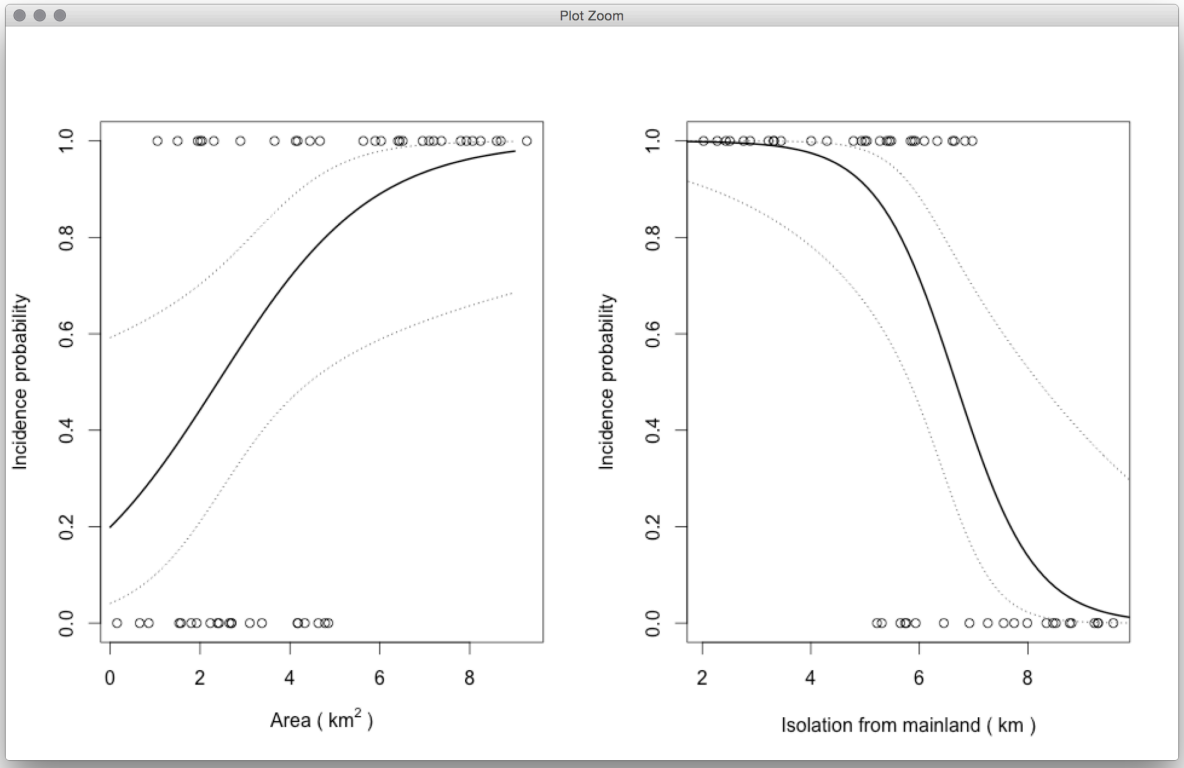
Advancing in R



Module 6:
Generalized linear models

Introduction

In this practical, we will generalize the linear model for response variables that have non-normal error distributions as well as nonlinear responses. We will begin by focussing on a dataset with a count response variable, and then build a binomial response model.

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, and that you have reviewed material covered in the preceding practical exercises. If you need to, review this material before attending the practical session.

Learning outcomes

Know how to recognize count and binomial response data

Know how to specify error structure and build generalized linear models

Know how to generate and interpret diagnostic plots for generalized linear models, and to identify high influence and high leverage records

Know how to request chi-squared likelihood ratio tests when comparing different generalized models

Know how to request back-transformed fitted values of a generalized model

Know how to encode two-vector binomial responses (supplementary exercises)

Know how to fit models that account for over- or under-dispersion (supplementary exercises)

Know how to use the inverse of a link function to back-transform predictions and confidence intervals (supplementary exercises)

Libraries

In addition to the basic libraries automatically loaded with R, you may wish to use the following additional packages. If necessary, make sure you either have internet access on your machine or that you install these packages prior to the practical.

