Generalized Linear Models (GLM)

Silwood Masters Statistics: Lecture Three

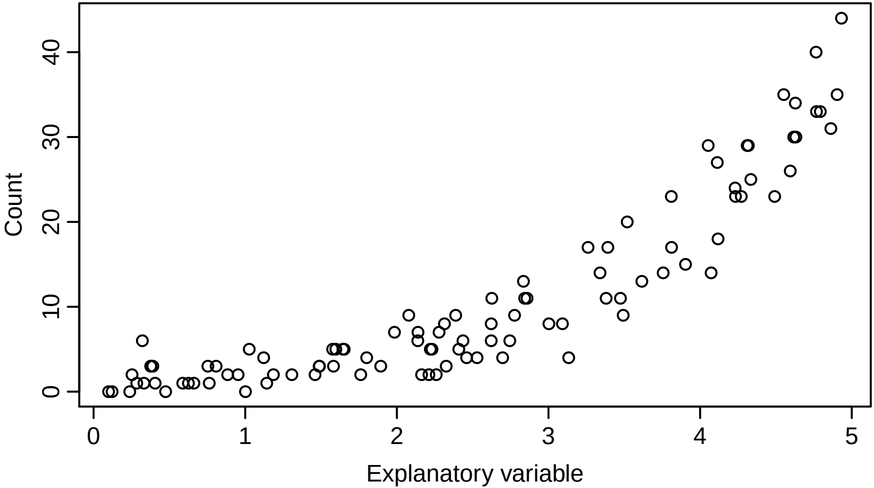
David Orme

Lecture outline

- Why do we need GLMs?
- Two opposing problems
    - Assessing model fit
    - Finding the model parameters
- Reconciling the two problems
- Model criticism

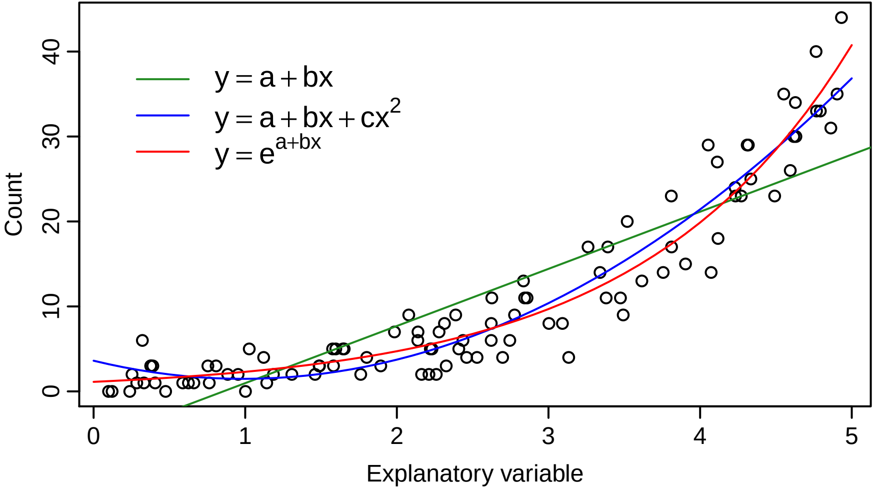
Count data

Using count data as an example:



Linear models[1] on count data

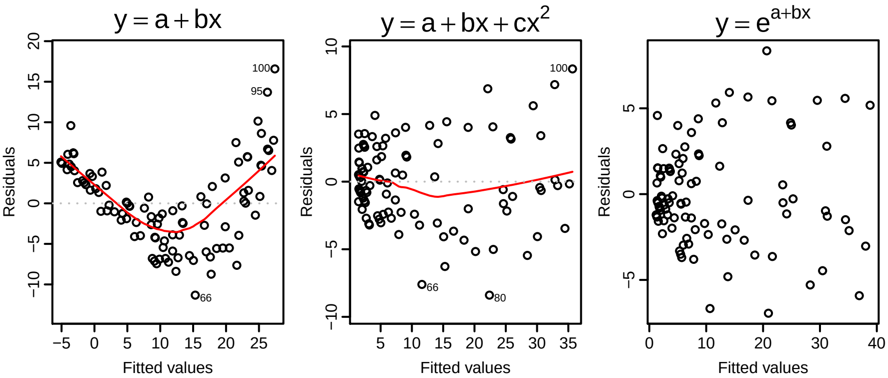
Just try fitting some simple models



[1] Actually, one is non-linear!

