matings.r: Explanation of code

Goals of code:

- Fitting Poisson regression to count data
- Fitting a logistic regression to events and trials data
- Fitting overdispersed count or proportion data

The data are in 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 a Poisson regression: `glm(, family=poisson)`
A Poisson regression is fit using the glm() function, introduced in donner.r. The response variable contains the number of events for that subject. The family= argument specifies a Poisson distribution for the response.

The quantity being modeled is the log of the mean count. The set of helper functions and options described in the explanation of donner.r can be used with a Poisson regression. As with a logistic regression, you have two choices of predicted values: predictions on the link scale and predictions of responses. For a Poisson regression, those are log count (link scale) and count (response scale). The choice is specified by the type= argument to predict.

Fitting a Binomial response with events and trials: `glm(cbind(success, fail)`
If the count of events for each individual is really the number of successes with a known number of trials, the data probably should be modeled as Binomial with a known number of trials. This models the log odds of a success, instead of the log mean count. The difference is especially important when the number of successes is more than 1/2 the number of trials. Although the data start as yes/no for each trial for a subject, the information is completely summarized by the number of yes's and the number of trials.

This sort of data is modeled using glm(, family=binomial). The difference between events/trials and individual yes/no data is how the response variable is specified. For individual yes/no data, the response is one variable with either 1 / 0 or a factor. For events and trials, the response is a pair of vectors. The first vector is the number of successes; the second is the number of **failures**. These assembled into a matrix with two columns with cbind().

Note: R wants successes and failures. Not successes and number of trials! That's why the code computes Attempts - Matings.

Fitting overdispersed count or Binomial data: `glm(, family=quasipoisson)`;
R provides two ways to model overdispersed data. The simplest is to use the quasi distributions. These are family=quasipoisson and family=quasibinomial. These distributions have characteristics like the named distributions except that the variance is some number times the usual variance. That number is called the scale or dispersion parameter. For overdispersed data, the scale parameter is larger than 1. The scale parameter is used to adjust standard errors of coefficients (and hence tests, p-values, and confidence intervals) and changes in deviance.

If you want to evaluate overdispersion, fit the appropriate quasidistribution and look for the line in summary() output that starts (Dispersion parameter .... The number at the end of the line is the estimated scale parameter.

The other way to account for overdispersion is to fit the negative binomial distribution, which assumes a variance that could be larger or smaller than the Poisson variance. In spite of the name, there is no connection between this use of the negative binomial and the binomial distribution for yes/no or events/trials data. The negative binomial model is fit by the glm.nb() function in the MASS library, and by various other functions. The syntax for glm.nb() is identical to the poisson model for glm(), except that glm.nb() only fits a negative binomial distribution, so the family= argument is not needed.

For the matings data, the overdispersion is small (1.15) and the conclusions from overdispersed models qualitatively the same as those from a Poisson model. The choice of overdispersion model doesn't matter (in this case, it can in others).