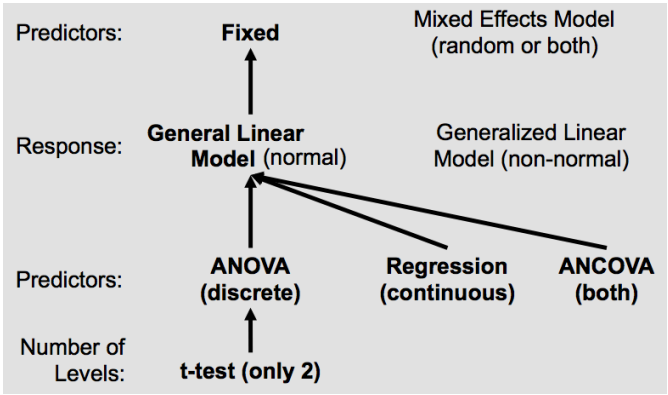
General Linear Models

General linear models are a powerful and flexible statistical approach that encompasses most of the statistics you have probably learned and used so far (e.g. ANOVA, regression, t-test).

What are general linear models?

General linear models are a flexible statistical framework for analyzing a variety of types of data. They encompass many of the statistical approaches that you are probably already familiar with (e.g. ANOVA, regression) and also provide a foundation for dealing with more complex analyses including non-normal error distributions (generalized linear models) and the inclusion of random effects (mixed-effect models).



ANOVA, t-tests, regression and ANCOVA are just special cases of general linear models (glm). A t-test is just an ANOVA with only two levels in the one factor. An ANOVA is just a glm with only categorical predictors. A multiple regression is a glm with only continuous predictors. An ANCOVA has both categorical and (usually only one) continuous predictors.

A general linear model (glm) is a statistical model that predicts some response variable, y, by linear combinations of a variety of predictor variables. For example,

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X2 + \beta_3 X_3 + \beta_4 X_4 + e.$$

How is this different from a multiple regression? It isn't really. The only restriction is that in a multiple regression all of the X's are continuous variables. In an ANOVA all of the X's are categorical. In a general linear model we can use the same approach for all of these cases as well as those where we have a mix of categorical and continuous variables,

How do glms deal with categorical predictors?

I think most people are pretty comfortable with the idea of a multiple regression (after some instruction). Where things get a little bit trickier is when we have

categorical predictors. In the simplest case we might have a binary (two possible values) predictor, such as true/false, alive/dead, or treatment/control. In this case, the Xs might be coded as 0 or 1 to indicate alive and dead. So for example, we might want to compare the body mass of males and female chipmunks. In this case our linear model would be:

$Mass_i = \beta_0 + \beta_1 Sex_i + e_i$.

where the mass of chipmunk i is predicted by whether or not the individual is a male or a female. When we have a categorical predictor in a glm the software uses 0s and 1s to code the categories. Males might be coded as 1 and females as 0. You don't need to do this re-coding. The statistical software will do this for you! In this case the new variable coding is equivalent to the new coding variable asking "Is individual i a male?" If yes, then code with a 1. If no, then code with a zero.

Even though this analysis is essentially the same as a t-test, in the context of a glm we will still estimate the equivalent of an intercept (β_0) and a slope (β_1). What do these mean? Well in order to interpret them we need to know how the software program is coding the variables. If males are coded as 1 and females as zero then the intercept is equal to the mean mass of females and the slope is equal to the difference between the mean mass of males and the mean mass of females.

How do we deal with more than two levels in a factor (i.e. an ANOVA)? We can still use our system of binary yes/no questions but now we need more questions to be able to code more levels using this binary system. We will always need k-1 binary questions in order to completely code k categories. So in our example, let's say that there are some chipmunks that we aren't able to score as male or female so they are coded in the data as unknown. We can code these three levels (female, male, unknown) using two binary (yes/no) questions: 1) is the observation for a male? 2) is the sex unknown? If the answer to both of these questions is no then the observation must be for a female. So a vector of data that look like this:

M
M
F
Unknown

...would be coded as

data		is male?	is unknown?
M		1	0
M		0	0
F		0	0
unknown		0	1

Again, you DO NOT need to set up your datafile in this way. I am just showing you how a categorical variable with 3 levels in interpreted by the statistical package.

Contrasts are used to code categorical predictors.