Linear models on count data are poor

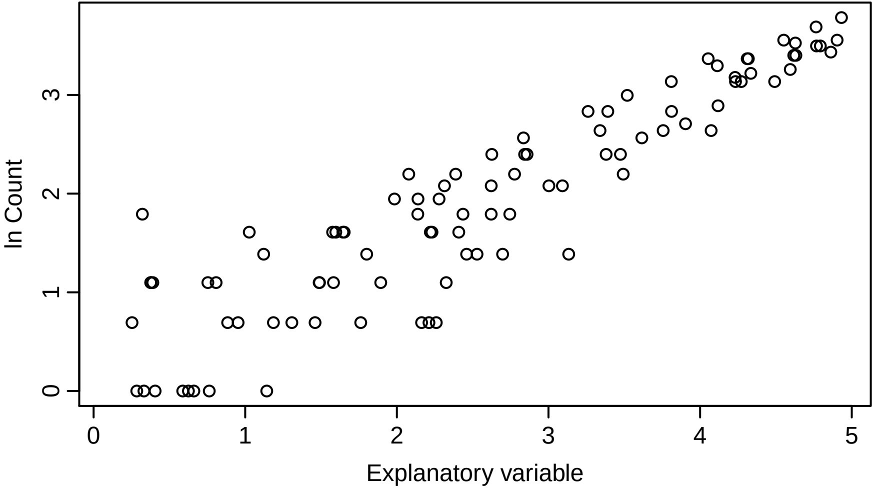
Those go through the data[2] but the residuals have non-constant variance.



[2]Apart from $y = a + bx$!

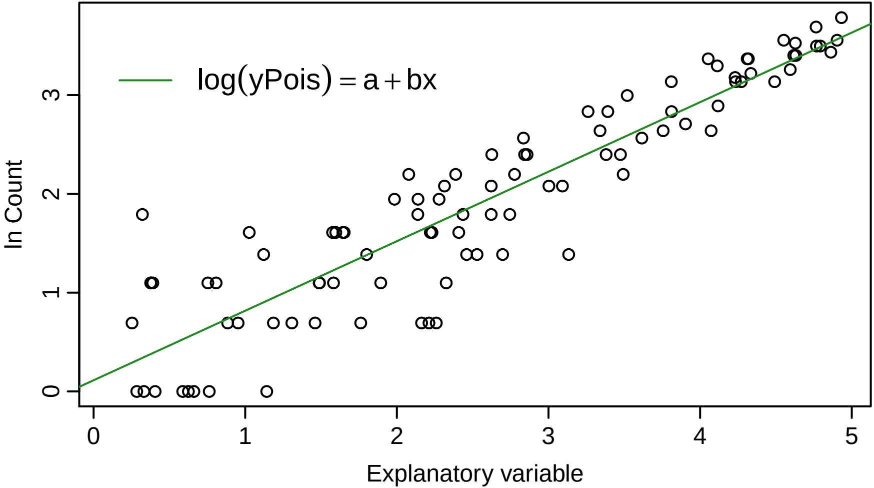
Transforming count data

- Try logging the data
- Have to exclude zero values since $\log(0)$ is undefined.



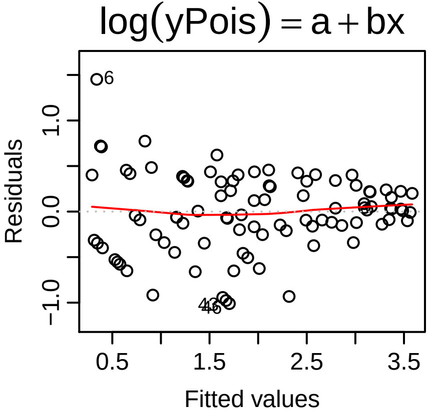
Transforming count data

…and fit a model ($\log(y) = a + bx$)to that.



