Generalized linear models for disease ecologists

Ben Bolker

May 21, 2012

1 Introduction

You will need the bbmle, ggplot2, and coefplot2 packages installed, and optionally glmmML, lme4, and glmmADMB. coefplot2 and glmmADMB need to be installed via

```
install.packages("coefplot2",repos="http://r-forge.r-project.org")
install.packages("glmmADMB",repos="http://r-forge.r-project.org")
```

Why linear models? Stats 101: everything is normally distributed (and the residuals have constant variance). If the response variable depends on a *continuous* (as opposed to *categorical*) predictor variable, then the relationship is linear.

Linear models are:

- Fast and numerically stable (works for huge and/or wonky data sets)
- No need for starting values
- Easy to account for *random effects* (experimental blocks etc.)
- Lots of data (especially in economics, business, etc.) is reasonably normal
- Transformations can often fix problems with heteroscedasticity (non-constant variance)/non-normality/non-linearity
- Assuming normality (equivalently *least-squares* solutions) is usually OK *asymptotically* (large data sets)

Why not?

- More specific models are more efficient/powerful
- Would like a model that reflects the data better
- Some data (discrete, zero-rich) are resistant to transformation

- Understanding *effect sizes* on transformed scales is hard
- Linear relationships often force out-of-bounds predictions (proportions outside $\{0, 1\}$; negative counts)
- If you need arguments in favor of GLMs: O'Hara and Kotze (2010); Warton and Hui (2011)

Generalized linear models (GLMs) allow (some) non-normal data (e.g. Poisson, binomial) and (some) nonlinear relationships (e.g. exponential, logistic curves). Retain advantages of linear models (fast, stable, usually don't need starting values . . .)

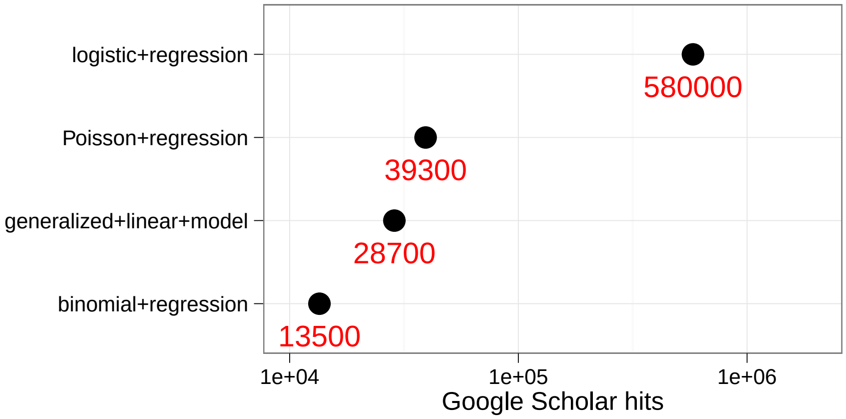
Basic definition Need to specify

- distribution (*family*) (e.g. Poisson=count data, binomial=proportion [count!] data)
- *link function* (e.g. log, logit): linearizing transformation (don't actually transform data)
- response variable and *continuous* and *categorical* predictors (R formulas: `r~x`, `r~x+y` (additive model), `r~x*y` (interaction))

Model is linear on scale of link function.

Almost all GLMs are logistic regressions.

These data were scraped from Google Scholar hits on the relevant search terms.



- logistic regression: *logit* link (logistic inverse-link)
- Poisson regression: *log* link (exponential inverse-link)

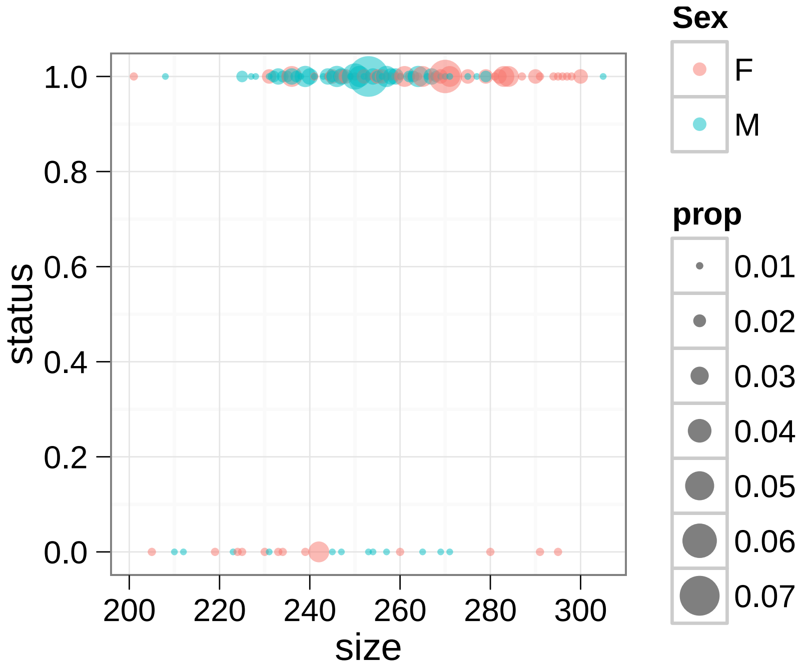
2 Logistic regression

2.1 Preliminaries

Binary (or maybe binomial) data. Logit link (others are possible but usually impossible to distinguish based on data).

