Harvard Department of Government 2003
Faraway Chapter 8, Generalized Linear Models

JEFF GILL

Visiting Professor, Fall 2024

Generalized Linear Models

- Definition: A **response** is modeled by a linear combination of the **predictors** related through a **link function** (McCullagh and Nelder 1989).

- The response variable must be a member of the exponentially family of distributions:

$$f(y|\theta, \phi) = \exp\left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right].$$

- The θ parameter is called the **canonical parameter** and is the measure of location.

- The ϕ parameter is called the **dispersion parameter** and is the measure of scale.

- Notice the use of "a", "b", and "c."

Exponential Family Form

- The exponential family form of a PDF or PMF is:

$$\begin{aligned} f(z|\zeta) &= \exp\big[t(z)u(\zeta)\big]r(z)s(\zeta) \\ &= \exp\big[t(z)u(\zeta) + \log r(z) + \log s(\zeta)\big], \end{aligned}$$

  where: r and t are real-valued functions of z that do not depend on ζ, and s and u are real-valued functions of ζ that do not depend on z, and $r(z) > 0,\ s(\zeta) > 0\ \forall z, \zeta$.

Exponential Family Form

- The canonical form obtained by transforming: $y = t(z)$, and $\theta = u(\zeta)$. Call θ the canonical parameter. This produces the final form:

$$f(y|\theta) = \exp\big[y\theta - b(\theta) + c(y)\big].$$

- The exponential family form is invariant to sampling:

$$f(\mathbf{y}|\theta) = \exp\left[\sum y_i\theta - nb(\theta) + \sum c(y_i)\right].$$

- And there often exists a *scale parameter*:

$$f(\mathbf{y}|\theta) = \exp\left[\frac{\sum y_i\theta - nb(\theta)}{\phi} + \sum c(y_i, \phi)\right].$$

The Normal Distribution

- Start with the standard expression:

$$f(y|\theta,\phi) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right]$$

  and rearrange to:

$$f(y|\theta,\phi) = \exp\left[\frac{y\mu - \mu^2/2}{\sigma^2} - \frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log(2\pi\sigma^2)\right)\right].$$

- To put this in the canonical form above, re-express as:
  - ▻ $\theta = \mu$
  - ▻ $\phi = \sigma^2$
  - ▻ $a(\phi) = \phi$
  - ▻ $b(\theta) = \theta^2/2$
  - ▻ $c(y,\phi) = -(y^2/\phi + \log(2\pi\phi))/2.$

The Poisson Distribution

- Start with the standard expression:

  $$f(y|\theta,\phi) = \frac{e^{-\mu}\mu^{y}}{y!}$$

  and rearrange to:

  $$f(y|\theta,\phi) = \exp(y\log\mu - \mu - \log y!).$$

- To put this in the canonical form above, re-express as:
  - $\theta = \log(\mu)$
  - $\phi \equiv 1$ (note the assumption here)
  - $a(\phi) = 1$
  - $b(\theta) = \mu = \exp(\theta)$
  - $c(y,\phi) = -\log y!$.

The Binomial Distribution

- Start with the standard expression:

$$f(y|\theta,\phi) = \binom{n}{y}\mu^y(1-\mu)^{n-y}$$

  and rearrange (Faraway is missing a close parenthesis):

$$\begin{aligned} f(y|\theta,\phi) &= \exp\left[y\log\mu + (n-y)\log(1-\mu) + \log\binom{n}{y}\right] \\ &= \exp\left[y\log\frac{\mu}{1-\mu} + n\log(1-\mu) + \log\binom{n}{y}\right] \end{aligned}$$

- To put this in the canonical form above, re-express as:
  - $\theta = \log\frac{\mu}{1-\mu}$
  - $\phi \equiv 1$ (note the assumption here)
  - $a(\phi) = 1$
  - $b(\theta) = -n\log(1-\mu) = n\log(1+\exp\theta)$
  - $c(y,\phi) = \log\binom{n}{y}$.

Exponential Family Form, Negative Binomial Example

- If Y is distributed negative binomial with success probability p and a goal of r successes, then the PMF in exponential family form is produced by:

$$\begin{aligned} f(y|r,p) &= \binom{r+y-1}{y} p^r (1-p)^y \\ &= \exp\left[\underbrace{y\log(1-p)}_{y\theta} + \underbrace{r\log(p)}_{b(\theta)} + \underbrace{\log\binom{r+y-1}{y}}_{c(y)} \right]. \end{aligned}$$

- The canonical link is easily identified as $\theta = \log(1-p)$.
- Substituting this into $b(\theta)$ and applying some algebra gives $b(\theta) = r\log(1-\exp(\theta))$.

Exponential Family Form, Normal Example (cont.)

- So far μ is the parameter of interest and σ^2 is the nuisance parameter, but we might want to look at the opposite situation.

