matings.sas: Explanation of code

Goals of code:

- Fitting models to count data
- Including overdispersion

The data set is matings.csv. We will use case study 22.1 (male elephant mating success). The response variable is the number of matings for each of 41 elephants in an 8 year study of a population in Amboseli Park, Kenya. The questions are whether there is an association between elephant age and the number of matings, and whether this relationship has a detectable maximum. The data are in matings.csv. Load the data.

Fitting models to count data: `model / d=poisson;`
Poisson regression: In donner.sas, we used either proc logistic or proc genmod to fit models to yes/no responses. When the response is a count and you want to fit Poisson regression, you need to use proc genmod. The only change from earlier is to replace `/d=binomial` with `/d=poisson`. You can use class statements to define grouping variables and (if appropriate) define interactions or quadratic terms "on the fly" by age*age.

Events/trials data: Sometimes, yes/no data are summarized as # events out of some number of attempts (or trials). I imagine that each elephant had 24 attempts at mating. Now, the maximum possible value for matings is 24, and the response is modeled as the log odds of event instead of as log mean (as in Poisson regression).

To indicate events and trials, provide two variable names on the left side of the model statement with a `/` in between. You need a variable name even when the number of attempts is the same for all individuals. Use `d=binomial` to indicate a binomial distribution for the response (just like individual yes/no data). Since this is a logistic regression now, you can also use proc logistic instead of proc genmod. proc logistic also allows events/trials syntax.

The genmod output for either Poisson or logistic models includes a large table of model fit statistics. The value/df column has information about overdispersion. The last table provides information about each regression coefficient: estimates, se's, confidence intervals and tests of coefficient = 0.

Including overdispersion: `scale = deviance`
The values in the value/DF column of the model fit statistics provide information about potential overdispersion. If the supplied distribution (poisson or binomial) is appropriate, these values should be close to 1. If substantially larger, the analysis should account for overdispersion. How big is substantially larger is subjective. I always adjust if value/DF is more than 1.2, or when past experience suggests that overdispersion is typical.

You can account for overdispersion in two ways: by rescaling the variance or by using an overdispered

distribution. Rescaling the variance requires adding `scale = deviance` or `scale=pearson` to the model options. The amount of overdispersion is estimated by deviance/DF for scale=deviance and Chi-square/DF for scale=pearson. The standard errors, confidence intervals and tests are adjusted for the overdispersion. This results in larger se's, wider confidence intervals and larger (less significant) p-values.

The alternative is to use a negative binomial distribution for counts, instead of the poisson. That is `d=negbin` in the proc genmod model statement. There is a beta-binomial distribution for overdispersed events/trials, but that is not available in genmod. If you need it, look at proc fmm.