Transforming count data is poor

- Residuals show decreasing variance
- Zero values are a problem.




```r
lm(log(yPois) ~ x)
## Error in lm.fit(x, y,
offset = offset,
singular.ok = singular.ok,
...): NA/NaN/Inf in 'y'
```

Two opposing problems

For count data, we're finding these two issues with standard linear models

- It is difficult to get well behaved residuals and assess model fit.
- It isn't easy to fit linear models to untransformed data.

We'll look at each of these problems in turn.

Assessing model fit

The underlying problem is that the standard linear model assumes:

$$\mu_i = \mathbf{X}_i \, \beta \qquad\qquad y_i \sim N(\mu_i, \sigma^2)$$

Which is *terrifying*.

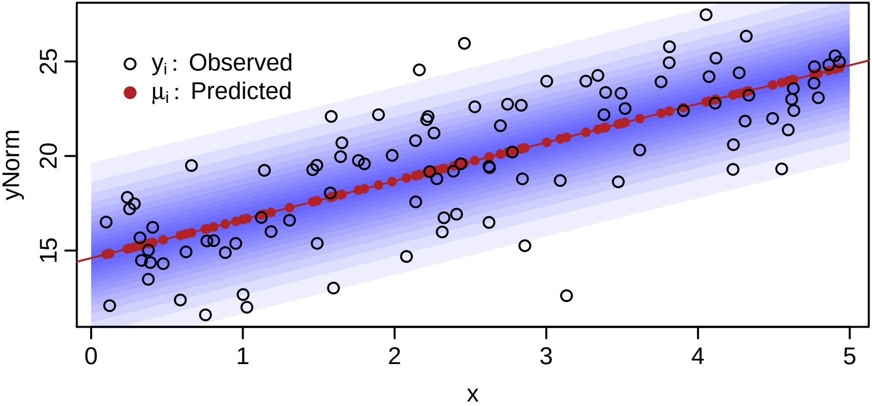
Assessing model fit

The underlying problem is that the standard linear model assumes:

$$\underbrace{\mu_i}_{\text{Predicted values}} = \overbrace{\mathbf{X}_i}^{\text{Design matrix}} \overbrace{\beta}^{\text{Coefficients}} \qquad \underbrace{y_i}_{\text{Observed data}} \sim \underbrace{N(\mu_i, \sigma^2)}_{\text{Normal distribution}}$$

Expected values in a linear model

$y_i \sim N(\mu_i, \sigma^2)$ Observed data (y_i) come from a normal distribution ($N()$) centred around the predicted value (μ_i) with a given constant variance (σ^2).



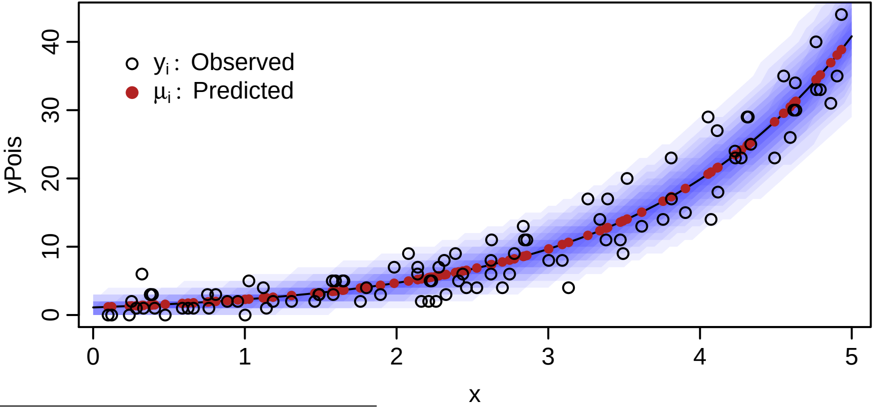
Expected values in a poisson model

But for count data, this doesn't work

- Count data has *Poisson* errors
- Variance (σ^2) increases with the predicted mean (μ)
- In fact, for Poisson data, the variance equals the mean ($\sigma^2 = \mu$).
- Bounded below at zero ($\mu \geq 0$).

