and if you're on the lookout for the same patterns you'll be able to spot an ill-fitting model.

Compare the coefficients of the two models (the one generated with starting values, and the one that self-started. Do they concur? Disagreement may reflect sensitivity to starting values, or it may reflect superfluous coefficients. Can you work out which of the three coefficients is superfluous? What is your reasoning?

Try building a two-parameter asymptotic function by omitting the questionable coefficient. You may need to alter the formula slightly to get things to work out:

```
MOD.ASYMP.2<-nls(BONE~a*(1-exp(-c*AGE)),
data=DEER,start=list(a=120,c=0.064))
```

Examine the diagnostics and the model summary. Do the coefficients accord well with your predictions from the plotted data? Create a publication-quality plot that illustrates your 2-parameter NLS regression, and use the lines() command to add the fitted line to the plot (fitting confidence intervals to the line is not straightforward because the coefficients are not independent: there are ways to approximate confidence limits around these curves, but these involve either making questionable assumptions or using resampling methods that are beyond the scope of our course). Finally, write one or two sentences of results text that summarize your findings and refer to this figure.

Remember to save your project and R script, and if you want, your figure as well.

~ End of Practical ~