- In this treatment, μ is not considered a scale parameter. Treating σ^2 as the variable of interest produces:

$$\begin{aligned} f(y|\mu,\sigma^2) &= \exp\left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y^2 - 2y\mu - \mu^2)\right] \\ &= \exp\left[\underbrace{\frac{1}{\sigma^2}}_{\theta}\underbrace{\left(y\mu - \frac{1}{2}y^2\right)}_{z} + \underbrace{\frac{-1}{2}\left(\log(2\pi\sigma^2) - \frac{\mu^2}{\sigma^2}\right)}_{b(\theta)}\right]. \end{aligned}$$

- Now the canonical link is $\theta = \frac{1}{\sigma^2}$. So $\sigma^2 = \theta^{-1}$, and we can calculate the new $b(\theta)$:

$$b(\theta) = -\frac{1}{2}\left(\log(2\pi\sigma^2) - \frac{\mu^2}{\sigma^2}\right) = -\frac{1}{2}\log(2\pi) + \frac{1}{2}\log(\theta) - \mu^2\theta.$$

Moments of the Exponential Family Form

- Mean and Variance:

$$EY = \mu = b'(\theta) \qquad\qquad var Y = b''(\theta)a(\phi)$$

- The mean is a function of θ only while the variance is a product of the location and the scale.
- The term $b''(\theta)$ is called the *variance function* and tells us how the variance relates to the mean.
- For the normal,

$$b''(\theta) = \frac{\partial^2}{\partial\theta^2}b(\theta) = \frac{\partial^2}{\partial\theta^2}\theta^2/2 = \frac{\partial}{\partial\theta}\theta = 1$$

  meaning that the variance is independent of the mean (a special circumstance).

- Weighting of cases done with $a(\phi) = \phi/w_i$, where w_i is a known weight.

Link Function

- Start with the *linear predictor* (also called the systematic component):

$$\eta = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = \mathbf{x}'\boldsymbol{\beta}.$$

- The link function, g, describes how the mean response, $EY = \mu$, is linked to the covariates through the linear predictor:

$$\eta = g(\mu).$$

- For GLMs, a *single index model* is assumed known and operating on all data, produced from the *canonical link* such that $\eta = g(\mu) = \theta$, meaning that $g(b'(\theta)) = g(EY) = g(\mu) = \theta$.

- Using a canonical link means that $\mathbf{X}'\mathbf{Y}$ is *sufficient* for $\boldsymbol{\beta}$.

Why is this handy?

- Consider the score function in this notation:

$$\dot{\ell}(\theta|\phi, y) = \frac{y - \frac{\partial}{\partial\theta}b(\theta)}{\phi}$$

- Which actually has n data values:

$$\dot{\ell}(\theta|\phi, \mathbf{y}) = \frac{\sum t(y_i) - n\frac{\partial}{\partial\theta}b(\theta)}{\phi}$$

- We then set this equal to zero and rearrange to get the *normal equation*:

$$\sum t(y_i) = n\frac{\partial}{\partial\theta}b(\theta)$$

- Returning to the normal case:

$$b(\theta) = \frac{\theta^2}{2}, \text{ and } \; t(y) = y, \text{ so } \; \hat{\theta} = \frac{1}{n}\sum y_i.$$

Expected Value Calculation with $b(\theta)$

- Take the expected value of the difference calculation:

$$E_Y\left[\frac{y - \frac{\partial}{\partial\theta}b(\theta)}{\phi}\right] = 0$$

$$\int_Y \frac{y - \frac{\partial}{\partial\theta}b(\theta)}{\phi} f(y)dy = 0$$

$$\int_Y yf(y)dy - \int_Y \frac{\partial}{\partial\theta}b(\theta)f(y)dy = 0$$

$$\int_Y yf(y)dy - \frac{\partial}{\partial\theta}b(\theta)\int_Y f(y)dy = 0$$

$$E[Y] = \frac{\partial}{\partial\theta}b(\theta)$$

Generalized Linear Model Theory

- Start with the standard linear model meeting the Gauss-Markov conditions:

$$\underset{(n\times 1)}{\mathbf{V}} = \underset{(n\times p)(p\times 1)}{\boldsymbol{X}\boldsymbol{\beta}} + \underset{(n\times 1)}{\boldsymbol{\epsilon}}$$

$$\underset{(n\times 1)}{E[\mathbf{V}]} = \underset{(n\times 1)}{\boldsymbol{\theta}} = \underset{(n\times p)(p\times 1)}{\boldsymbol{X}\boldsymbol{\beta}}$$

- Generalize slightly with a new "linear predictor" based on the mean of the outcome variable:

$$\underset{(n\times 1)}{g(\boldsymbol{\mu})} = \underset{(n\times 1)}{\boldsymbol{\theta}} = \underset{(n\times p)(p\times 1)}{\boldsymbol{X}\boldsymbol{\beta}}$$

Generalized Linear Model Theory, 4 Components

I. Stochastic Component: $\mathbf{Y}$ is the random or stochastic component which remains distributed i.i.d. according to a specific exponential family distribution with mean $\boldsymbol{\mu}$.