Read in the data and look at it:

```r
dat <- read.table("gophertortoise.txt",header=TRUE)
dat2 <- droplevels(subset(dat,Sex %in% c("M","F")))
(gplot1 <- ggplot(dat2,aes(size,status,colour=Sex))+
 stat_sum(alpha=0.5))
```



Visualizing binary data is tricky — here's one way:

```r
dat2$sizecat <- cut(dat2$size,breaks=c(seq(200,280,by=20),320))
proptab <- ddply(dat2,c("sizecat","Sex"),
      function(x) {
          data.frame(n=nrow(x),
                     size=mean(x$size),
                     status=mean(x$status))
      })
gplot1 + geom_point(data=proptab,aes(size=n),shape=2)
```

2.2 Picture and basic fit

We can quickly get `ggplot` to add a GLM fit to the `gplot1` graph by adding `geom_smooth(method="glm",family=binomial)` (try it!). However, this is only convenient for looking at pictures (not for testing hypotheses, finding good predictive models, etc.).

Try a basic glm fit:

```
(mod1 <- glm(status~size*Sex,family=binomial,data=dat2))
##
## Call:  glm(formula = status ~ size * Sex, family = binomial, data = dat2)
##
## Coefficients:
## (Intercept)         size         SexM    size:SexM
##    -10.7490       0.0482       6.6021      -0.0229
##
## Degrees of Freedom: 197 Total (i.e. Null);  194 Residual
## Null Deviance:     158
## Residual Deviance: 142  AIC: 150
```

2.3 Single-model methods

What does R know how to do with mod1? Try this:

```
class(mod1)
methods(class="glm")
```

```
coef(mod1)           ## coefficients
summary(mod1)        ## various summary info
coef(summary(mod1))  ## just the coefficient table with p-values
etc.
confint(mod1)        ## profile confidence intervals
fitted(mod1)         ## fitted values
predict(mod1)        ## predictions ON LINEAR PREDICTOR scale
## ... on response scale (probabilities)
predict(mod1,type="response")
pframe <- data.frame(size=250,Sex="M")
predict(mod1,newdata=pframe)   ## predictions for new data
## simulated data from the fitted model
simulate(mod1)
residuals(mod1)                ## residuals(deviance)
plot(mod1)           ## these are terrible!
AIC(mod1)
deviance(mod1)       ## -2 log L
logLik(mod1)
## SEQUENTIAL analysis of deviance
anova(mod1,test="Chisq")
drop1(mod1,test="Chisq")            ## drop single terms and test
## marginal tests (danger Will Robinson!)
drop1(mod1,test="Chisq",scope=.~.)
```

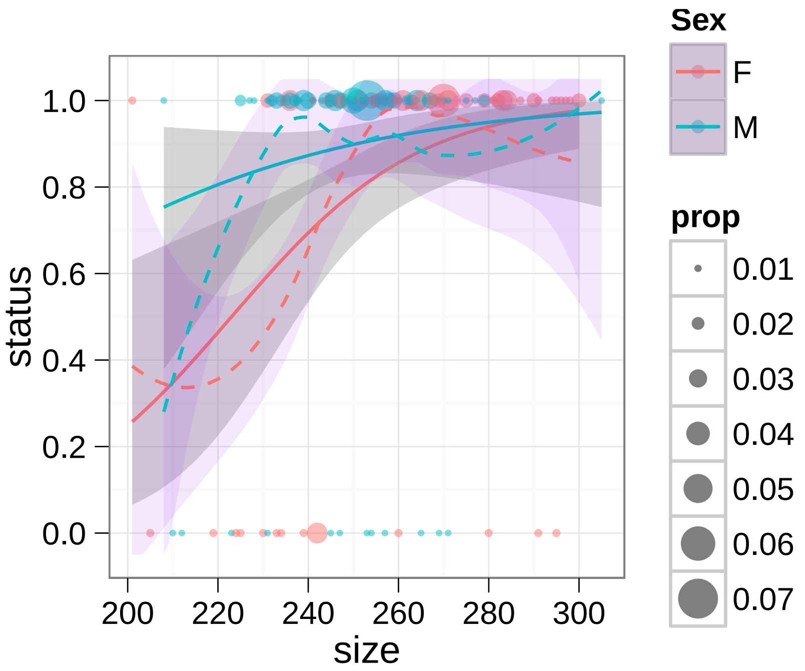
Notes and pitfalls:

- the *p*-values given by `summary` etc. are approximate (Wald tests), those from `drop1`/`anova` (likelihood ratio tests) are better (although still approximate)

- `predict` gives predictions on the linear predictor (logit) scale by default (not probabilities): use `type="response"` for probabilities

- diagnosing lack of fit etc. is hard for binary data. One possibility is to compare the GLM fit with a non-parametric fit, as follows:

```
library(scales)
gplot1+geom_smooth(method="glm",family="binomial")+
    geom_smooth(linetype=2,fill="purple",alpha=0.1)+
    scale_y_continuous(limits=c(-0.05,1.05),oob=squish)
```



(The standard goodness-of-fit test for logistic regression, Hosmer-Lemeshow, has some problems (Hosmer et al., 1997); an improved version is available by using `lrm` in the `rms` package and using `resid(f, 'gof')`)

- `anova` actually gives an analysis of *deviance*, and it is **sequential** ("type I" in SAS language)

- `drop1(glm_fit,scope=.~.)` does a marginal ("type III") analysis, which has its own issues when there are interactions in the model (you need to be *very* careful interpreting the meaning of the main effects)

2.4 Interpreting parameters

What do the parameters mean???

People like me are always complaining that researchers should consider *effect size*, the biological significance of the parameters, not just the statistical significance. In order to do this, you need to understand what the parameters mean.

There are two hard parts of interpreting GLM parameters: (1) contrasts (how R parameterizes the differences between groups) (this issue is general to all modeling in R) and (2) the log-odds scale (this is specific to logistic regression).

```
coef(mod1)
## (Intercept)        size        SexM   size:SexM
##   -10.74901     0.04820     6.60211    -0.02289
```

intercept (≈ -10): the logit probability (log-odds) of seropositivity for a female with size$=0$ (i.e., log-odds in the *baseline* condition)

size (≈ 0.05): increase in log-odds per unit (mm) of size, *for females* (baseline)

SexM (≈ 6.6): difference between females and males at size 0

size:sexM (≈ -0.02): difference between male and female slope, on the log-odds scale

```
 ## probability of infection of a female, size=0
plogis(-10.7)
## [1] 2.254e-05
## probability of infection of a female, size=100
plogis(-10.7+0.048*100)
## [1] 0.002732
## probability of infection of a male, size=0
plogis(-10.7+6.6)
## [1] 0.0163
## probability of infection of a male, size=100
plogis(-10.7+6.6+(0.048-0.023)*100)
## [1] 0.168
```

```
dat2$csize <- dat2$size-250
mod1c <- update(mod1,.~csize*Sex)
```

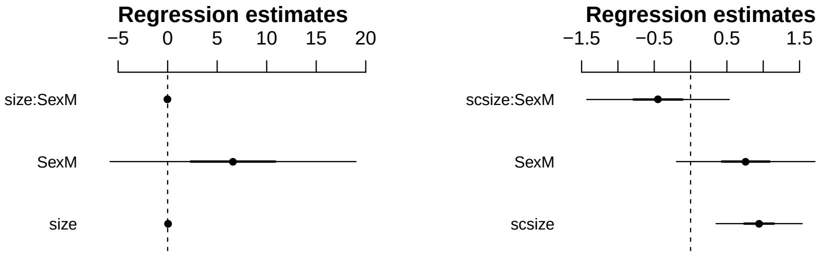
Exercise: Confirm for yourself that the `Intercept` and `SexM` terms have changed, but not the slopes.

Sometimes it's more useful to scale the continuous predictors as well; one easy procedure is to center (subtract the mean) and divide by the standard deviation of the predictor, which puts all of the predictors on a similar scale (Schielzeth, 2010). (The `scale` function in R also does this, but with some side effects we don't want.)

```
dat2$scsize <- with(dat2,(size-mean(size))/sd(size))
mod1sc <- update(mod1,.~scsize*Sex)
```

Which of the following plots is more useful?

```
library(coefplot2)
par(mfrow=c(1,2))
coefplot2(mod1)
coefplot2(mod1sc)
```



Exercise: . Change the baseline level from females to males (the default is alphabetical) by using

```
dat2$Sex <- relevel(dat2$Sex,"M")
```

Refit either the scaled or the unscaled model and convince yourself that you understand how the parameters have changed (and that the overall meaning of the model has not changed).

