How to deal with count data?

Ben Maslen

Stats Central

Mark Wainwright Analytical Centre

April 11, 2019

How to deal with count data?

- Properties of count data - GLMs and link functions
- Poisson regression
- Overdispersion
- Negative binomial regression
- offsets
- Binomial count data
- Extensions

Example: Revegetation counts

Anthony wants to evaluate how well invertebrate communities are re-establishing following bush regeneration efforts. Here are some worm counts from pitfall traps across sites:

Treatment	C	R	R	R	C	R	R	R	R	R	C	R	R	R	...
Count	0	3	1	3	1	2	12	1	18	0	0	5	0	2	...

(C=control, R=bush regen)

Is there any evidence that bush regeneration (revegetation) is working?

Count data = Discrete Data

continuous: quantitative data that can take any value in some interval
$\Rightarrow$ **linear models**

discrete: quantitative data that takes a "countable" number of values (*e.g.* $0, 1, 2, \ldots$) $\Rightarrow$ **generalised linear models (GLMs)**

If your data are discrete but the counts are all fairly large, you can ignore the discreteness and use linear models anyway. If you have small counts and zeros though it is very important to use GLMs instead.

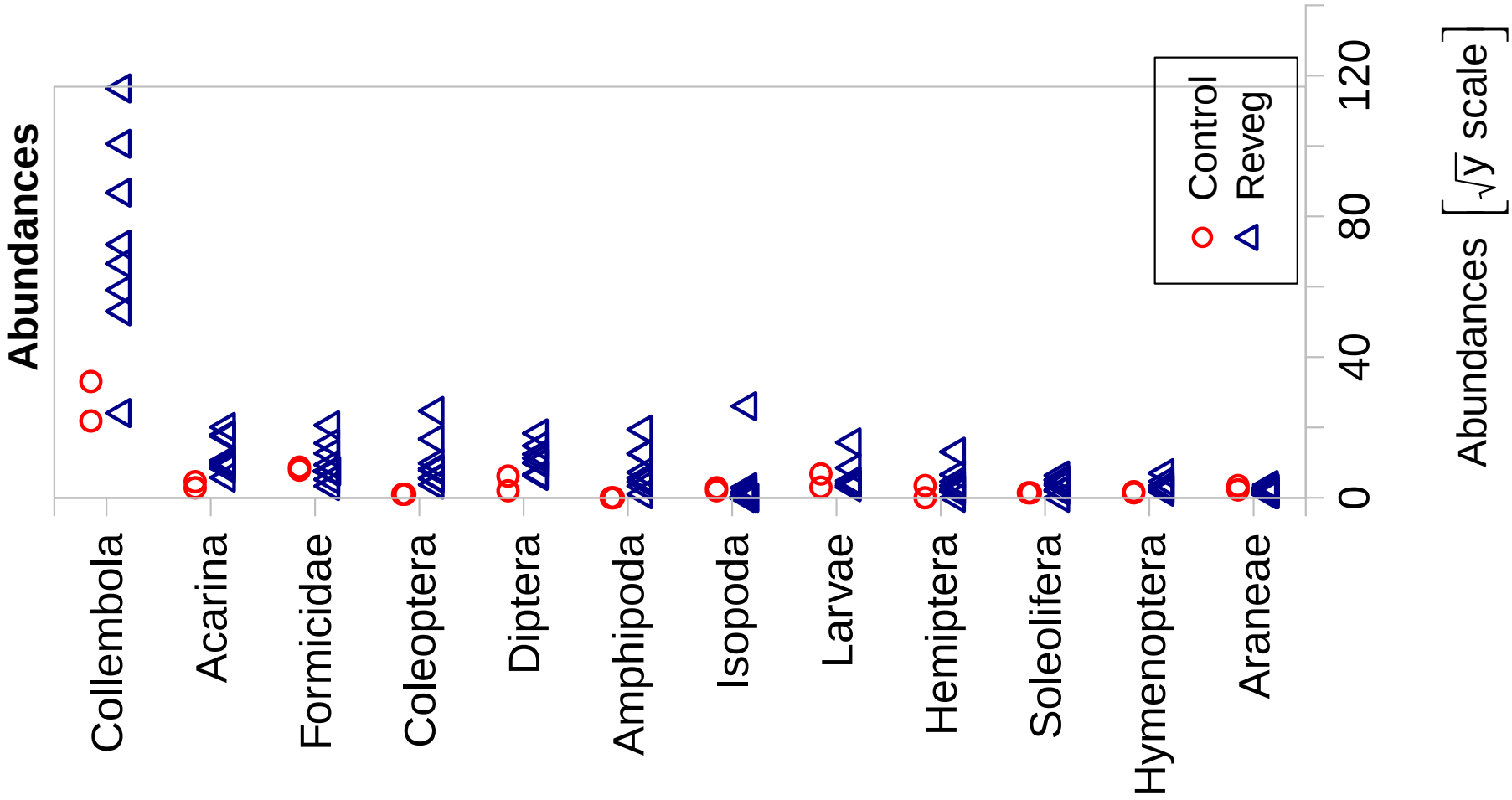
Why does discreteness matter?

The main reason it matters is because it tends to induce a **mean-variance relationship** – as the mean changes the variance changes. When you have zeros and small counts you will have trouble getting rid of the mean-variance relationship via transformation.

This violates the linear model assumption of constant variance.

Why does it happen? Boundaries. Often occurs within experiments with low counts.

Anthony's revegetation count data (all taxa):



GLMs – relaxing linear modelling assumptions

Recall that linear models make the following assumptions:

1. The observed y values are **independent**, conditional on x
2. The y values are **normally distributed** with **constant variance**

$$y \sim \mathcal{N}(\mu_y, \sigma^2)$$

3. **straight line relationship** between mean of y and each x

$$\mu_y = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$$

Generalised linear model

Generalised linear models (GLMs) extend linear models to non-normal data. A GLM makes the following assumptions:

1. The observed y values are **independent**, conditional on x
2. The y values come from **a known distribution** (from the exponential family) with known **mean-variance relationship** $V(\mu)$
3. straight line relationship between **some known function of the mean** of y and each x

$$g(\mu_y) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$$

The function $g(\cdot)$ is known as **the link function**.