Why do we need GLMs?

$y_i \sim Poi(\mu_i, \phi)$ Observed data (y_i) come from a Poisson distribution ($Poi()$) centred around the predicted value (μ_i) with variance (μ_i) and a scale (ϕ)[3].



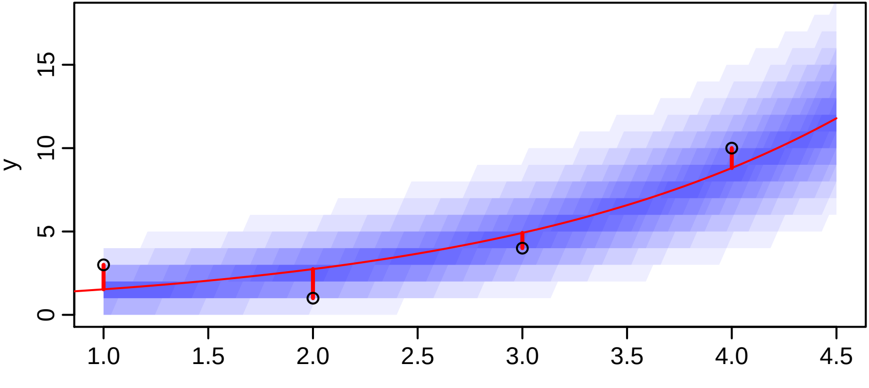
[3] We will come back to this to talk about dispersion

Log likelihood

For linear models, the standard measure of fit is the residual, but:

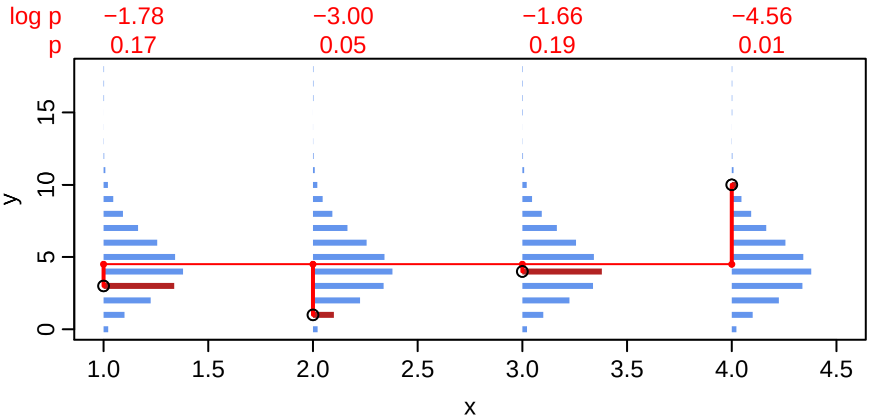
- Variance increases with the mean
- The poisson distribution isn't *symmetrical*
- Residuals vary in importance with position in the model.

In GLMs, we use log likelihood instead.