II. Systematic Component: $\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{\beta}$ is the systematic component with an associated Gauss-Markov normal basis.

III. Link Function: the stochastic component and the systematic component are linked by a function of $\boldsymbol{\theta}$ which is *exactly the canonical link function*, summarized in the Table below. We can think of $g(\boldsymbol{\mu})$ as "tricking" the linear model into thinking that it is still acting upon normally distributed outcome variables.

IV. Residuals: Although the residuals can be expressed in the same manner as in the standard linear model, observed outcome variable value minus predicted outcome variable value, a more useful quantity is the deviance residual described in detail below.

Link Function Summary

Family	Link	Variance Function
Normal	$\eta = \mu$	1
Poisson	$\eta = \log \mu$	μ
Binomial	$\eta = \log(\mu/(1-\mu))$	$\mu(1-\mu)$
Gamma	$\eta = \mu^{-1}$	μ^2
Inverse Gamma	$\eta = \mu^{-2}$	μ^3

Note: $f_{IG}(y|\mu,\lambda) = (\lambda/2\pi y^3)^{1/2} \exp[-\lambda(y-\mu)^2/2\mu^2 y], \qquad y, \mu, \lambda > 0.$

Fitting GLMs

- The log-likelihood for a single observation with $a_i(\phi) = \phi/w_i$:

$$\log L(\theta_i, \phi; y_i) = w_i \left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)}\right] + c(y_i, \phi).$$

- Usually we cannot get an analytical solution for the parameter vector $\boldsymbol{\beta}$, and have to resort to numerical procedures.

- Procedure: Iteratively Reweighted Least Squares (sometimes Iteratively Weighted Least Squares): a Newton-Raphson procedure with Fisher Scoring.

- These are the linear steps referred to in `R` as “Fisher scoring iterations” at the bottom of summary output:

```
Number of Fisher Scoring iterations: 8
```

GLM Model Assessment

- Recall, comparisons are made between:
  - the *null model*, a common mean μ for all y meaning $y = g^{-1}(\mu + \epsilon)$ (data is modeled as all random variation).
  - the *saturated* or *full* model, where the data are explained exactly but no data reduction or underlying trend information is obtained. This is typically n parameters for n datapoints (data is modeled as all systematic).
  - the proposed model where we have partitioned the data into systematic structures *and* a random component according to some theoretical consideration.
- The log-likelihood for the full model versus the research model can be compared in ratio terms:

$$2(\ell(y, \phi|y) - \ell(\hat{\mu}, \phi|y))$$

- Assuming iid data and $a(\phi_i) = \phi/w_i$, this becomes:

$$\sum_i 2w_i(y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i))/\phi$$

- Call $D(y, \hat{\mu})$ the *deviance*, and the above forms the scaled deviance ($D(y, \hat{\mu})/\phi$).

Deviance Summary

GLM	Unscaled Deviance, $D(y, \hat{\mu})$
Gaussian	$\sum_i (y_i - \hat{\mu}_i)^2$
Poisson	$2\sum_i \left[y_i \log(y_i/\hat{\mu}_i) + (y_i - \hat{\mu}_i)\right]$
Binomial	$2\sum_i \left[y_i \log(y_i/\hat{\mu}_i) + (m - \hat{\mu}_i)\log(((m - y_i)/(m - \hat{\mu}_i))\right]$
	where m is the sample size so μ is the count not the proportion here.
Gamma	$2\sum_i \left[-\log(y_i/\hat{\mu}_i) + (y_i - \hat{\mu}_i)/\hat{\mu}_i\right]$
Negative Binomial	$\sum_i (y_i - \hat{\mu}_i)^2/(\hat{\mu}_i^2 y_i)$

GLM Model Assessment

- Two main tools: the deviance and Pearson's statistic:

$$X^2 = \sum_i \mathbf{R}^2_{Pearson} = \sum_i \frac{(\mathbf{y}_i - \hat{\mu}_i)^2}{\text{Var}(\hat{\mu}_i)}$$

  which lead to asymptotic χ^2 tests with the degrees of freedom equal to the difference in the number of parameters.

- Paradigm: We compare two *nested* models where the more parsimonious model is one that puts linear restrictions (usually $\beta_j = 0$) on the parameters.

- Intuition: an unrestricted model versus a restricted model.

- The goodness of fit test to the data is just a nested test where the nesting model is the saturated model.