2.5 Comparing models

Most GLM inference etc. is based on comparing *multiple* models rather than a single model (the likelihood ratio test. Allow for the interaction of age × sex:

```
mod2sc <- update(mod1sc,.~.-scsize:Sex)
mod3sc <- update(mod1sc,.~scsize)
mod4sc <- update(mod1sc,.~Sex)
mod5sc <- update(mod1sc,.~1)
anova(mod1sc,mod2sc,mod3sc,mod5sc,test="Chisq")
## Analysis of Deviance Table
##
## Model 1: status ~ scsize + Sex + scsize:Sex
## Model 2: status ~ scsize + Sex
## Model 3: status ~ scsize
## Model 4: status ~ 1
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       194        142
## 2       195        143 -1    -0.81   0.3686
## 3        196          148 -1     -4.77    0.0290 *
## 4        197          158 -1    -10.11    0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

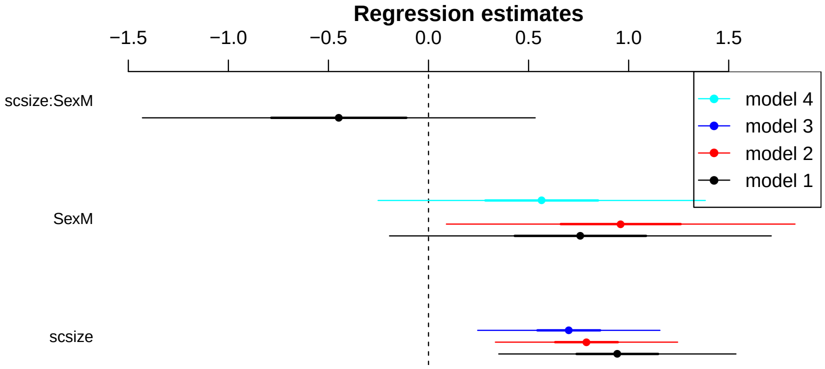
(the p-values are the tests of each model against the next, i.e. significance tests of the parameters that are being dropped)

Or:

```
library(bbmle)
AICtab(mod1sc,mod2sc,mod3sc,mod4sc,weights=TRUE)
##         dAIC df weight
## mod2sc  0.0 3  0.55374
## mod1sc  1.2 4  0.30517
## mod3sc  2.8 2  0.13885
## mod4sc 11.0 2  0.00224
```

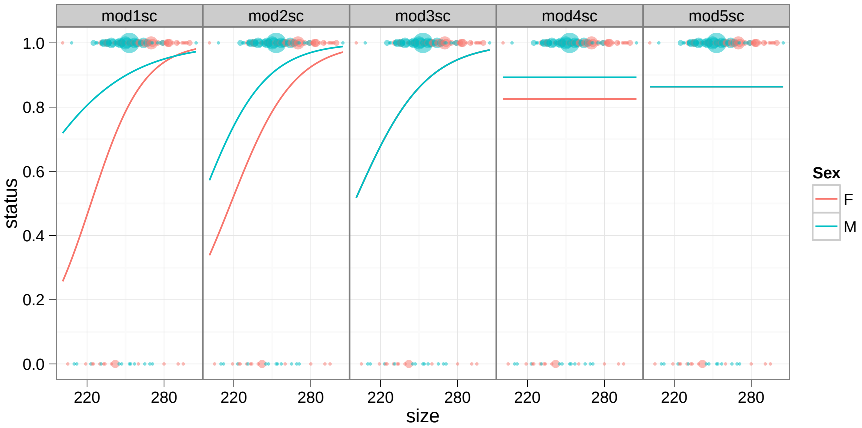
Or:

```
coefplot2(list(mod1sc,mod2sc,mod3sc,mod4sc),
          legend=TRUE,legend.x="topright",
          col=c(1,2,4,5))
```



(I should probably have named these models more informatively!)

Model comparison:



Exercise: : convince yourself that the overall fit of the model (based on `logLik`) and the likelihood ratio test of the full model (comparing the full model with the null/constant model) do not change when you change the parameterization (either by scaling size, or by changing the baseline level).

The logit/log-odds scale is initially hard to understand, but it has really useful properties for estimating changes in risk. The hardest part is that the effects of changes in a predictor on probability depending on the baseline risk.

If you need to, you can use the `dredge` function from the `MuMIn` package to fit all subsets of a model . . .

Some rules of thumb:

- When the starting probability is very low, the logistic curve is approximately exponential, so parameters approximately describe proportional changes (e.g. parameter of 0.1 ≈ 10% (*relative*) increase in probability per unit change)

- when the starting probability is intermediate (say 0.3-0.7), the *absolute* change in probability per unit change is $r/4$ (→ parameter of 0.1 implies 0.025 increase in probability per unit change)

- when the starting probability is high the change in complementary risk (probability of the event *not* happening) changes proportionally

2.6 Binomial regression

If you have $N > 1$ per category, you can either specify the results

- as $(k, N-k)$ (e.g. `cbind(num_dead,num_alive)~x+y+z`); you would often compute this on the fly, as `cbind(num_dead,num_total-num_dead)~x+y+z`).

- with the `weights` specification, e.g. `prop~x+y+z,weights=num_total` or `num_dead/num_tot~x+y+z,weights=num_total`

Exercise: : aggregate the data by size class.