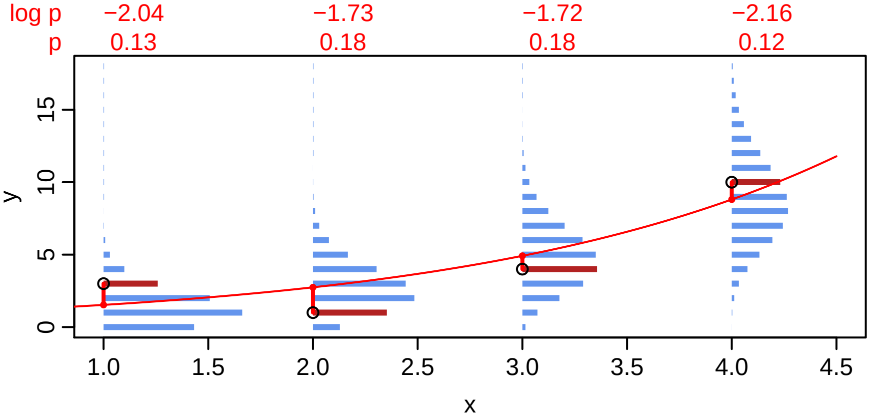
Three models: the null model

- The null model predicts a shared grand mean only
- The log likelihood is -11.001:



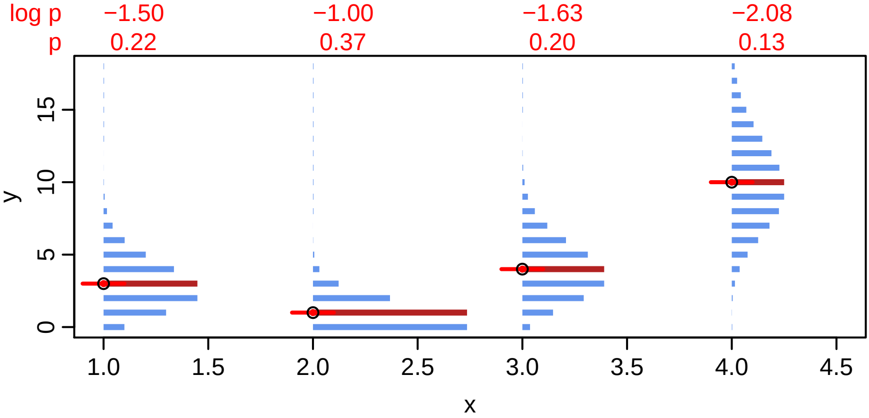
Three models: the candidate model

- The candidate model includes an explanatory variable x
- The log likelihood is -7.659:



Three models: the saturated model

- The saturated model fits each point independently
- The log likelihood is -6.207:



Comparing model fit

Residual sum of squares Decreases towards zero with better models

Log likelihood Increases with better models

Deviance Derived from log likelihood but decreases towards zero with better models.

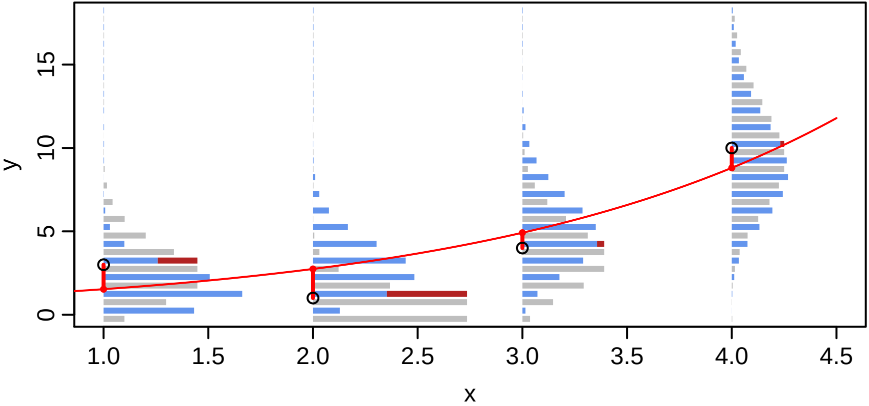
$$\text{Deviance} = D = -2 \times (L_M - L_S)$$

Twice the difference in log likelihood of a given model (L_M) from the saturated model (L_S).