Basically, a GLM adds two features to linear models:

- a **mean-variance relationship** $V(\mu)$, in place of constant variance
- a **link function** $g(\cdot)$ used to transform the mean before assuming linearity

What distributions can I use?

Not all distributions can be used with generalised linear models, but a few important ones for count data can:

Poisson this should be your "default" for counts

negative binomial well, this is kind of a GLM. Great for count data which is too variable to fit a Poisson.

binomial for presence/absence data, or "x-out-of-n" counts across n independent events.

There are more distributions that you could use...

Mean-variance relationship

Each of these distributions assumes a special mean-variance relationship for the response variable:

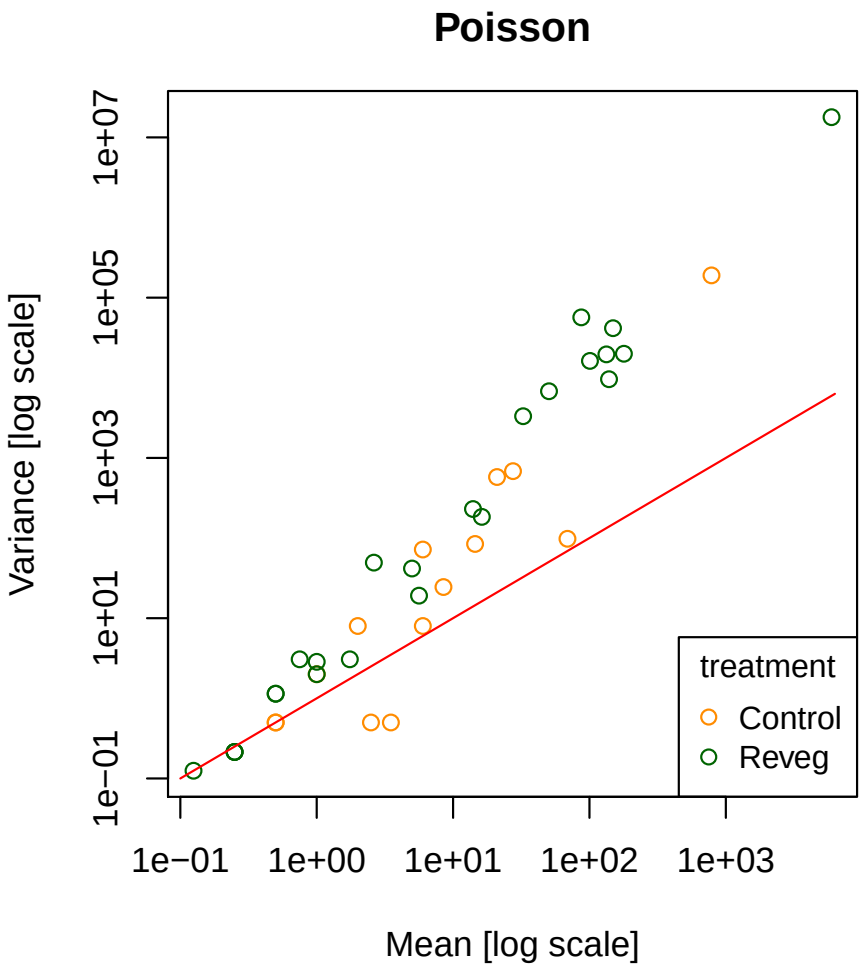
Poisson $V(\mu) = \mu$. Watch out for this assumption, it can be quite restrictive.

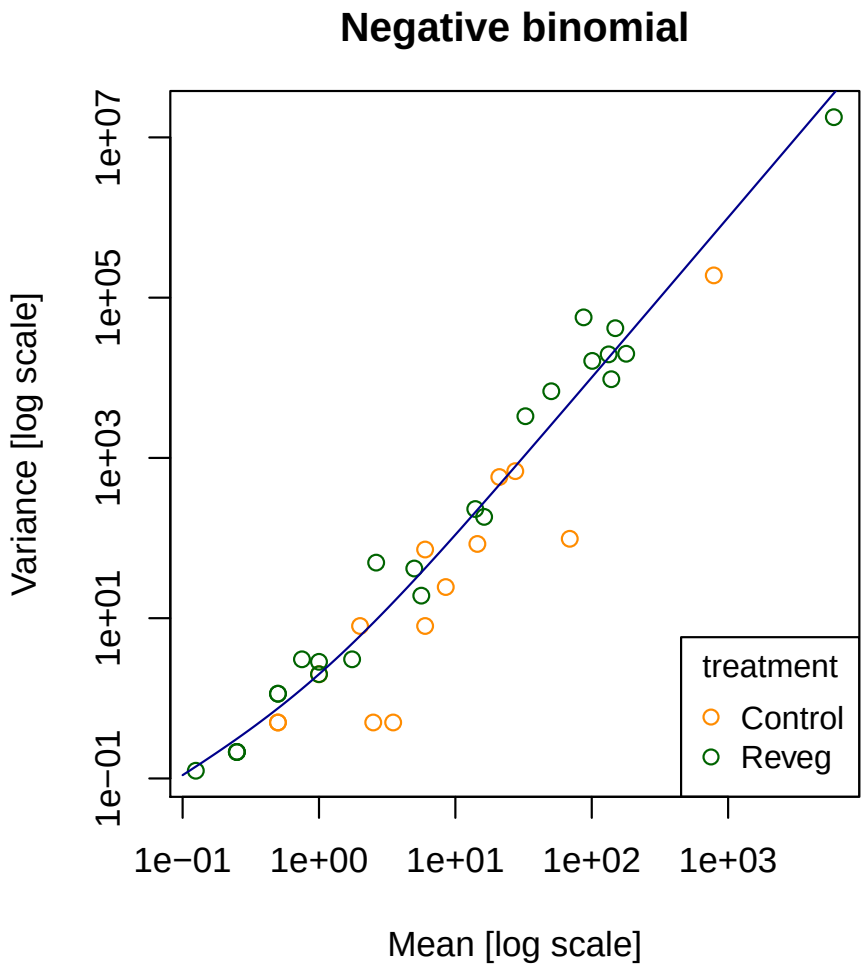
negative binomial $V(\mu) = \mu + \phi\mu^2$. There is a parameter in there (ϕ, the "overdispersion parameter") which controls the degree to which the data are more variable than a Poisson (Usually $0 \leq \phi < 1$, if $\phi = 0$ then data are Poisson)

binomial $V(\mu) = n\mu(1-\mu)$. For presence-absence data, $n = 1$ so this simplified to $V(\mu) = \mu(1-\mu)$