{boot}
{ggplot2}

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.

anova()
avPlots()
ggplot()
glm()
glm.diag.plots()
jitter()

levels()
lines()
predict()
points()
rep()
seq()
split()
subset()
summary()
update()

Data quality control and exploration

Begin by loading the RStudio software. 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 "species.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 **str**ucture of the object. These data include measures of species RICHNESS in plots that have different measures of BIOMASS (a continuous predictor) and three levels of a categorical predictor, pH (PH). 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. What are the properties of the response variable RICHNESS? Is it a continuous variable, or does it only take integer values? Can you relate this property to the nature of the response? Is there any reason to suspect that the variable has a lower bound? What would such a bound imply about the error distribution for this variable? What is the distribution of the predictors? How are replicates balanced across levels of the categorical predictor? Are the predictor variables sufficiently correlated to cause concerns about collinearity?

Plot the data so that you can simultaneously visualize the effects of both predictors on the response in a single panel. Do you think there's an interaction between the two predictors? What would that look like? How many coefficients should there be in a model without an interaction? How many extra coefficients do you expect to model the interaction? What sign do you expect for each of the coefficients in the maximal model? You will notice that a linear fit would probably suit the data fairly well. What are two reasons that a linear model is inappropriate for these data?

To help you anticipate the coefficients for your generalized model, replot the figure above, but this time, log transform the response variable (i.e., make your plot in loglinear space). Although it is a crude approximation of what the generalized model is trying to do, this plot is nevertheless ideal for allowing you to predict the coefficients of your generalized model. Does this plot suggest an interaction that was less evident in the original plot above?

Generalized linear models for count response data

Counts of objects are one of the most frequent kinds of response variables, and modelling them requires considering both the distribution of errors and the nonlinearity of the prediction. We can use the function glm() to call a generalized linear model for the data. (Remember that this is NOT the same thing as a general linear model, which can be coded in R using the overloaded lm()

function). When we specify family=poisson in the function, R will use the "canonical" (i.e., the typical) link for Poisson data, which is the log-link. This means that R will assume errors have a Poisson distribution and that the predicted values will be log-linear.

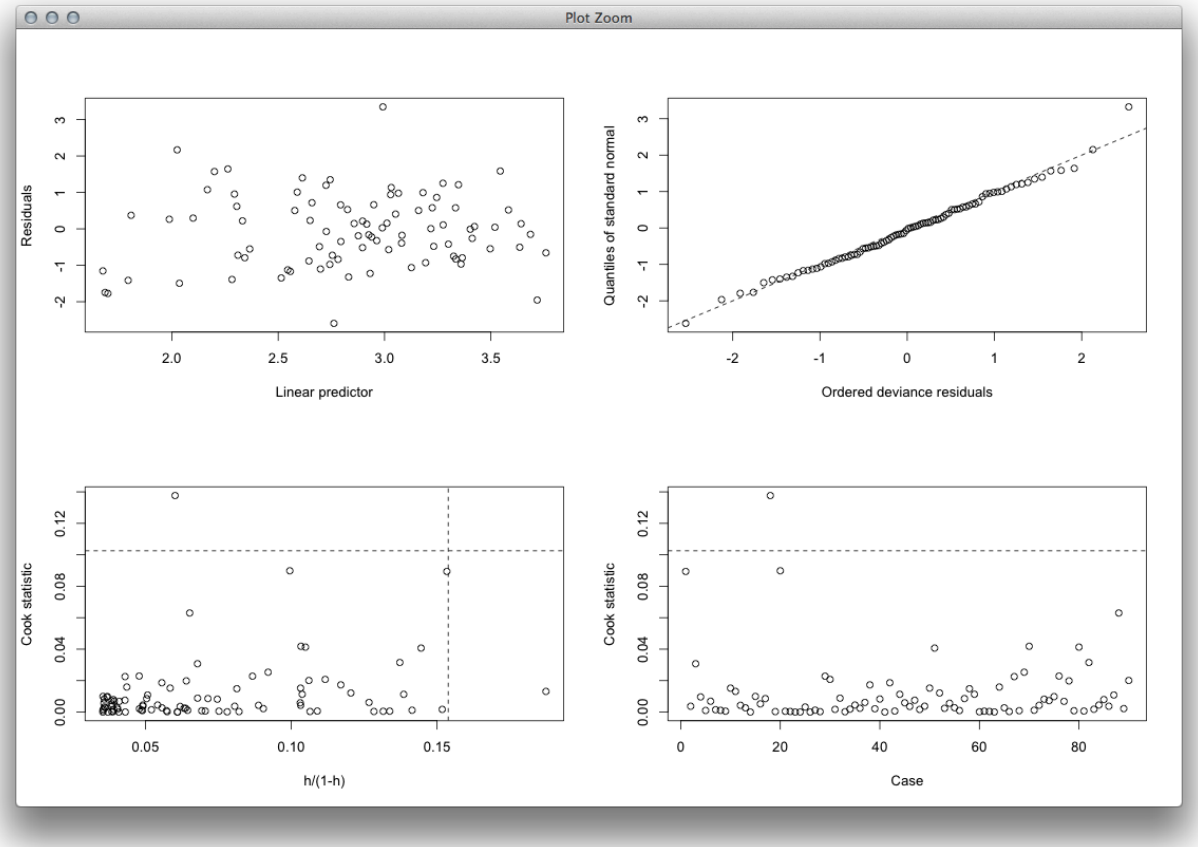
```
MOD.1<-glm(RICHNESS~BIOMASS*PH,data=SPECIES,family=poisson)
```