```
library(plyr)
sizetab <- ddply(dat2,c("size","Sex"),
                function(x) c(tot=nrow(x),pos=sum(x$status)))
sizetab$prop <- sizetab$pos/sizetab$tot
```

use `sizetab` to run the analysis above as a binomial model. The parameters should stay the same, and the *differences* in log-likelihood, deviance, and AIC among models should be the same, but the baseline values of log-likelihood etc. will differ.

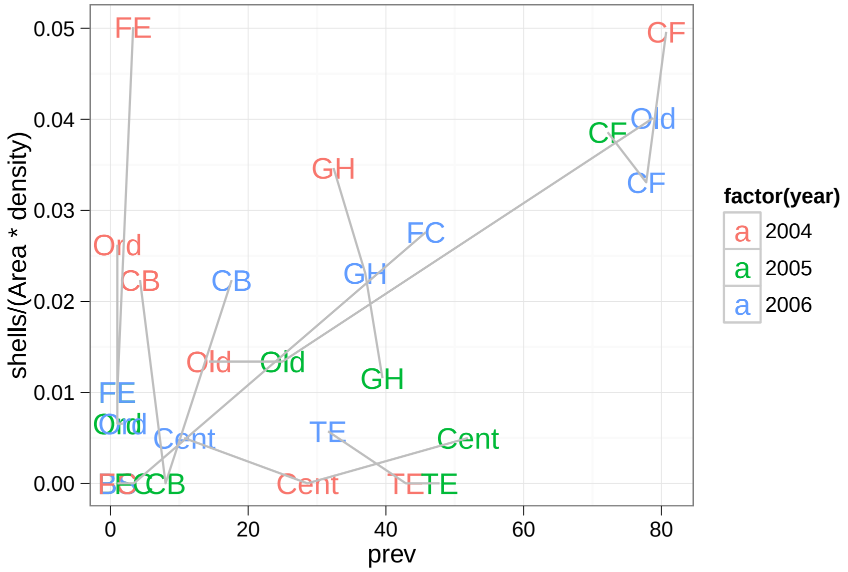
3 Poisson/negative binomial regression

Gopher tortoise shell data.

```
load("gopherdat2.RData")
head(Gdat)

##    Site year shells  type Area density prev
## 2    BS 2004      0 Fresh 15.2     4.8  1.0
## 4    BS 2005      0 Fresh 15.2     4.8  1.0
## 6    BS 2006      0 Fresh 15.2     4.8  1.0
## 9    CB 2004      1 Fresh 16.0     2.8  4.3
## 11   CB 2005      0 Fresh 16.0     2.8  8.0
## 13   CB 2006      1 Fresh 16.0     2.8 17.6
```

```
ggplot(Gdat,aes(x=prev,y=shells/(Area*density),
                colour=factor(year),group=Site))+
    geom_text(aes(label=Site))+geom_line(colour="gray")
```



Fit a Poisson model (log link), with an *offset* of area times density. An offset adds a term to the linear predictor, without fitting the parameters: if P=prevalence, a=area, d=density

$$\begin{aligned}
\log(\lambda) &= \beta_0 + \beta_1 P + \log(ad) \\
\log(\lambda) - \log(ad) &= \beta_0 + \beta_1 P \\
\log\left(\frac{\lambda}{ad}\right) &= \beta_0 + \beta_1 P
\end{aligned}$$

```
m_pois_prev <-
glm(shells~prev+offset(log(Area*density)),data=Gdat)
```

A Poisson model assumes variance=mean, which is usually wrong. There is often *overdispersion*: we can test for it by looking at the sum of squared Pearson residuals ($\sum(\text{expected}_i - \text{obs}_i)^2/\text{var}_i$, where var_i is the expected variance of point i based on the model), which should be (*approximately*) $\sim \chi^2_{n-p}$ if the data are really Poisson:

```
devsq <- sum(residuals(m_pois_prev,type="pearson")^2)
devsq/m_pois_prev$df.resid  ## ratio should be approx. 1
## [1] 3.438
pchisq(devsq,df=m_pois_prev$df.resid,lower.tail=FALSE)
## [1] 2.032e-09
```

There are (at least) two ways to account for the overdispersion, by fitting a quasi-likelihood model