- Two Caveats: if $\phi \neq 1$, more elaborate testing required (estimates of ϕ), and sample sizes need to be large for less-granular responses (huge for dichotomous).

GLM Model Assessment

- Setup: we are comparing a large (possibly saturated) model Ω to a restricted research model of interest ω.

- The difference in the scaled deviances, $D_\omega - D_\Omega$ is asymptotically χ^2 with df the number of restrictions.

- The restricted model will have larger deviances because we are making theoretical statements away from just trending through the data.

- General test: model 1 has p parameters and model 2 has $q > p$ parameters. Then

  $$D_p - D_q \underset{\sim}{asym} \chi^2_{df=q-p}.$$

  If this difference is “small” then the restrictions make sense. If this difference is “large” then they take us far from what the data want to say.

- Our claim is that the $p - q$ parameters all have a coefficients equal to zero (this is the restriction).

GLM Model Assessment

- Parameters determine degrees of freedom:
  - the saturated model has n parameters
  - the research model with p parameters (counting the intercept)
  - the null model has 1 parameter to account for the mean.
- Null model versus saturated model:
  - saturated gives D_Ω, null gives D_ω.
  - degrees of freedom are $\#params(\Omega) - \#params(\omega) = n - 1$
  - large values of $D_\omega - D_\Omega$ support the saturated model:
    `Null deviance:  64.76327 on 4 degrees of freedom`.
- Research model versus saturated model:
  - saturated gives D_Ω, research gives D_ω.
  - degrees of freedom are $\#params(\Omega) - \#params(\omega) = n - p$
  - large values of $D_\omega - D_\Omega$ support the saturated model:
    `Residual deviance:  0.37875 on 3 degrees of freedom`.

GLM Model Assessment

- What if we don't have $\phi = 1$?
  - Use an estimate:

$$\hat{\phi} = X^2/(n-p) = \sum_i \frac{(\mathbf{y}_i - \hat{\mu}_i)^2}{\text{Var}(\hat{\mu}_i)}(n-p)^{-1}$$

  - The statistic:

$$f = \frac{(D_\omega - D_\Omega)/(\#params(\Omega) - \#params(\omega))}{\hat{\phi}}$$

    is approximately F distributed with degrees of freedom $df_1 = \#params(\Omega)$ and $df_2 = \#params(\omega)$.

  - Exactly F distributed for the Gaussian model (you've already done this).
  - Be careful for small sample sizes which give poorer estimates of ϕ.

Bliss Insect Data

- Load package and data:

```
library(faraway)
data(bliss)
bliss
  dead alive conc
1    2    28    0
2    8    22    1
3   15    15    2
4   23     7    3
5   27     3    4
```

- This is a balanced experiment since each concentration level is applied to 30 insects.

Bliss Insect Data Model

- Run the simple model:

```
modl <- glm(cbind(dead,alive) ~ conc, family=binomial, bliss)
summary(modl)

Deviance Residuals:
      1        2        3        4        5
-0.4510   0.3597   0.0000   0.0643  -0.2045

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.324      0.418   -5.56  2.7e-08
conc           1.162      0.181    6.40  1.5e-10

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 64.76327  on 4  degrees of freedom
Residual deviance:  0.37875  on 3  degrees of freedom
AIC: 20.85
```

Bliss Insect Data Analysis

- Do a deviance test of this model, where large values indicate a deviance in excess of what we would accept for supporting the research model under the χ^2 test:

```
df.residual(modl)
[1] 3
deviance(modl)
[1] 0.37875
1-pchisq(deviance(modl),df.residual(modl))
[1] 0.9446
```

- The null model is quite poor: `Null deviance:  64.76327 on 4 degrees of freedom`.
- We can *directly* compare our model to the null model:

```
anova(modl,test="Chi")
     Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                     4        64.8
conc  1     64.4         3         0.4    1e-15
```

- Notice that β_1 significant at any reasonably level.

Bliss Insect Data

- Now trying adding a quadratic term to see if a polynomial model is appropriate:

```
modl2 <- glm(cbind(dead,alive) ~ conc+I(conc^2), family=binomial,bliss)
summary(modl2)
Deviance Residuals:
      1         2         3         4         5
-0.1997   0.3241  -0.2185   0.0126   0.0513

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.4959     0.5987   -4.17  3.1e-05
conc          1.4102     0.6170    2.29    0.022
I(conc^2)    -0.0612     0.1432   -0.43    0.669

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 64.76327  on 4  degrees of freedom
Residual deviance:  0.19549  on 2  degrees of freedom
AIC: 22.67
```

Bliss Insect Data

- There are two identical ways to test this new model with slightly different output:

```
anova(modl,modl2,test="Chi")
Analysis of Deviance Table
Model 1: cbind(dead, alive) ~ conc
Model 2: cbind(dead, alive) ~ conc + I(conc^2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         3      0.379
2         2      0.195  1    0.183     0.67

anova(modl2,test="Chi")
Terms added sequentially (first to last)
          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                          4       64.8
conc       1     64.4         3        0.4    1e-15
I(conc^2)  1      0.2         2        0.2     0.67
```

- The first is an explicit model comparison, whereas the second is a serial change to the model. Notice also that the *Wald test* (looking at coefficient/se) supports this conclusion: `I(conc∧2) -0.0612 0.1432 -0.43 0.669`.