Comparing model fit

Residuals in GLMs

- Use differences in log likelihood from the saturated model to assess the fit of individual points



Residuals in GLMs

- Points with a poor fit have a high contribution to deviance.
- Deviance contributions are always positive.
- Use the sign of the *response residual* ($\text{res}_R = y - \mu$)
- Define a *deviance residual* for each point:

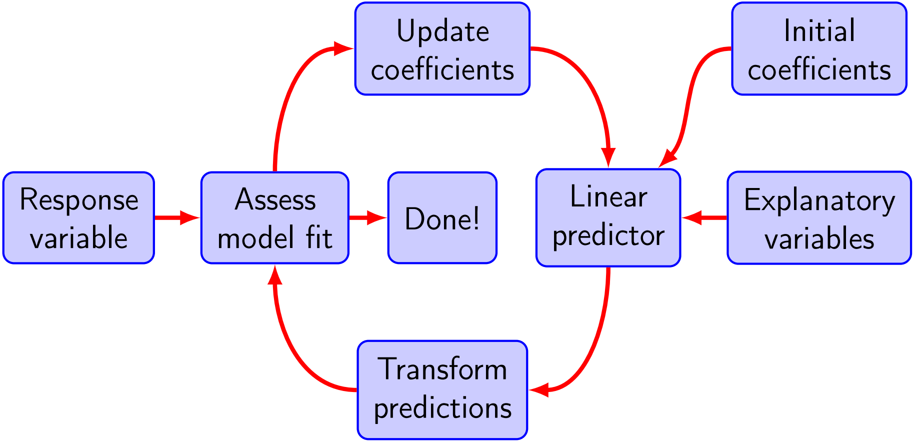
$$res_D = \text{sign}(res_R) \times \sqrt{2 \times (L_M - L_S)}$$

x	y	μ	res_R	L_M	L_S	L_M-L_S	D	res_D
1	3	1.53	1.47	-2.04	-1.50	0.55	1.10	1.05
2	1	2.74	-1.74	-1.73	-1.00	0.73	1.47	-1.21
3	4	4.92	-0.92	-1.72	-1.63	0.09	0.18	-0.43
4	10	8.81	1.19	-2.16	-2.08	0.08	0.15	0.39
				-7.66	-6.21		2.90	

Finding the parameters

- Now we have a process to assess model fit.
- How do we find models to assess?!
- Identify a *link function*:
  - A transformation onto a scale suitable for linear models
  - Has an *inverse link* function to map predictions back to the response data
  - For count data, the log is commonly used
- The model coefficients are described using this transformed prediction: the *linear predictor*
- Use an iterative process to find the maximum likelihood solution

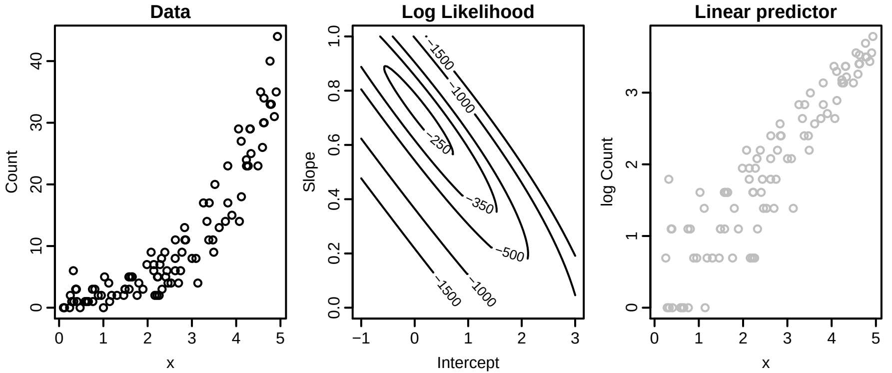
Fitting a GLM



Finding the parameters

Starting from a null model:

- Assess fit to the data and then adjust to improve
- Stop when the coefficients stabilize
- The transformed data (gray) isn't used at all - it is just for show



The GLM model

$$g\left(\mu_i\right) = \mathbf{X}_i\beta \qquad y_i \sim Poi(\mu_i, \phi)$$

Oh no! Not again!