Mean-variance assumptions for reveg counts.

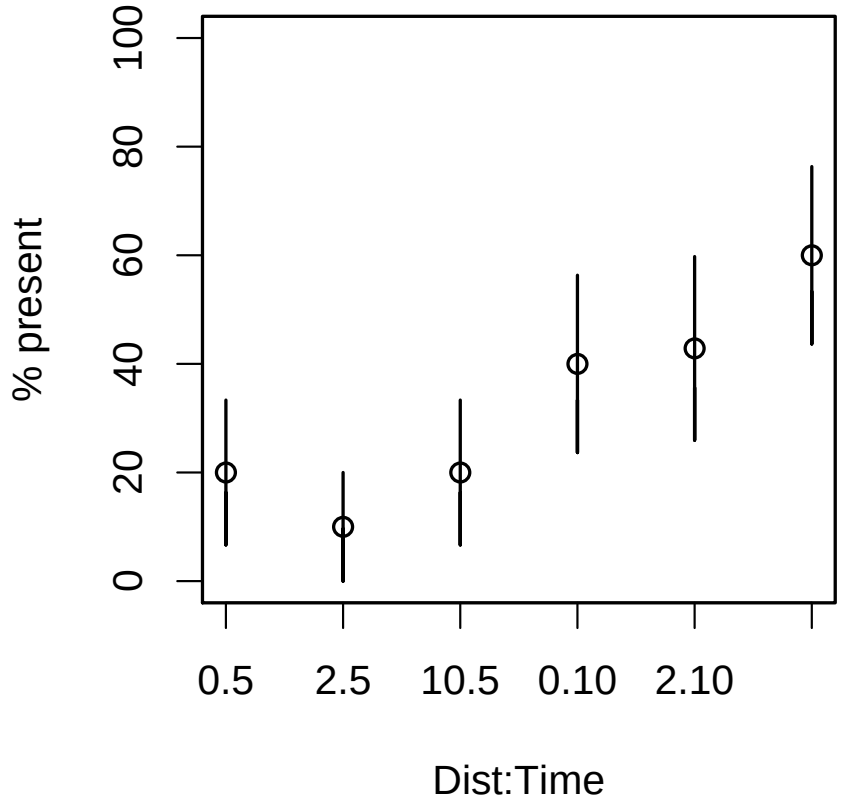
Which mean-variance assumption looks more plausible?

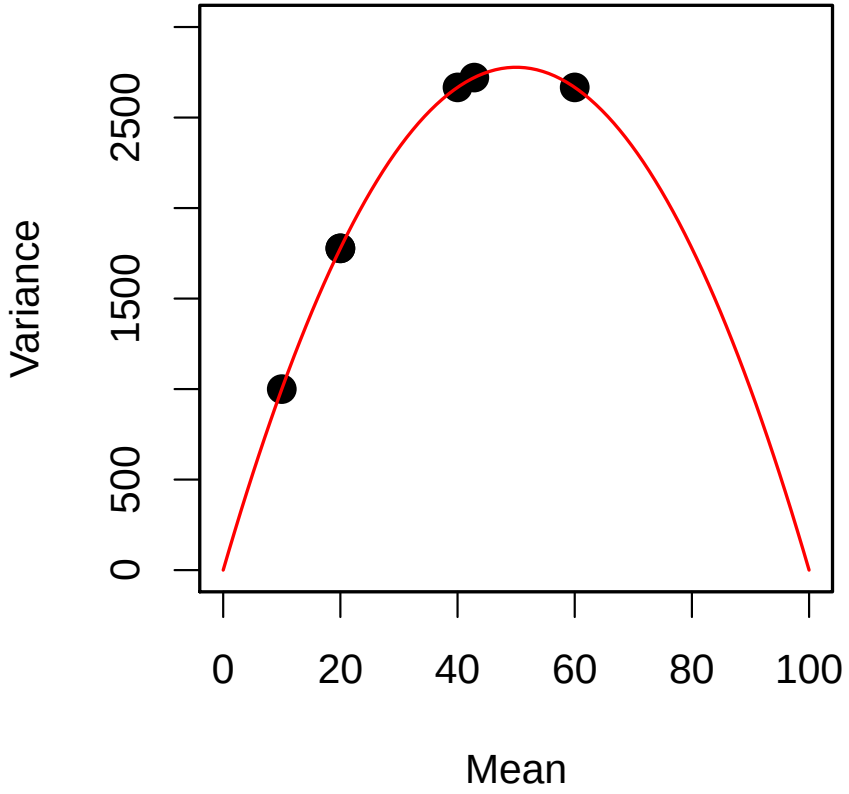




Binomial mean-variance relationship – note that it is **exactly** quadratic

$V(\mu) = n\mu(1-\mu)$





Link functions

Below are the common link functions used for different distributions.

Poisson the log-link, $\log(\mu) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$, is almost always used. This gives us a multiplicative model, often called a "log-linear model".

negative binomial usually the log-link, $\log(\mu) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$.

binomial the logit function $\text{logit}(\mu) = \log(\frac{\mu}{1-\mu}) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$. This is multiplicative in terms of the "odds".
Sometimes the probit function $\Phi^{-1}(\mu)$ where Φ is the probability function of the standard normal.
Sometimes the complementary log-log link, $\log(-\log(1-\mu))$ (if you had Poisson log-linear counts which you truncated to pres/abs).