GLM Diagnostics, Residuals

- Response residual:

$$r_R = y - \hat{\mu}$$

- Pearson residual:

$$r_P = \frac{y - \hat{\mu}}{\sqrt{\text{Var}(\hat{\mu})}}$$

where $\text{Var}(\hat{\mu}) = b''(\theta)$.

- Deviance residual:

$$r_D = sign(y - \hat{\mu})\sqrt{d_i}$$

from $d_i = 2w_i(y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i))$.

- For example, the Poisson has:

$$r_D = sign(y - \hat{\mu}) \left[2(y \log y/\hat{\mu} - y + \hat{\mu})\right]^{1/2}$$

GLM Diagnostics, Residuals

- For the Bliss dataset, we can get the various types of residuals for `modl`.

```
residuals(modl,"deviance")
        1         2         3         4         5
-0.451015  0.359696  0.000000  0.064302 -0.204493
residuals(modl,"pearson")
          1           2           3           4           5
-4.3252e-01  3.6437e-01 -3.6486e-15  6.4147e-02 -2.0811e-01
residuals(modl,"response")
          1           2           3           4           5
-2.2505e-02  2.8344e-02 -3.3307e-16  4.9898e-03 -1.0828e-02
bliss$dead/30 - fitted(modl)
          1           2           3           4           5
-2.2505e-02  2.8344e-02 -3.3307e-16  4.9898e-03 -1.0828e-02
residuals(modl,"working")
          1           2           3           4           5
-2.7709e-01  1.5614e-01 -1.3323e-15  2.7488e-02 -1.3332e-01
```

- The deviance residuals are the default from `residuals` and `resid`, but if you type: `modl$residuals` only you get the working residuals, which can be confusing.

GLM Diagnostics, Residuals and Influence

- For the linear model we used $\hat{y} = Hy$ where H is $X(X'X)^{-1}X'$ and the leverages are given by the diagonal values. They are a function of X only and the influence depends on h_i and Y_i both.

- IWLS uses weights in it's linear ascent up mount likelihood, and these are totally internally generated (not user-specified weighting), but they do show leverage by reflecting linearity at MLE.

- Form the matrix $W = \text{diag}(w_i)$ where the w_i are the working residuals.

- Then produce the weighted hat matrix according to:

$$H_w = W^{1/2}X(X'WX)^{-1}X'W^{1/2}$$

  then interpret the h_{ij} a similar way as the linear model: gives the influence of the jth on the i^{th} predicted value: $\hat{y}_i$.

- In R:

```
influence(modl)$hat # DIAGONALS OF THE HAT MATRIX
      1       2       3       4       5
0.42550 0.41331 0.32238 0.41331 0.42550
```

Other Common Tools

```
rstudent(modl) # LEAVE ONE OUT (STUDENTIZED) RESIDUALS
        1          2          3          4          5
-0.584786  0.472135  0.000000  0.083866 -0.271835
```

This is: $t_{(i)} = \frac{R_i}{s_{(i)}\sqrt{1-h_{ii}}}$ where R_i is the i^{th} residual from the linear model, $s_{(i)}$ is the standard error when the i^{th} case is omitted, and h_{ii} is the i^{th} diagonal of the hat matrix.

```
influence(modl)$coef
  (Intercept)       conc
1  -0.2140015  0.0806635
2   0.1556719 -0.0470873
3   0.0000000  0.0000000
4  -0.0058417  0.0084177
5   0.0492639 -0.0365734
```

The i^{th} row is the change in the estimated coefficients when the i^{th} case is jackknifed out.

```
cooks.distance(modl)
         1          2          3          4          5
1.2059e-01 7.9710e-02 4.6731e-30 2.4704e-03 2.7917e-02
```

GLM Diagnostics, General

- Two approaches: finding oddness in the data given your model, and challenging the assumptions of the model.
- Most GLM diagnostics are an attempt to use/modify LM diagnostics.
- Also consider predictions, tabular analysis, and in-sample predictions.
- Faraway uses the Galápagos data here:

```
data(gala)
gala <- gala[,-2] # REMOVES Endemics (NATURAL TO THAT ISLAND)
names(gala)
[1] "Species"   "Area"      "Elevation" "Nearest"   "Scruz"     "Adjacent"
```

