Day 1: Self-guided practical - Using R for linear regression and GLMs

Andrew Parnell

Introduction

Welcome to the first user-guided practical on linear regression and GLMs. In this practical you will:

- Fit a basic linear regression model
- Check the model fit and the residuals
- Fit a Binomial GLM model
- Fit a Poisson GLM and compare it to a negative binomial one

There are four sections. You should work your way through the questions and put your hand up if you get stuck. The answer script is also available in the practicals folder but I urge you not to look at it unless as a last resort!

You can run the code from these practicals is by loading up the `.Rmd` (Rmarkdown) file in the same directory in Rstudio. Feel free to add in your own answers, or edit the text to give yourself extra notes. You can also run the code directly by highlighting the relevant code and clicking `Run`. Much of this material overlaps with the class slides so sometimes if you get stuck you might get a clue by looking at the `.Rmd` file in the `slides` folder.

One final small note: if you are copying R commands from the pdf or html files into your R script or console window sometimes the inverted commas can copy across incorrectly. If you get a weird message saying `Error: unexpected input` you usually just need to delete/replace the inverted commas.

Task set 1: linear regression

The `airquality` data set is included with R. You can find a description of the data at `help(airquality)` and can get at it simply by typing `airquality` at the R command prompt. Let's suppose we're interested in predicting the variable `Ozone` from the other variables

Tasks:

1. First create some plots of the variables and make sure you understand the relationship between them. Hint: A good start can be found in the `help(airquality)` file (see the example at the bottom of that page), though that plot isn't very helpful for visualisation of the month and day variables, nor for visualising the shape of individual variables (e.g. using `hist`)

2. Fit a linear regression model using `lm` with `Ozone` as the response and all the other variables as covariates. Use the `summary` method to interpret your findings. (Note that R here is automatically removing rows with missing variables)

3. Have a look at the residuals of the model (e.g. histograms and QQ-plots). Does the model fit well?

4. Try another model but this time using the log of Ozone instead. Does it fit better?

5. Identify the one strange observation and see if you can work out what happened that day

6. You can get the AIC of this model with e.g. `AIC(my_model)`. Try some more models and see if you can get a better fit. Some ideas might include: interactions between terms (e.g. include `+ Wind*Temp`),

quadratic functions (e.g. include `+ I(Wind^2)`), and changing month and day to be factor rather than numerical variables.

Task set 2: logistic regression

1. Load in the `horseshoe.csv` data set from the data directory and familiarise yourself with the data structure from the `data_description.txt` file
2. Turn the `color` and `spine` variables into factors with their proper names
3. Familiarise yourself by plotting the data and exploring the structure between the variables
4. Create a binary variable which contains only whether the satellite variable is >0 or not. We will use this as our response variable. Create a plot which shows the relationship of this variable with `width`.
5. Fit a binomial glm (a logistic regression) with the binary variable as the response and `width` as a covariate. Summarise your findings
6. Create a plot of the fitted values on top of a scatter plot of the data (hint: width on x-axis, binary response variable on y-axis)
7. Try fitting some more models to the data with more variables (and perhaps interactions) to see if you can get the AIC lower. Compare your new models' fitted values to the first model

Task set 3: Poisson and Negative Binomial regression

1. This time fit a Poisson GLM to the horseshoe data, using the original number of satellites rather than the binary version you fitted previously. Again use `width` as the sole covariate, and again plot the fitted values
2. Now try a model with all of the covariates (make sure not to include the binary variable you created before). Summarise and see if there's any improvement. You might notice that more variables are important (compared to the previous binary logistic regression) because we're using more of the data
3. A common occurrence is that the Poisson distribution is a poor fit to the data as the mean=variance relationship is rarely met. (You could check if the mean and the variance match for these data). A common alternative is to fit a Negative-Binomial GLM which has an extra parameter to measure excess variance (over-dispersion). The `glm` function doesn't have this distribution in it by default so you need to call in the MASS library with `library(MASS)`. The family will now be called `negative.binomial` and you'll need to specify a value of the excess variance `theta`. The ratio of the variance to the mean of the response would be a good start. Fit a Negative Binomial GLM to these data, interpret your findings, see if the AIC improves, and plot your output.

Extra questions

If you did all the above and have time to spare, try these:

1. Another data sets which is worth fitting a linear regression model to is the `geese_isotopes.csv` data. You might like to see if one of the isotope values is affected by some of the other variables (sex, adult, etc)
2. The whitefly data set. This is binomial (as used in the lectures) but you might like to additionally try some of the other variables and see which are important and why. The data set has particular issues with zero-inflation. See if you can find which zero data points are poorly predicted by the model.