The regular diagnostic plots can be helpful for diagnosing generalized linear models, but I find it useful to look at a specially designed quartet of plots to diagnose GLMs. To generate these, we need to load the {boot} library (if you haven't already installed {boot}, you will need to do so before loading the package).

```
library(boot)
```

```
glm.diag.plots(MOD.1)
```

Study the plots, and work out what is expected for each of them if the model has a good fit. The help file for this function has details on how to interact with the graphical result of this function to identify high influence or high leverage points in the two plots involving Cook's distances (hint: you will need to enter information in the console, even though I normally say you shouldn't do this, and click on suspicious points.) These arguments may be useful if you want to exclude data that you think might have to strong an effect on your estimates of coefficients.



Having examined the diagnostics, examine the model summary. First, examine the magnitude of the residual error, and compare this to the degrees of freedom. You will find this information below the table of coefficients, and it should look something like this:

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 452.346  on 89  degrees of freedom
Residual deviance:  83.201  on 84  degrees of freedom
AIC: 514.39

Number of Fisher Scoring iterations: 4
```

The ratio of the residual deviance to residual degrees of freedom should be roughly 1, as indicated by the statement just below the table of coefficients, that the dispersion parameter is taken to be 1. (Although it is beyond the scope of our practical exercise, the reasons for this involve the simplifying assumptions necessary to reduce the curved line you are trying to fit into a set of coefficients that look similar to a linear model.) If this ratio is <0.5 or greater than 2, we need to account for under- or overdispersion, respectively (in other words, the simplifying assumptions I mentioned above have been violated). The easiest way to do this (although it tends to be hyperconservative) is to use a quasipoisson rather than poisson family designation (for binomial models, you can use a quasibinomial family designation instead of binomial). More complex solutions involve fitting extra parameters, and generally are less conservative but require a bit more reading and fiddling. Is there any need to worry about these problems for your model?

Assess your guesses on the number of coefficients and their signs. Make sure you understand any discrepancies before moving on. Do the values of the coefficients make sense, especially when compared to your plot of the data in loglinear space? How would you transform these estimates to get back to the raw scale in which the response was first estimated? Do you think there is strong evidence from this summary for an interaction?

To begin model simplification, build an alternate model without an interaction between the predictors. Use the anova() command to compare models, but make sure you add an argument that tells R to use a chi-squared test instead of an F-test, or you won't get a p-value:

```
anova(MOD.1,MOD.2,test="Chi")
```

On the basis of this test, should you favour the more parsimonious or the more complex model? Continue simplifying until you obtain your minimal adequate model.

To help interpret the coefficients, make a publication quality plot that illustrates the fitted effects of your model. Because you are trying to illustrate an interaction, the easiest way to do this is to generate a separate predict() statement for each of the three levels of your categorical predictor, and then add the line of best fit for each of these levels in turn.

Recall that the predict() function needs at least two pieces of information: the model for which you want to generated fitted (predicted) values, and new data vectors that you will feed into the model (this must include a separate vector for each of the predictors in your model, and all of these

vectors must be equal in length (i.e., have equal numbers of rows). For example, if your model contains two predictor variables, BIOMASS and PH, you must provide the predict command with

1) the name of the model for which you will predict the effect;

2) a "new" vector of BIOMASS entries that would be useful for illustrating its effect on the response variable (e.g., that runs from the minimum value of BIOMASS to the maximum in small intervals, which you can create using the seq() command:

```
NEWBIOMASS<-seq(0,10,length=100)
```

3) a vector of PH levels that is equal in length (i.e., number of rows) to the NEWBIOMASS variable. Since we want to draw a separate line for each level of PH, we'll want to create three of these, one for each level. For example:

```
HIGHPH<-rep("high",100)
```

In addition to these required parts of the predict command, you can use the argument type="response" in the predict() call, which will force the predicted values to be stored in raw units rather than on the log scale. This saves you from having to backtransform later, a trick that is useful when the link function is more complex than a log-transform:

```
PREDHIGHRICHNESS<-predict(MOD.1,list(PH=factor(HIGHPH),
BIOMASS=NEWBIOMASS), type="response")
```

Note that I have nested a factor() command in the above call to make sure that predict interprets the list of repeated "high" entries as a factor. Until recently, the drawback in this approach was that you couldn't accurately get confidence intervals around the lines (now the authors of the package have added the capacity to estimate the se.fit for your prediction as well – feel free to experiment with using this argument to add confidence intervals around your prediction). As an alternative, and because it's a useful way to test your knowledge about generalized models, I like generating the confidence limits around lines of best fit back back-transforming predictions from loglinear space (to get predictions in loglinear space, either omit the "type=" argument, or change it to type="link"), and include an argument asking for the standard error around fitted values (se=TRUE). You must then exponentiate (using the exp() command) the fitted values as well as the 95% confidence limits in order to plot the "asymmetrical confidence" in linear space around the line. Why does this confidence look asymmetrical around the best fit line? Why do we exponentiate the fitted values?

Having plotted the curves, consult the table of coefficients once again. Can you make sense of all the estimates, including those for the interaction terms? Write a sentence or two of results text that summarizes your findings and parenthetically cites the statistics and figure.

Binomial logistic models

Clear your workspace and load the datafile "isolation.csv" into a well-named dataframe. This dataset features information on the number of bird species found on islands as a function of their areas and distances to the mainland. It includes a binomial response variable, INCIDENCE: in this

case it is scored as a 1 if a bird species was found on an island, and 0 if the bird did not inhabit the island. The two predictor variables are island AREA (in km^2) and ISOLATION (distance to mainland in km). Make sure that your dataframe doesn't share a name with one of the vectors within it!

Examine the structure of your dataframe, and check for outliers. Note that incidence is coded numerically even though it can only take two values – this is good! You don't need (or want) to recode incidence as a factor, because R is smart enough to treat it appropriately when you specify a binary logistic model. Are the predictor variables sufficiently correlated to cause concerns about collinearity?

Plot the response as a function of the two variables. Because the two predictors are continuous, it's hard to envision their combined effect, so make two plots with each variable as the lone predictor. Can you guess the sign of the coefficients? If you want to make sure you can see all of the points in your plot, you may wish to use the jitter() command to add small random deviations to individual records.

```
plot(jitter(ISLAND$INCIDENCE,factor=0.25)~ISLAND$AREA)
```

Build a generalized linear model that predicts the partial effects of AREA and ISOLATION on INCIDENCE, and includes an interaction. Use a similar syntax to the one you used for the Poisson glm above, but this time specify family=binomial. Quickly check for under- or over-dispersion by comparing the residual deviance of the model to the residual degrees of freedom. Are any changes to the error structure necessary?

Examine the diagnostic plots using the purpose-built function from the {boot} library. You should note any potential problems at this stage, but conduct model simplification before you take hard steps to address any concerns. Remember to use the argument test="Chi" in your anova() function when comparing models. Having obtained a minimal adequate model, re-examine diagnostic plots, and attend to any concerns. If there are high influence or high leverage records, check whether excluding these data makes a difference for the coefficients and null hypothesis tests.

Build publication quality plots that illustrate the effects of the predictors on the response variable. Use your judgment about whether and how to illustrate partial effects and curves so that your reader can best appreciate the nature of the predicted relationships. In the exercise above, you predicted the effect of BIOMASS at three different PH levels. In the current exercise, your two predictors are both continuous. How can you illustrate the effect of one variable while holding the other constant? Write several sentences that summarize your findings and refer to the statistics and figure(s) as appropriate. What concept does the fitted line in your plots represent (i.e., does this represent A) an estimate of the predicted incidence, or B) something else? Hint: It's not A.)

In the supplementary exercises we'll consider how to analyse another format for data featuring binomial responses. In the meantime, remember to save your project and R script, and if you want, your figures as well.

~ End of Practical ~