```
library(MASS)
m_quasi_prev <- update(m_pois_prev,family="quasipoisson")
m_nb_prev <-
glm.nb(shells~prev+offset(log(Area*density)),data=Gdat)
```

Exercise: compare these three models (Poisson, quasi-Poisson, negative binomial) looking at `coef(summary(fit))` and at `coefplot2(list(model1,model2,model3), xlim=c(-0.01,0.08))`. What do you conclude about the coefficient estimates?

The `aod` package incorporates a wider spectrum of methods and tests for dealing with overdispersion (which, oddly, is more widely recognized in Poisson than in binomial models).

4 Intermediate topics

4.1 Offsets

Offsets add known components to the model (as shown above). They're most commonly used to account for unequal sampling areas/times, in cases where we expect results to be *strictly proportional* to the offset (time, area sampled). This is often a way to handle ratios (e.g. sibling negotiations per chick in a study of begging behavior by owlets) without losing the discrete nature of the response.

Can also be used in tricky ways, e.g. to fit the Ricker model.

Suppose we think $N(t+1) = aN(t)e^{-bN(t)}$. On the log scale this is

$$\log N(t+1) = \log a + \log N(t) - bN(t)$$

This is a linear equation in $N(t)$ plus an offset of $\log N(t)$. In other words, if we use a log link then we can fit `N ~ Nprev + offset(log(Nprev))` (The intercept term $\log a$ and the slope $-b$ are implicit in the R formula.) **Exercise:** What model would we fitting if we included $\log N$ as a variable rather than an offset?

4.2 Alternative link functions

Don't have to use the standard link functions. We could try to use them to get a slightly better fit to the shape of the nonlinearity, but I use them more often as a slightly tricky way to fit more *mechanistic* models (see e.g. Strong et al. (1999)).

Warning: you're more likely to run into convergence problems, be asked to specify starting values, etc. when using non-standard link functions.

- Holling type II functional response via the *inverse* link: if number eaten is $N_e = aN/(1+ahN)$, then risk of being eaten is $p = N_e/N = a/(1+ahN)$. If we use the inverse link then we are fitting $(1/p) = 1/a + (1/h) \cdot N$ — this linear in N ... (we can even fit Holling type III responses by doing a bit more algebra and including $1/N$ as a predictor)

- Chain binomial: suppose the probability of infection if $p = 1 - \exp(-\beta I_t)$. Then $1 - p = \exp(-\beta I_t)$, or in other words the log of the probability of *not* being infected is $-\beta I_t$. If we have chain-binomial data, then, we can fit

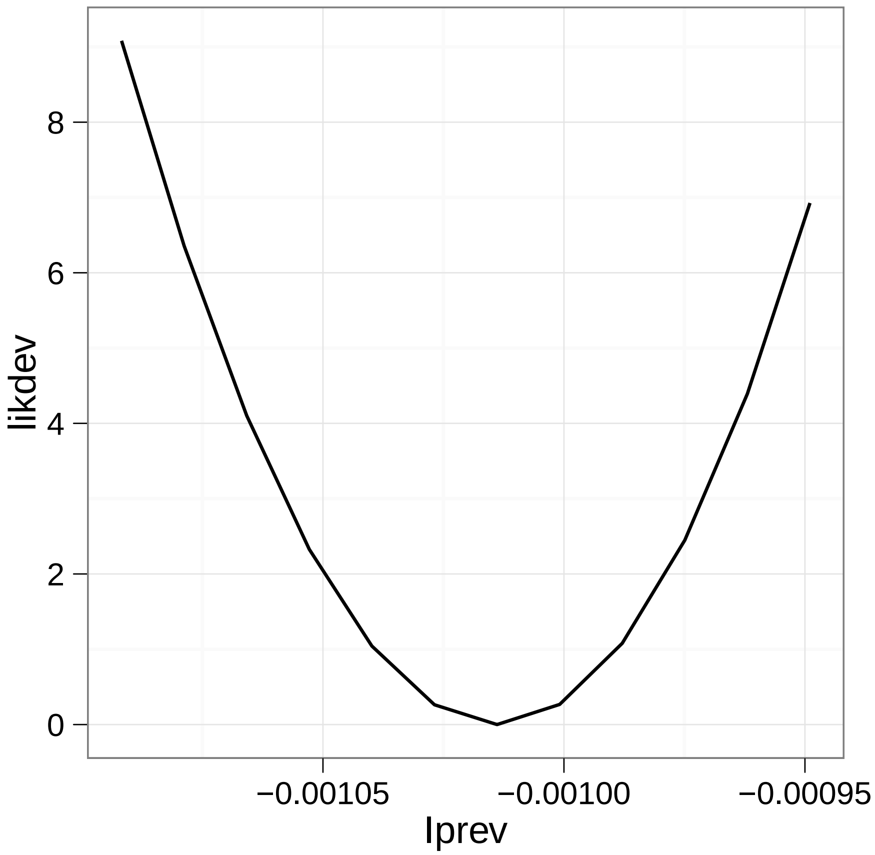
  ```
  glm(prob_not_inf ~ previous_inf-1, weights=previous_susc,
  family=binomial(link="log"))
  ```