Family argument

Use the family argument in the following ways:

Poisson log-linear `family=poisson`

logistic regression `family=binomial`

probit regression `family=binomial(link="probit")`

complementary log-log regression `family=binomial(link="cloglog")`

Poisson linear `family=poisson(link="identity")`

For the negative binomial and tweedie distributions, you need to use a special package.

Remember to set the family argument! If you forget it, `glm` defaults to `family=gaussian` (a linear model).

Poisson Regression e.g.

Anthony wants to evaluate how well invertebrate communities are re-establishing following bush regeneration efforts. Here are some worm counts from pitfall traps across sites:

Treatment	C	R	R	R	C	R	R	R	R	R	C	R	R	R	...
Count	0	3	1	3	1	2	12	1	18	0	0	5	0	2	...

(C=control, R=bush regen)

Is there any evidence that bush regeneration (revegetation) is working?

Poisson Regression

You use the `glm` function:

```
> data_revS = read.csv("data/revegSmall.csv")
> data_revS$Treatment <- as.factor(data_revS$Treatment)
> hap_pois <- glm(Haplotaxida~Treatment,family="poisson",data=data_revS)
> anova(hap_pois,test="Chisq")
Analysis of Deviance Table

Model: poisson, link: log

Response: Haplotaxida

Terms added sequentially (first to last)


          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                         29     174.94
Treatment  1    31.46        28     143.48 2.036e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

GLM assumptions

Generalised linear models (GLMs) extend linear models to non-normal data. A GLM makes the following assumptions:

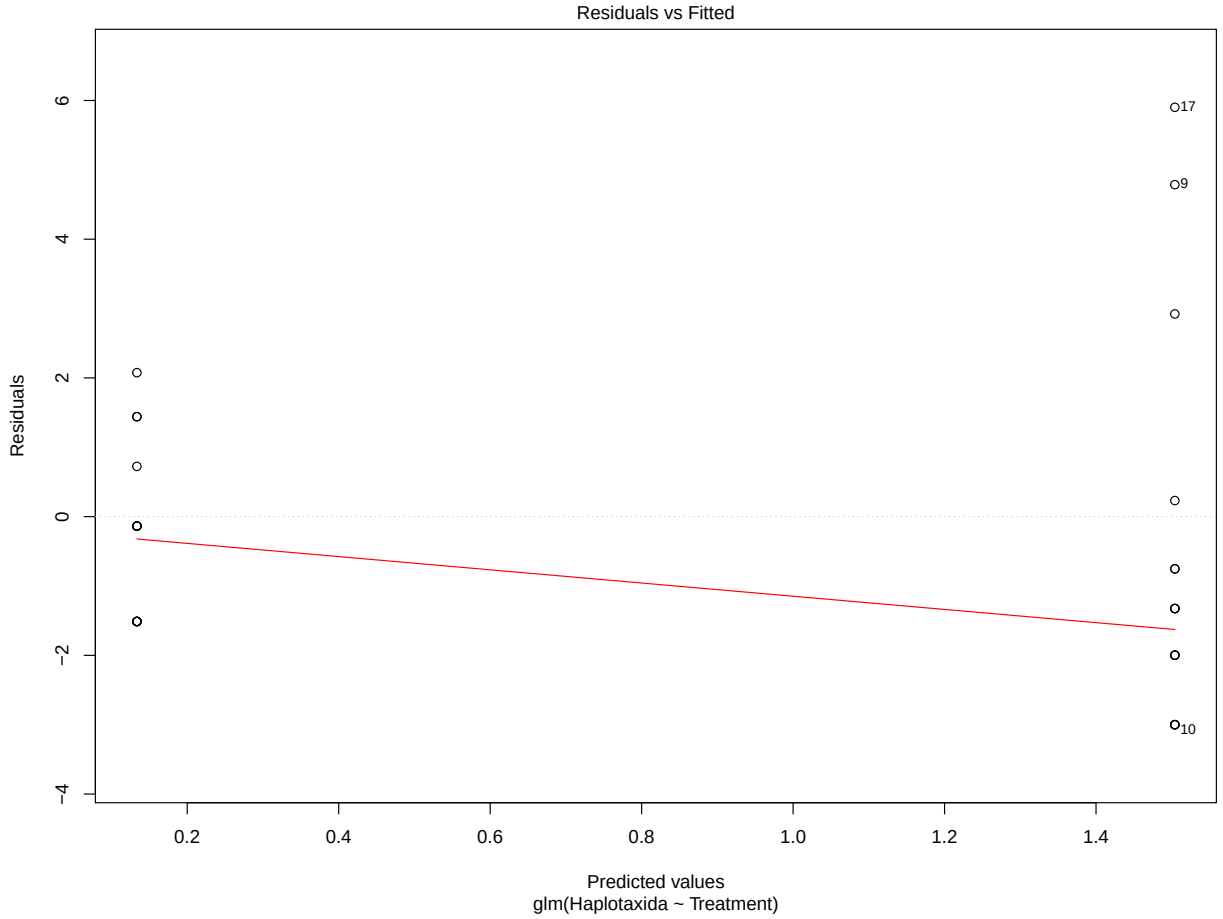
1. The observed y values are **independent**, conditional on x
2. The y values come from **a known distribution** (from the exponential family) with known **mean-variance relationship** $V(\mu)$
3. straight line relationship between **some known function of the mean** of y and each x

$$g(\mu_y) = \beta_0 + \mathbf{x}^T\boldsymbol{\beta}$$

The function $g(\cdot)$ is known as **the link function**.

How do you check assumptions?

Don't just look at numerical measures (residual deviance, AIC, *etc*) – plot your data! As with linear models, use residual plots to checking for no pattern. Try `plot(data_revS)`:



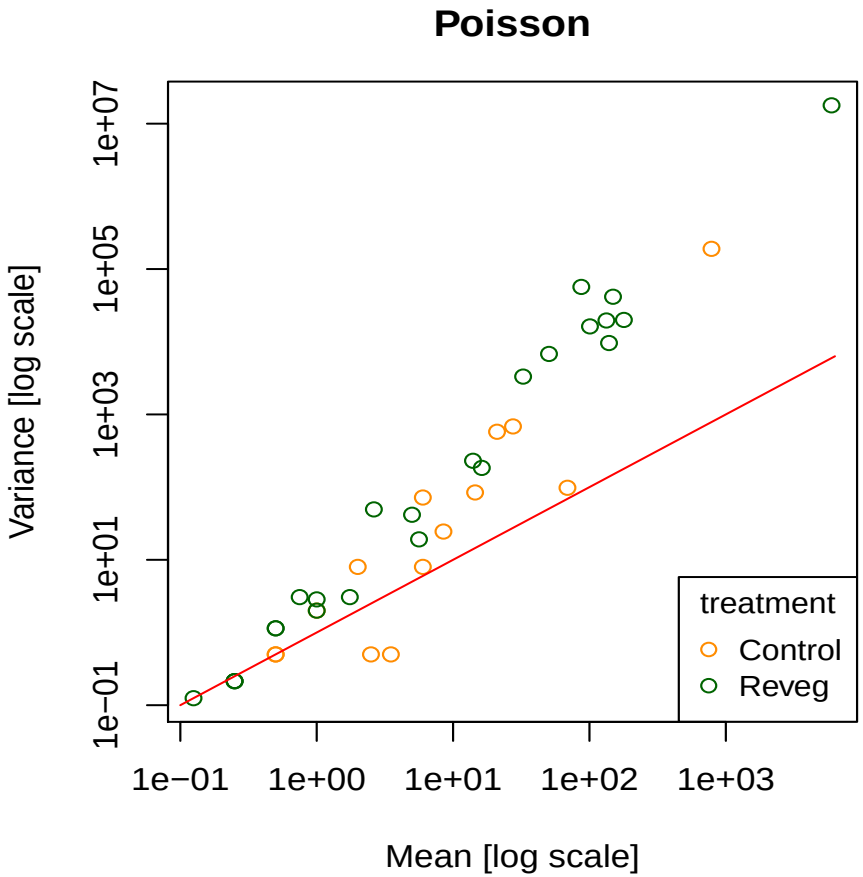
Interpret residual plots

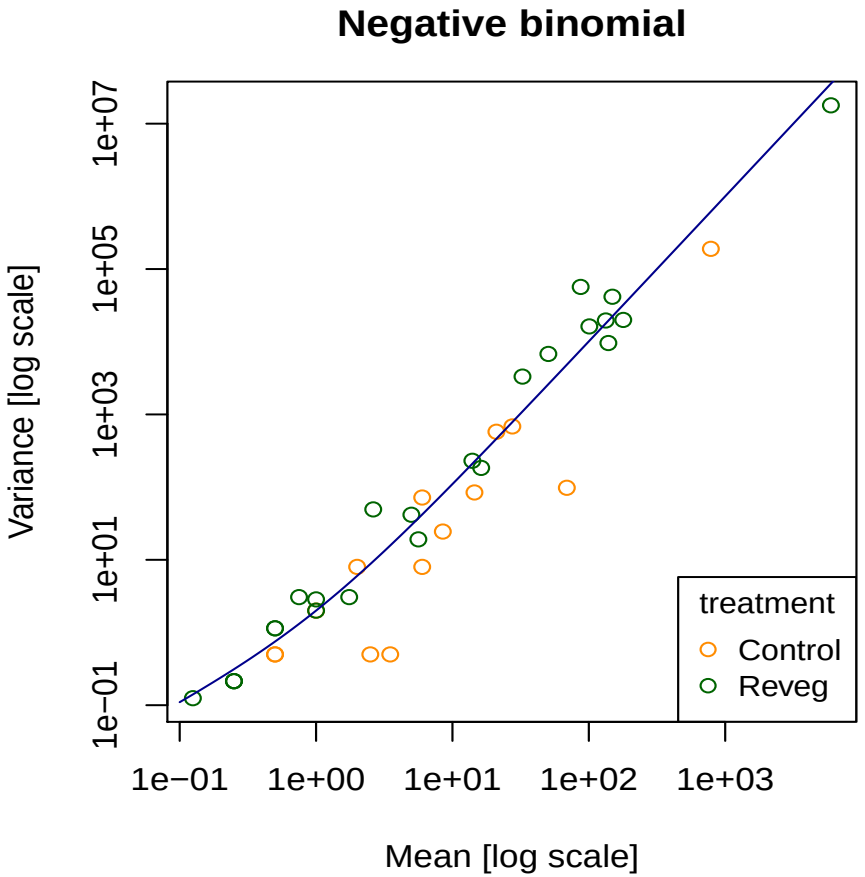
- U-shape $\Rightarrow$ violation of linearity assumption
- Fan-shape $\Rightarrow$ violation of mean-variance assumption

Counts that don't fit a Poisson distribution - overdispersion

Poisson: $V(\mu) = \mu$

Negative Binomial: $V(\mu) = \mu + \phi\mu^2$





Counts that don't fit a Poisson distribution - overdispersion

At larger values of the mean, data are more variable than expected and we have what is called "overdispersion". This is because the Poisson mean variance assumption ($V(\mu) = \mu$) can be a bit restrictive.

Instead we will use the negative binomial distribution with mean variance assumption ($V(\mu) = \mu + \phi\mu^2$, with "overdispersion parameter" ϕ).

Negative Binomial Regression

Negative binomial regression can be fitted using the `glm.nb` function in the `MASS` package or the `manyglm` function in the `mvabund` package:

```
> library(mvabund)
> hapnb_many = manyglm(Haplotaxida~treatment, family="negative.binomial",
+ data=data_revS)
> plot(hapnb_many)
```

or

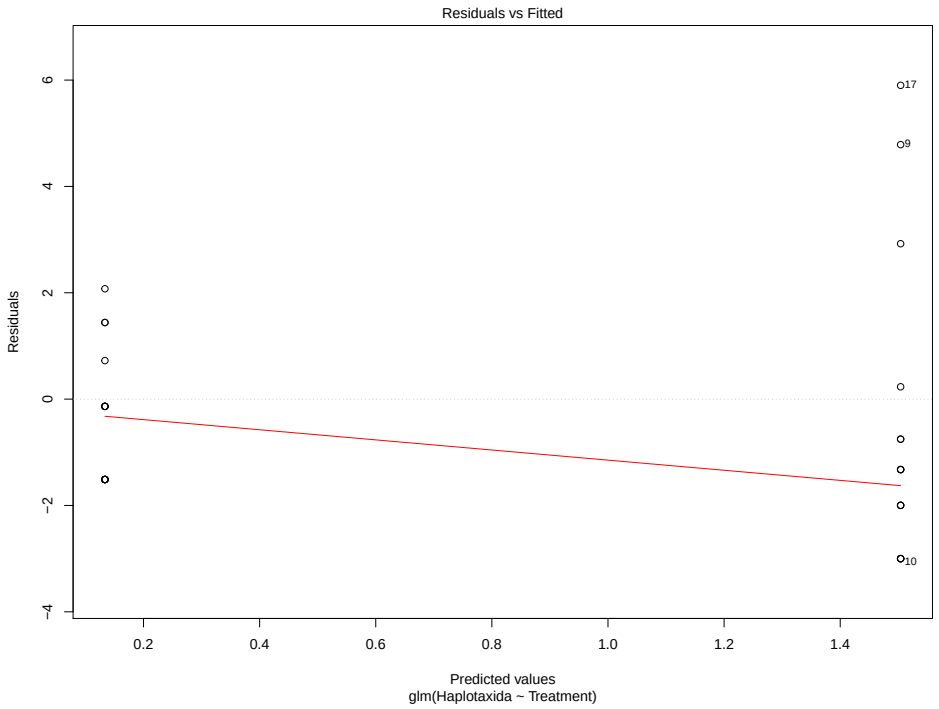
```
> library(MASS)
> hapnb_glm = glm.nb(Haplotaxida~treatment, data=data_revS)
> plot(hapnb_many)
```

Note: `plot.manyglm` uses Dunn-Smyth residuals which 'jitter' around points, which is particularly useful for Binomial 0/1 GLMs, where assumptions are harder to check.

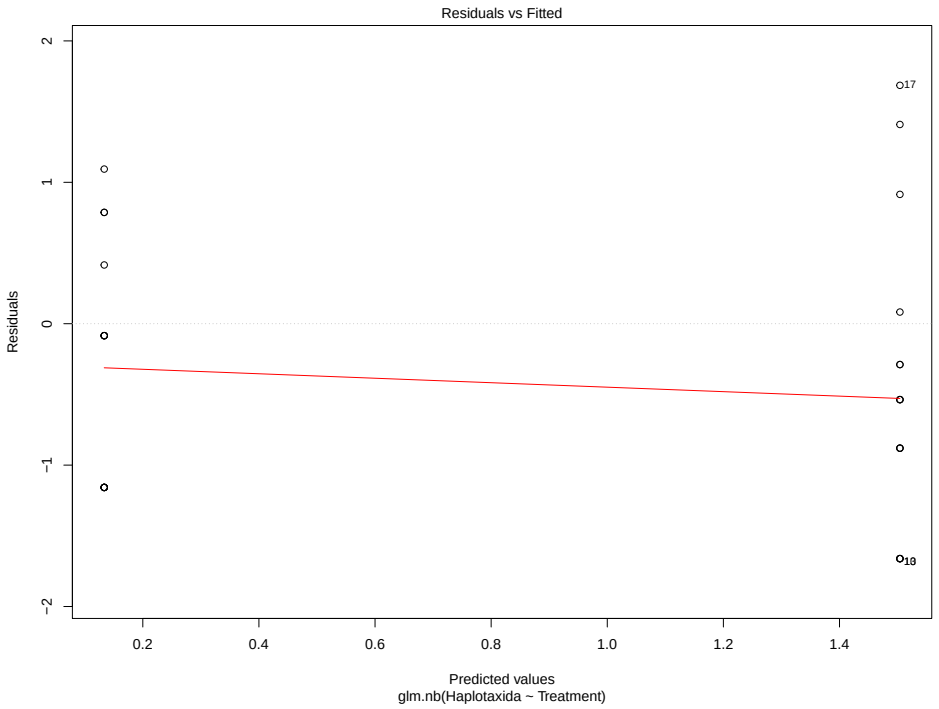
Negative Binomial Regression

Is the negative binomial distribution producing a better model?

Poisson:



Negative Binomial:



Negative Binomial Regression

Using the `glm.nb` function:

```
> data_revS = read.csv("data/revegSmall.csv")
> data_revS$Treatment <- as.factor(data_revS$Treatment)
> library(MASS)
> hap_nb <- glm.nb(Haplotaxida~Treatment,data=data_revS)
> anova(hap_nb,test="Chisq")
Analysis of Deviance Table

Model: Negative Binomial(0.6794), link: log

Response: Haplotaxida

Terms added sequentially (first to last)


          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         29     37.422
Treatment  1   6.6753        28     30.746 0.009776 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Is there any evidence of an effect of revegetation on worm counts?

Inference from generalised linear models

All the same functions as for linear models work:

Confidence intervals use the `confint` function

Hypothesis testing use the `summary` or `anova` function. The latter is generally better for hypothesis testing.

Model selection similar to linear models (`stepAIC`, `predict` for cross-validation)

Using the `anova` function with GLMs

The term `anova` is a little misleading for GLMs – technically, what we get is an analysis of deviance table, not analysis of variance.