This coding of a categorical variable is called the ***contrasts*** or ***contrast matrix***. There are many different ways that software packages might set up their contrasts or that you can ask the software to use but most people don't pay any attention to this. They simply use the default contrast matrix for the program they use. People generally don't care about the

contrast matrix because 1) it is all happening in the background when the stats package is doing the math, and 2) it doesn't affect your inference (the significance of your parameters). That said, it is crucial that you know what the contrast matrix is if you are going to make any sense of what the parameters (e.g. β_0, β_1, β_2, β_3) in your model output mean.

R uses a default contrast matrix called ***treatment contrasts***. In this case the first level of your factor is taken to be the control level and all other levels are considered to be treatment levels where their effects are measured relative to the controls. In some cases this makes sense. In other cases you might have a different level that is more logical to use as your reference level (e.g. a control that you would like to compare against several manipulated treatments). You can reorder the levels in a factor in R using the *relevel* function.

S-plus uses a more complicated contrast matrix, called Helmert contrasts. There are apparently some mathematical benefits to using this contrast matrix but it makes it more difficult for the user (biologist) to interpret the meaning of the parameters. The bottom line is that *you do not need to recode your data* but in order to make sense of what the parameters mean in your glm output *you need to know what contrast matrix your stats package is using*.

A worked example:

Let's say we are trying to put together a statistical model to explain variation in the clutch size of some species of bird. We measure clutch size in a bunch of birds as well as some predictor variables that we think might be important. These predictors might be body mass (continuous variable in grams), whether or not the bird was experimentally supplemented with food (binary; yes or no), and what its reproductive history was (categorical, 3 levels: FirstYear, SecondYear, Older bird).

So you might collect data that look like:

clutch size	body mass	supple-mented?	Age Class
4	37.2	No	FY
6	47.3	No	SY
7	28.9	No	FY
3	32.9	Yes	O
5	29.9	Yes	SY
3	32	Yes	FY

These data might then be recoded (in the background by the stats program) to look like:

clutch size	body mass	supple-mented	Is FY?	Is SY?
4	37.2	0	1	0
6	47.3	0	0	1
7	28.9	0	1	0
3	32.9	1	0	0
5	29.9	1	0	1
3	32	1	1	0

The reason your stats package does this recoding is that now you can use linear combinations of these recoded data to predict clutch size:

$y = \beta_0 + \beta_1 BodyMass + \beta_2 Supplement + \beta_3 IsFY? + \beta_4 IsSY? + e.$

Note that you would have indicated to the stats package (e.g. R) that you wanted to include three predictors (mass, supplement and age class) but now you have a model that includes 5 parameters (one intercept and 4 slopes). Some of you who are really thinking might already realize that this is precisely why we use 2 degrees of freedom for a factor that has three levels. This is because we need to estimate two parameters for that factor! In our case we needed to estimate β_3 and β_4 to be able to test the effect of age class.

How are parameters estimated in a glm?

This is about to get technical, so proceed with caution!

You have already seen how factors get recoded in the background, so now you are ready to think about how these parameters (betas) actually get estimated. This is basically a matrix algebra problem. We have a vector of observed values (see below) and matrix of predictors. The only thing we need to add to this matrix is a column of constants. We then want to estimate what the vector of betas is that will minimize our errors (e). In matrix notation $\mathbf{Y=X\beta+e}$. We can rearrange this equation and solve for β.

$\mathbf{e = Y - X\beta}$

$\mathbf{e^2 = (Y - X\beta)^2}$

$\mathbf{e = Y^2 - 2YX\beta + (X\beta)^2}$

We then want to minimize e^2 with respect to $\boldsymbol{\beta}$, which means taking the derivative, set it to zero and solve for β. This is called the least squares method. You have a known **Y** and **X** and you solve for $\boldsymbol{\beta}$.

- $\mathbf{Y=X\beta+\varepsilon}$
- β_0=intercept, β_1=slope for x_1,…

Y
4
6
7
3
…

=

Const	Mass	Supplement	IsFY	IsSY
1	37.2	0	1	0
1	47.3	0	0	1
1	28.9	0	1	0
1	32.9	1	0	0
…				

β
β_0
β_1
β_2
β_3
β_4

$y = \beta_0 + \beta_1 BodyMass + \beta_2 Supplement + \beta_3 IsFY? + \beta_4 IsSY? + e$

Solving the model is a matter of finding the parameters that minimize the errors (i.e. variance in e).

The basic idea is to find the parameters that minimize the errors (residuals, unexplained variation), in our model. This is the same as finding the **'line of best fit'** but now we have some continuous and some categorical variables.

Interpreting the linear model and what the parameters mean

We will go over these ideas in the PDF document on glms.