```
load("cbsim.RData")
cbsim$Sprev <- c(NA,cbsim$S[-nrow(cbsim)])
cbsim$Iprev <- c(NA,cbsim$I[-nrow(cbsim)])
cbsim$pnotinf <- cbsim$S/cbsim$Sprev
cbplot <- ggplot(cbsim,aes(pnotinf,Iprev))+geom_point()
fit1 <- glm(pnotinf ~ Iprev -1 , family=binomial(link="log"),
    weights=Sprev,data=cbsim)
confint(fit1)

##      2.5 %     97.5 %
## -0.0010641 -0.0009653

##
cbplot+geom_smooth(method="glm",family=binomial(link="log"),
```

```
##                     formula=~x-1)
pp <- profile(fit1)
likdat <-
data.frame(likdev=pp$Iprev$z^2,Iprev=pp$Iprev$par.vals[,"Iprev"])
ggplot(likdat,aes(x=Iprev,y=likdev))+geom_line()
```



```
## chain-binomial data from Becker:
cbdat <-
data.frame(chains=c(1,11,111,12,1111,112,121,13,11111,
                    1112,1121,113,1211,122,131,14),

n=c(423,131,36,24,14,8,11,3,4,2,2,2,3,1,0,0))  ## total 664
## Greenwood, no slope; Reed-Frost, no intercept
## log(m_{ij} q_{ij})= log(m_{ij})+alpha_i+beta_i j
```

This is discussed by Becker (1989); you can see some of it on Google books at `http://tinyurl.com/chainbinom`

- Another useful case is the *complementary log-log* (`cloglog`) link with binomial data. Suppose we measure the amount of mortality over differing exposure periods of length Δt. If we use the `cloglog` link with an offset of $\log(\Delta t)$ we get the right behavior.

  The `cloglog` function is $C(x) = \log(-\log(1-\mu))$, its inverse is $C^{-1}(x) = 1-\exp(-\exp(x))$. Thus if we expect mortality μ over a period $\Delta t = 1$ and the linear predictor $\eta = C^{-1}(\mu)$ then $C^{-1}(\eta+\log\Delta t) = (1-\exp(-\exp(\eta)\cdot\Delta t))$, which is what we want.

4.3 Bias-reduced GLM

Separation of variables refers to the situation where some threshold value of predictor(s) can perfectly separate the 0 and 1 responses — in this case the maximum likelihood solution doesn't exist and `glm` breaks.

The `brglm` (bias-reduced GLM) and `logistf` (Firth logistic) packages implement a specific solution to this problem; more generally, it tends to provide more reliable results for very small data sets (people tend to forget that GLM relies, more heavily than regular linear models, on asymptotic assumptions) — but it currently works only for binomial models.

Another option is to use Bayesian GLMs (e.g. `bayesglm` from the `arm` package); adding even a weak Bayesian prior can stabilize the fit.

4.4 Generalized linear mixed models

This can be a fairly hairy topic in general (see Bolker et al. (2009), and `http://glmm.wikidot.com/faq`, but at root the concepts are fairly simple: here we add a *random effect* of tortoise ID to the model.

What is a random effect? There are actually several possible, overlapping answers:

- Effects that are drawn (randomly?) from a larger population of possible effects
- Effects where we are interested in the distribution of the levels (variance among levels), rather than the values of specific levels
- Effects that are "nuisance" aspects of the experimental design
- Effects that we want to estimate with *shrinkage*, i.e. pulling poorly estimated values toward the population average

In general fitting as random effects works best when we have many levels with small and uneven amounts of data per level; it works very poorly when there are fewer than 4–6 levels. Typical examples: experimental blocks (spatial or temporal); genotypes or individuals within populations; taxa (species, genera) within higher-level taxa (genera, families). The interactions of fixed and random effects get treated as random; for example, among-pond variation in trends over time (around the population-level average trend).