For GLMs we have to tell `anova` what test statistic to use (otherwise it won't use any!). Add the argument `test="Chisq"` – it will do likelihood ratio tests, and compare test statistics to a chi-squared distribution to get P-values.

`summary` vs `anova` for GLMs

The `summary` function on the other hand uses Wald tests – comparing $\widehat{\beta}/\text{se}(\widehat{\beta})$ to a standard normal distribution. This is less accurate, and especially for logistic regression, can give quite different results.

Inference for small samples

The `summary` and `anova` tests are both approximate – they work well in large samples (well, `summary` can be a bit weird) but can be quite approximate when sample size is small.

We can beat this problem exactly by calculating P-values by simulation, specifically by resampling the data.

The simplest way to do this is using the `mvabund` package, which uses resampling by default whenever you call `summary` or `anova` for a `manyglm` object.

Inference for small samples

```
> hapnb_many <-manyglm(Haplotaxida~Treatment,family="negative.binomial",data=dat_revS)
> anova(hapnb_many)
Time elapsed: 0 hr 0 min 0 sec
Analysis of Deviance Table

Model: manyglm(formula = Haplotaxida ~ Treatment, family = "negative.binomial",
Model:     data = data)

Multivariate test:
Res.Df Df.diff   Dev Pr(>Dev)
(Intercept)     29
Treatment       28       1 5.942    0.073 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Arguments: P-value calculated using 999 resampling iterations via PIT-trap resampling.
```

Is there any evidence of an effect of revegetation on worm counts?

Offsets

Sometimes there is some variable known not just to be important to the response, but its precise relationship with response is also known. This most commonly happens with sampling intensity (sample twice as hard you expect to get twice as many observations).

e.g. Anthony actually sampled five pitfall traps in nine sites, but only four pitfall traps in a tenth site, as follows:

Treatment	C	R	R	R	C	R	R	R	R	R
Count	0	3	1	3	1	2	12	1	18	0
# pitfalls	5	5	5	5	5	5	5	4	5	5

How can we account for the different sampling effort at different sites in our model?

Offsets

Don't just rescale or average values. This changes the distribution of your data, and stuffs up the mean-variance relationship.

Instead, we include an **offset** – a predictor variable known to be exactly proportional to the response. Because we model $\log(\mu)$ in Poisson and negative binomial regression, our offset is log(# pitfalls).

This is most easily done by adding an offset of `log(pitfalls)` to your model formula using:

```
offset(log(pitfalls))
```

Offsets

```
> hapnb_many <-manyglm(Haplotaxida~Treatment+offset(log(pitfalls)),
+ family="negative.binomial",data=dat_revS)
> anova(hapnb_many)
Time elapsed: 0 hr 0 min 0 sec
Analysis of Deviance Table

Model: manyglm(formula = Haplotaxida ~ Treatment + offset(log(pitfalls)),
Model:     family = "negative.binomial", data = data)

Multivariate test:
Res.Df Df.diff   Dev Pr(>Dev)
(Intercept)     29
Treatment       28       1 6.019    0.065 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Arguments: P-value calculated using 999 resampling iterations via PIT-trap resampling.
```

Example: Glow in the dark Guinea Pigs

Professor Puckeridge is interested in comparing the effectiveness of two new drugs that make pets glow in the dark. He is trialing the drugs on 15 families of Guinea pigs and has counted how many siblings in each family glow in the dark at the end of his 10 week trial period. How can Professor Puckeridge analyse his data?

Drug	A	A	A	A	A	A	A	B	B	B	B	B	B	B	B
Glow	7	3	5	3	3	3	6	5	5	5	4	2	3	5	12
Family size	8	9	8	7	7	3	8	5	7	7	4	5	4	6	14

(A=Lumosium Drug, B=Glowethradiatum Drug)

Is there any evidence that one drug is more effective at making guinea pigs glow in the dark?

Binomial Count regression

"x-out-of-n" counts across n independent events.

Binomial Distribution The probability of observing $X = k$ successes in n independent trials where the probability of observing a success is $p \in [0, 1]$ is the following:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

We say X follows a binomial $B(n, p)$ distribution.

Binomial Count regression: Coding it

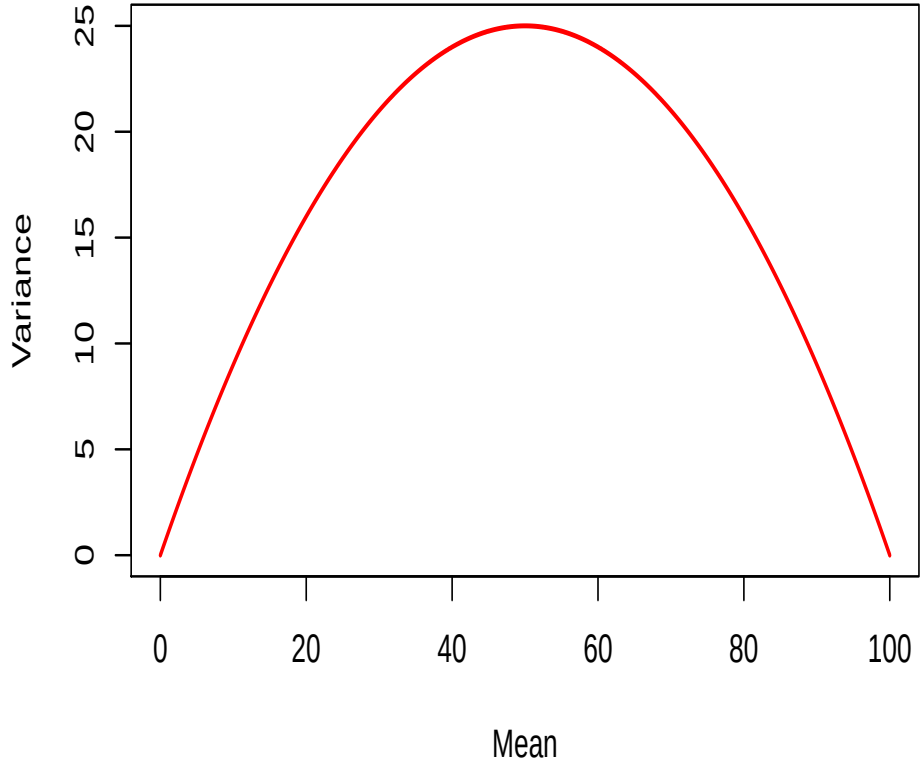
We use the `glm` function with the response as proportion of "successes" and `weights` equal to the total number in each trial.

```
> fit_glow <- glm(Glow/Total~Drug,family=binomial,
+ data = Glow_Dat,weights=Total)
> anova(fit_glow,test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Glow/Total

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                    14     24.227
Drug  1   4.3228        13     19.904  0.03761 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Is there evidence that one drug is more effective at making guinea pigs glow in the dark?

Binomial Count regression: Caution!

Remember that the Binomial mean-variance relationship is **exactly** quadratic ($V(\mu) = n\mu(1-\mu)$), which similar to Poisson regression, can often be restrictive!



Binomial Count regression: Solution?

This can be understood as airising because of missing predictors across clusters.

Solution: Add observation level random effects that help account for potential missing predictors, and relaxes the binomial mean-variance assumption.

Can use the function `glmer` in package `lme4` where `(1|ID)` represents a random intercept for each level of `ID`.

Works similar to the overdispersion parameter ϕ in negative binomial regression!

Binomial Count regression: Coding it

We use the `glmer` function with a random intercept `(1|familyID)` for each family of guinea pigs. When using anova in glmer we need to also create a "null" model without the predictor of interest.