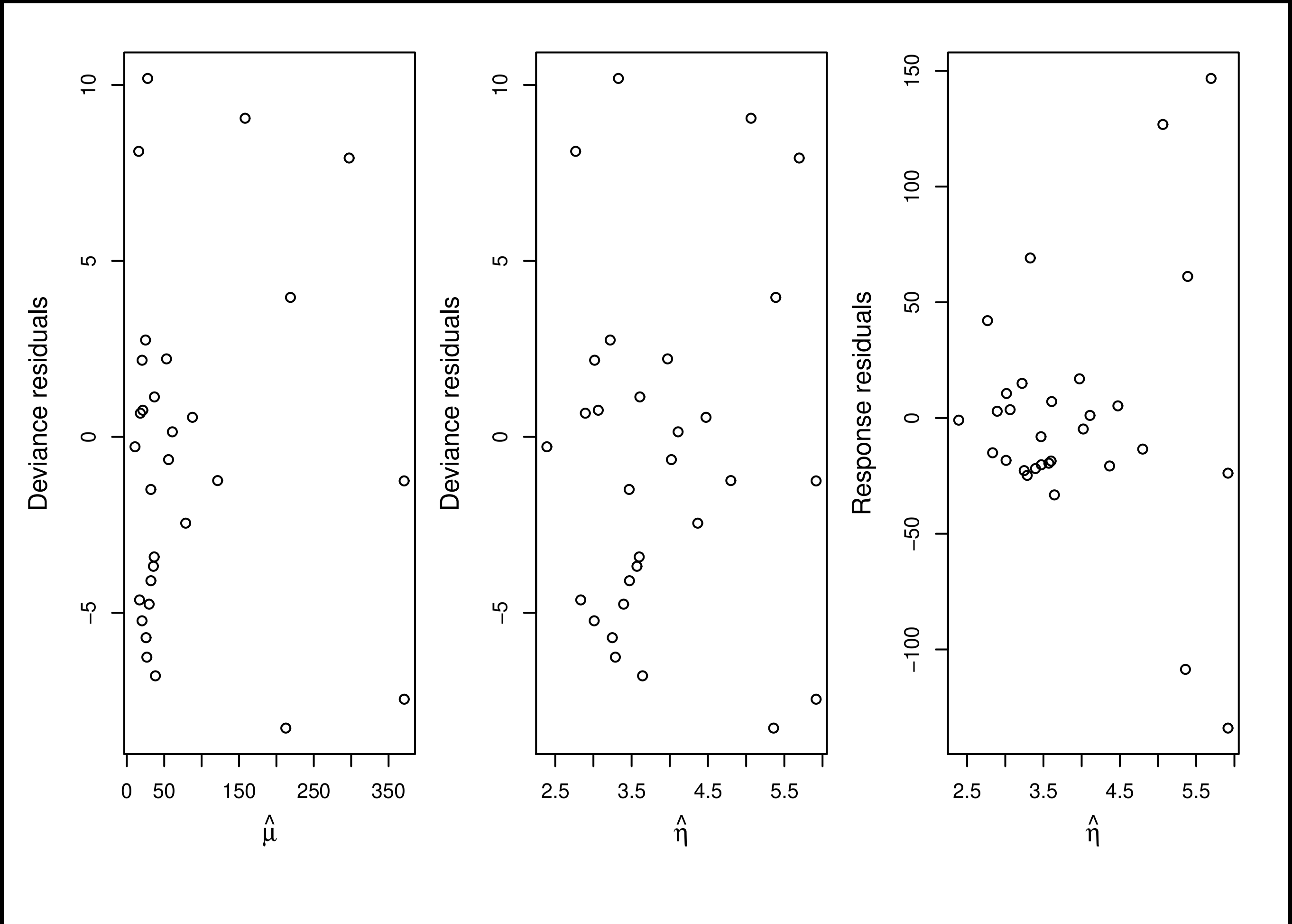
GLM Diagnostics, General

- Faraway's graph:

```
modp <- glm(Species ~ .,family=poisson,data=gala)
postscript("Class.MLE/Images/gala1.ps",width=7,height=5)
par(mfrow=c(1,3),mar=c(6,4,1,1),oma=c(1,1,1,1),cex.lab=1.3,bg="white")

plot(residuals(modp) ~ predict(modp,type="response"),
    xlab=expression(hat(mu)),ylab="Deviance residuals")

plot(residuals(modp) ~ predict(modp,type="link"),
    xlab=expression(hat(eta)),ylab="Deviance residuals")

plot(residuals(modp,type="response") ~ predict(modp,type="link"),
    xlab=expression(hat(eta)),ylab="Response residuals")

dev.off()
```

- This shows that scale of $\hat{\eta}$ is more useful than the scale of $\hat{\mu}$, and that the Pearson residuals can be more revealing (note the classic Poisson increasing spread with level in the last frame).