```
library(lme4)
(mod1mix <-
glmer(status~scsize*Sex+(1|TortID),family=binomial,data=dat2))
```

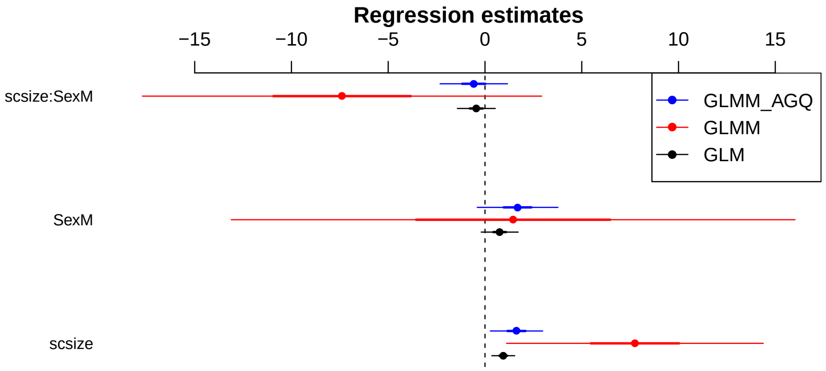
```
## Generalized linear mixed model fit by the Laplace approximation
## Formula: status ~ scsize * Sex + (1 | TortID)
##    Data: dat2
##  AIC BIC logLik deviance
##  120 137  -55.2      110
## Random effects:
##  Groups Name        Variance Std.Dev.
##  TortID (Intercept) 288      17
## Number of obs: 198, groups: TortID, 123
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)     8.53       4.67    1.82    0.068 .
## scsize          7.75       3.38    2.29    0.022 *
## SexM            1.45       7.43    0.20    0.845
## scsize:SexM    -7.39       5.26   -1.40    0.161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) scsize SexM
## scsize       0.285
## SexM        -0.629 -0.179
## scsize:SexM -0.183 -0.643  0.381
```

```
library(glmmML)
(mod1mixB <-
glmmML(status~scsize*Sex,cluster=TortID,family=binomial,data=dat2,
                    method="ghq"))
```

```
##
## Call:  glmmML(formula = status ~ scsize * Sex, family = binomial, data = dat2,      clust
##
##
##                coef se(coef)      z Pr(>|z|)
## (Intercept)  2.617    0.894  2.926   0.0034
## scsize       1.629    0.690  2.360   0.0180
## SexM         1.684    1.064  1.583   0.1100
## scsize:SexM -0.578    0.889 -0.651   0.5200
##
## Scale parameter in mixing distribution:  2.72 gaussian
## Std. Error:                             1.07
##
```

```
##          LR p-value for H_0: sigma = 0:  0.00173
##
## Residual deviance: 133 on 193 degrees of freedom  AIC: 143
```

```
coefplot2(list(GLM=mod1sc,GLMM=mod1mix,GLMM_AGQ=mod1mixB),
          legend=TRUE,legend.x="topright",
          col=c(1,2,4))
```



Things to keep in mind:

- getting reliable p values etc. is hard, unless you have so many groups in your random effect (> 40) that the finite-size correction doesn't matter
- it's easy to put make a model that's too complex to fit reliably. In the example above we have to use the more
- there are a variety of packages that can fit GLMMs, with overlapping capabilities (`lme4`, `glmmML`, `glmmADMB`, `MCMCglmm`) — if possible, it's good to fit difficult models with more than one package, to cross-check

5 Top 10 GLM mistakes

- applying discrete models (Poisson, binomial) to non-discrete data
- ignoring overdispersion
- equating negative binomial with binomial rather than Poisson
- using GLMs where linear models will do (i.e. `glm` instead of `lm`) (harmless but annoying)
- ignoring blocking factors (failing to use GLMMs where necessary)
- confusion in interpreting effects
- worrying about marginal rather than conditional distributions of data*

- applying ± standard errors
- using (k, N) rather than $(k, N-k)$ in binomial models
- getting confused by predictions on the linear predictor scale

6 Topics left out

- Zero-inflated/hurdle models (`pscl`, `tweedie`, `glmmADMB`);
- generalized additive models (`mgcv`);
- penalized regressions and shrinkage methods (`glmnet`, `penalized`);
- polytomous/ordinal data;
- spatial/temporal/phylogenetic correlations;
- GLMs on **big** data (`biglm`) ...

See also slides at `http://www.slideshare.net/bbolker/`, in particular `http://www.slideshare.net/bbolker/glms-and-extensions-in-r`, `http://www.slideshare.net/bbolker/trondheim-glmm`, `http://www.slideshare.net/bbolker/opensource-glmm-tools-7562082`

References

Becker, N. G. (1989, May). *Analysis of Infectious Disease Data.* CRC Press.

Bolker, B. M., M. E. Brooks, C. J. Clark, S. W. Geange, J. R. Poulsen, M. H. H. Stevens, and J. S. White (2009). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology & Evolution 24*, 127–135.

Hosmer, D. W., T. Hosmer, S. Le Cessie, and S. Lemeshow (1997, May). A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine 16*(9), 965–980. PMID: 9160492.

O'Hara, R. B. and D. J. Kotze (2010, June). Do not log-transform count data. *Methods in Ecology and Evolution 1*(2), 118–122.

Schielzeth, H. (2010). Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution 1*, 103–113.

Strong, D. R., A. V. Whipple, A. L. Child, and B. Dennis (1999). Model selection for a subterranean trophic cascade: root-feeding caterpillars and entomopathogenic nematodes. *Ecology 80*, 2750–2761.

Warton, D. I. and F. K. C. Hui (2011, January). The arcsine is asinine: the analysis of proportions in ecology. *Ecology 92*, 3–10.