```
> library(lme4)
> Glow_Dat$familyID <- as.factor(c(1:15))
> fit_glow_id <- glmer(Glow/Total~Drug+(1|familyID),family=binomial,
+ data = Glow_Dat,weights=Total)
> fit_glow_idnull <- glmer(Glow/Total~(1|familyID),family=binomial,
+ data = Glow_Dat,weights=Total)
> anova(fit_glow_idnull,fit_glow_id)
Data: Glow_Dat
Models:
fit_glow_idnull: Glow/Total ~ (1 | familyID)
fit_glow_id: Glow/Total ~ Drug + (1 | familyID)
Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
fit_glow_idnull  2 53.792 55.208 -24.896   49.792
fit_glow_id      3 52.717 54.841 -23.358   46.717 3.0753      1    0.07949 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Is there still evidence that one drug is more effective at making guinea pigs glow in the dark?

Binomial Count regression: Checking

How do we check if we need observation level random effects?

- Check residual vs. fitted plots and check for horizontal ellipse shape.
- Compare information criteria (AIC, BIC etc.)

Extensions of GLMs

There are a few important additional features and extensions of GLMs worth knowing about.

- Zero-inflated models
- Generalised additive models
- Generalised linear mixed models
- Multivariate abundance models

Zero-inflated models

Ecological counts often have many zeros *e.g.* consider Anthony's revegetation counts:

Treatment	C	R	R	R	C	R	R	R	R	R
Count	3	0	0	0	4	1	0	0	0	0
# pitfalls	5	5	5	5	5	5	5	4	5	5

It is tempting to use a **zero-inflated** model (Welsh et al. 1996) – a model which expects more zeros than the Poisson.

For details on how to fit such models, see the `VGAM` and `pscl` packages.

However – just because you have lots of zeros doesn't mean that your data are necessarily zero-inflated. (*e.g.* a Poisson distribution with $\mu = 0.1$ already expects 90% of values to be zero!)

The above cockroach data are actually very well fitted by a Poisson distribution.

If trying out a zero-inflated model, please **check that you needed it** – maybe your data aren't actually zero-inflated (Warton, 2005)!

Generalised additive models

Sometimes the assumption of straight-line relationship $\mathbf{x}^T\boldsymbol{\beta}$ is too restrictive. One way to extend this is replace it with **smoothers** *e.g.* spline smoothers.

Such models are referred to as Generalised additive models (GAMs). For more details, see Faraway (2005). In `R`, these may be fitted using the `mgcv` package. Note however that the `mgcv` package does not use Dunn-Smyth residuals for residual plots.

N.B. It is important to remember that a "generalised linear model" does **not** need to be linear: by including functions of x as predictors (*e.g.* quadratic, cubic terms, periodic functions), you can use the "straight line relationship" to fit some fairly non-linear functions.

Generalised linear mixed models

Arguably the most important extension to GLMs is the inclusion of **random effects**.

Sometimes you might have random factors (*e.g.* nested design). Generalised linear models (the `glm` and `manyglm` functions) only handle fixed effects.

If you have random effects and a non-constant assumed mean-variance relationship then you want to fit a **generalised linear mixed model**.

A good package for this, as in the linear mixed effects case, is the `lme4` package.

There aren't really any new tricks – just use the `glmer` argument as you would normally use `lmer`, but be sure to add a `family` argument.

Current limitations:

- As before, no nice residual plots (on our to-do list).
- As before, doesn't handle `family=negative.binomial` with unknown overdispersion. But you can try a workaround where you use the Poisson with additional random effects to introduce overdispersion.
- GLMM's can take much longer to fit and even then only give approximate answers. The mathematics of GLMM's are way harder then traditional GLM's.
- There is an optional argument `nAGQ` that you can try for simple models to get a better approximation for GLMMs (*e.g.* `nAGQ=4`).

Multivariate Abundance Models

If we are interested in testing for effects accross a group of measured count variables, we can use a **multivariate abundance model**.

e.g. Counts of different fish species where counted in natural and artificial reefs. Researchers are interested in testing for a difference in the fish community structure between the two reef types.

For more details see the `mvabund` package, there is a great video titled "What does multivariate analysis have in common with Rick Astley" that I would recomend watching:
https://www.youtube.com/watch?v=KnPkH6d89l4

Method was developed here at UNSW, pretty cool!