The GLM model

$$\overbrace{g}^{\text{Link function}}\left(\underbrace{\mu_i}_{\text{Predicted values}}\right) = \overbrace{\mathbf{X}_i\beta}^{\text{Linear predictor}} \qquad \underbrace{y_i}_{\text{Observed data}} \sim \overbrace{Poi(\mu_i, \phi)}^{\text{Poisson distribution}}$$

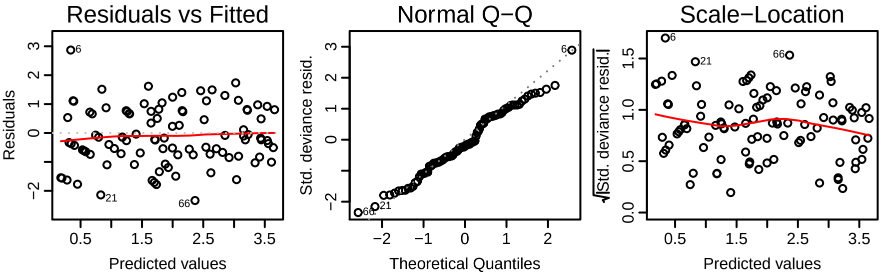
- Describe the model on a scale that facilitates easy model descriptions
- Assess the fit of the model using the original *error structure*

Iteratively Re-weighted Least Squares

- The animation on the earlier slide used an inefficient search algorith to find the maximum likelihood.
- The actual GLM code in R uses an efficient Newton method called *iteratively weighted least squares* (IRLS).
- If you understand how this works, can you come and explain it to me!

Model criticism

- Is the model a *good* explanation of the data?
- Fortunately, deviance residuals have very similar expectations to response residuals in a linear model.
  - Normal distribution
  - No patterns in residuals with respect to predicted values
- Use the same diagnostic plots as for a linear model:



Model criticism

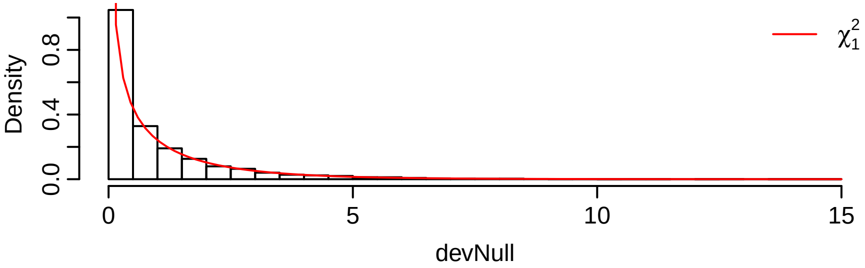
- Is the model an *interesting* explanation of the data?
- Linear models use the change in explained variance relative to change in degrees of freedom
- GLMs use the difference in deviance (D) relative to change in degrees of freedom

```
model <- glm(yPois ~ x, family=poisson(link=log))
null <- glm(yPois ~ 1, family=poisson(link=log))
deviance(model)

## [1] 100

deviance(null)

## [1] 1020

deviance(null) - deviance(model)

## [1] 920
```

Model criticism

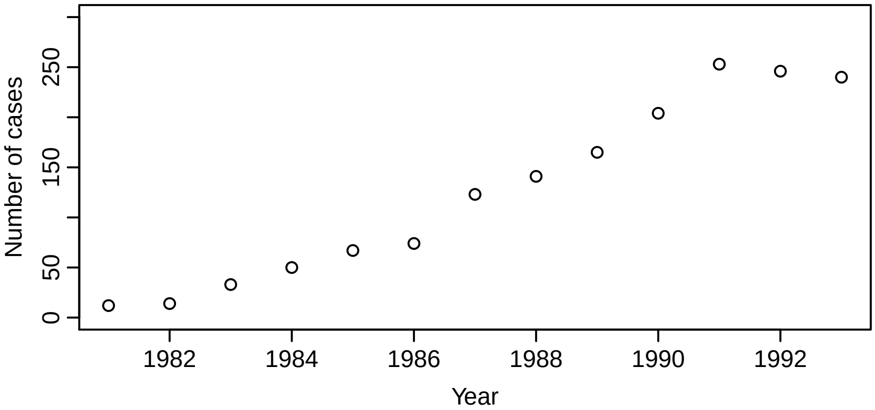
We need to assess whether an observed improvement in model fit is surprising, compared to improvements observed when null model is true.