Galápagos First Figure



GLM Diagnostics, General

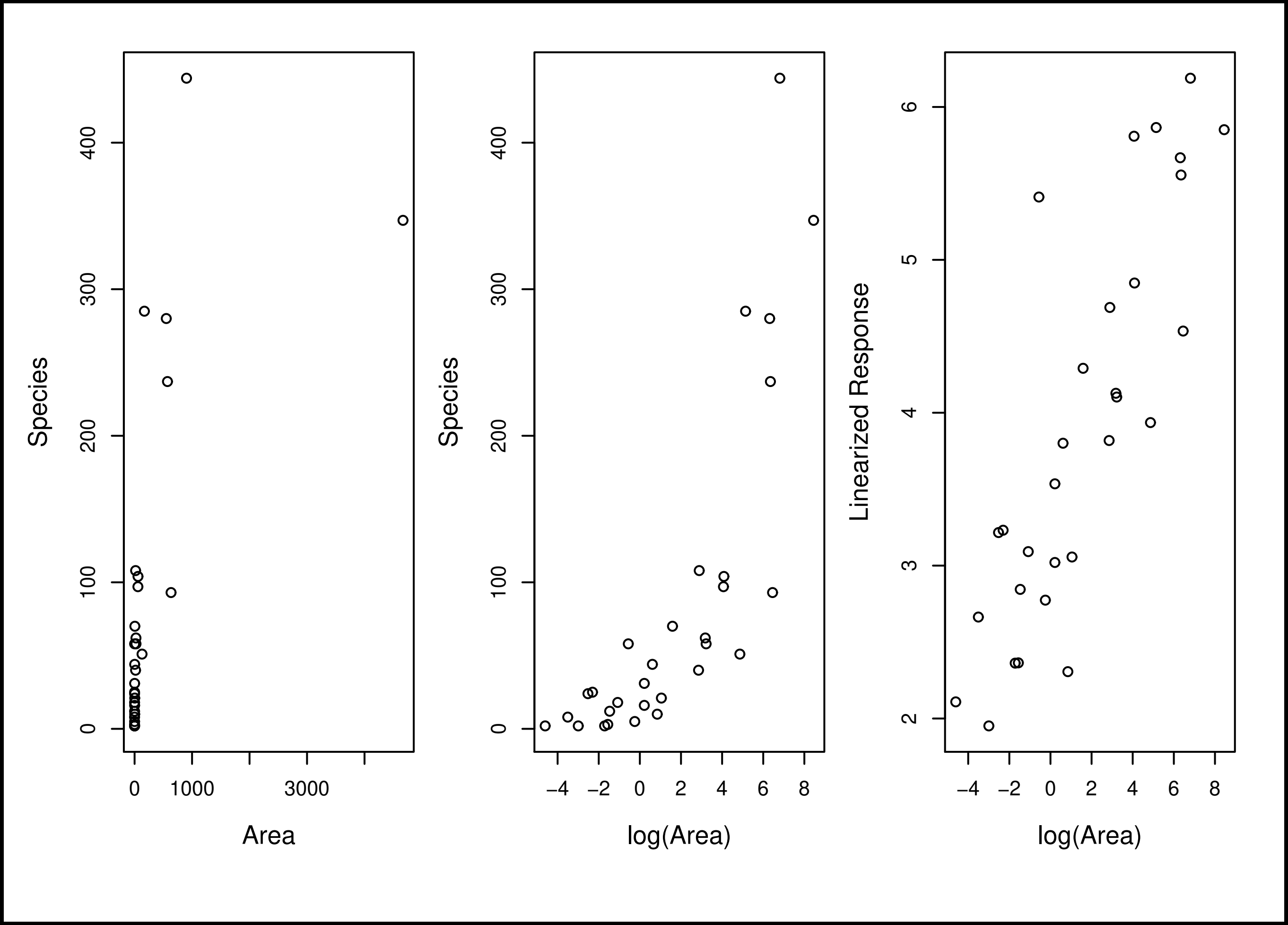
- Sometimes just data plots of bivariate relationships show informative features, but remember that these are only approximations of the real relationships because the other variables are not controlled for.

- The analysis below shows that `Area` needs to be logged to get decent dispersion. To justify this with the log function in the Poisson, create a linearized response from: $z = \eta + (y - \mu)\frac{\partial\eta}{\partial\mu}$, so differences from η are linearized with the derivative.

```
postscript("Class.MLE/Images/gala2.ps",width=7,height=5)
par(mfrow=c(1,3),mar=c(6,4,1,1),oma=c(1,1,1,1),cex.lab=1.3,bg="white")
plot(Species ~ Area, data=gala)
plot(Species ~ log(Area), data=gala)
mu <- predict(modp,type="response")
z <- predict(modp)+(gala$Species-mu)/mu
plot(z ~ log(Area), gala,ylab="Linearized Response")
dev.off()
```

- Where $\frac{\partial\eta}{\partial\mu} = 1/\mu$ since $\eta = g(\mu) = \log(\mu)$ for the Poisson (log-linear) model.