- With a linear model, we used the F distribution with the change in and residual degrees of freedom (e.g. $F_{1,48}$).
- With a Poisson GLM, we use the χ^2 distribution with the change in degrees of freedom (e.g. χ^2_1).



Model comparison

- Number of AIDS cases in Belgium from 1981 to 2013.
- Models with linear, quadratic and cubic linear predictors.



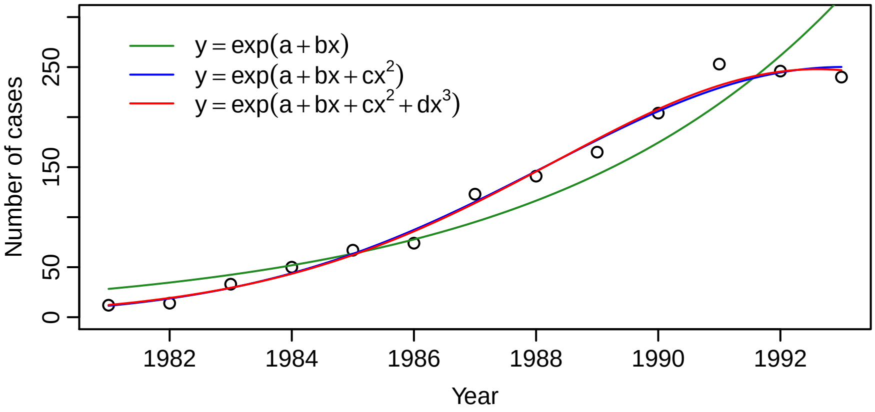
Fitting the models to be compared

- The code to fit a GLM in R is very similar to a linear model.
- Specify an error structure and a link function.

```r
null <- glm(cases ~ 1, data = belg.aids,
    family = poisson(link = log))
am1 <- glm(cases ~ year, data = belg.aids,
    family = poisson(link = log))
am2 <- glm(cases ~ year + I(year^2), data = belg.aids,
    family = poisson(link = log))
am3 <- glm(cases ~ year + I(year^2) + I(year^3),
    data = belg.aids, family = poisson(link = log))
```

Fitting the models to be compared

- The model is described on the scale of the linear predictor
- Showing those models on the original data:



Comparing models: Analysis of deviance

- Just as with Anova in a linear model, does the change in explained deviance merit the extra model complexity?

```
anova(null, am1, am2, am3, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: cases ~ 1
## Model 2: cases ~ year
## Model 3: cases ~ year + I(year^2)
## Model 4: cases ~ year + I(year^2) + I(year^3)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        12        872
## 2        11         81  1      792   <2e-16 ***
## 3        10          9  1       71   <2e-16 ***
## 4         9          9  1        0     0.63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

GLMs and AIC

- We can also compare models using AIC, does the change in AIC merit the extra model complexity?

```
AIC(null, am1, am2, am3)
```

```
##      df   AIC
## null  1 955.9
## am1   2 166.4
## am2   3  96.9
## am3   4  98.7
```

Exploring models: analysis of deviance

- So it looks like the quadratic model is best.
- Just like a linear model, we can use deviance to look at the significance of model terms

```
anova(am2, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cases
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                         12        872
## year       1      792        11         81   <2e-16
## I(year^2)  1       71        10          9   <2e-16
```

Exploring models: linear model summary

- And we can use linear models summaries to look at the significance of coefficients
- Importantly, model coefficients are estimated and reported *on the scale of the linear predictor*

```
summary(am2)
...
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.48e+04   1.05e+04   -8.07  7.3e-16
## year         8.51e+01   1.06e+01    8.05  8.4e-16
## I(year^2)   -2.14e-02   2.66e-03   -8.03  9.8e-16
##
...
```