Galápagos Second Figure



GLM Diagnostics, General

- Actually, a quick deviance comparison justifies logging all of the RHS variables.

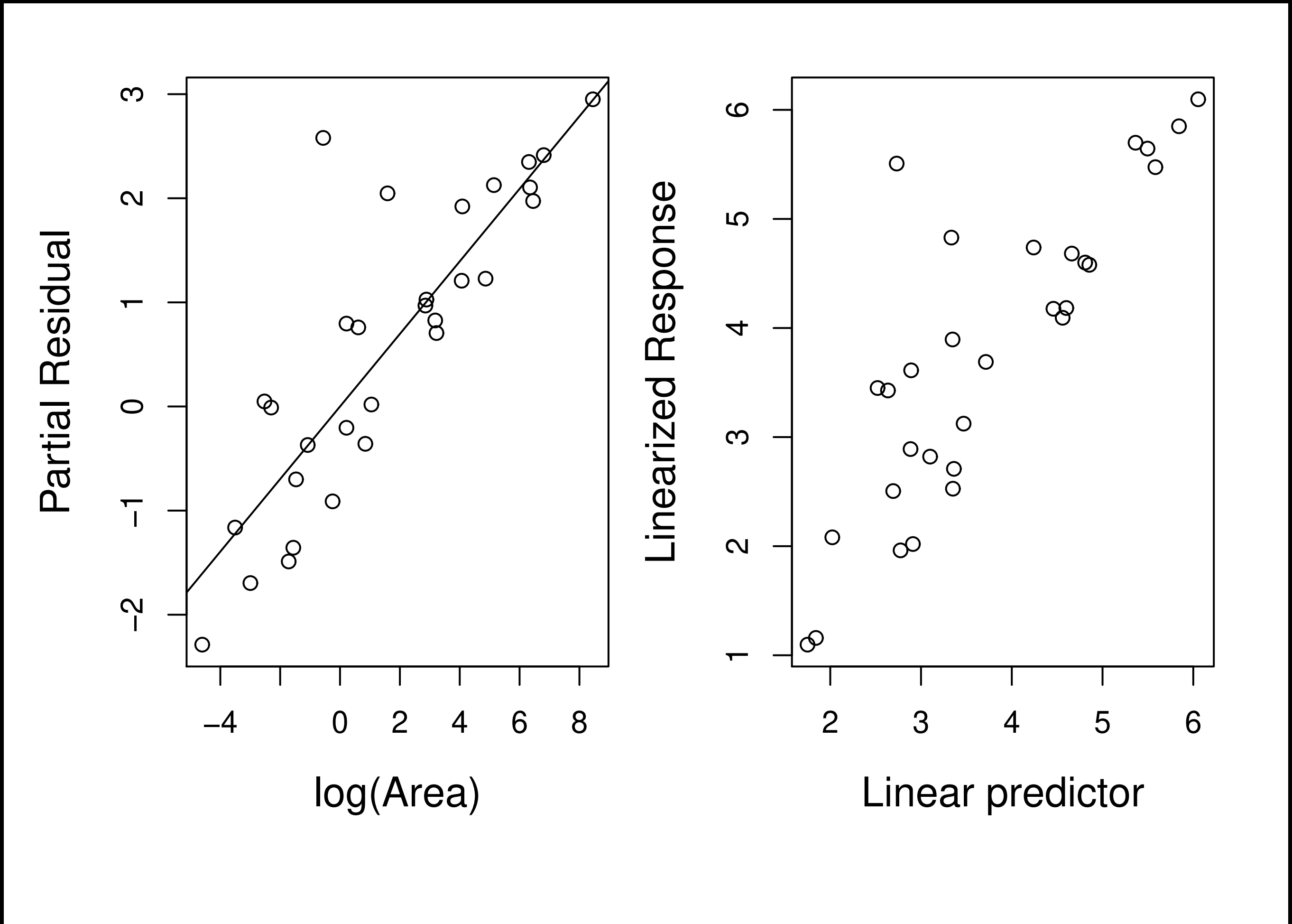
```
modpl <- glm(Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Scruz+0.1)
                      + log(Adjacent), family=poisson, data=gala)
c(deviance(modp),deviance(modpl))
[1] 716.85 359.12
```

- Recall that partial residual plots were useful in linear models. We can get an approximate version of those by plotting $z - \hat{\eta} + \beta_j x_j$ against x_j to look for linearity.

- And we haven't yet plotted the linearized response, z against the linear predictor $\hat{\eta}$.

```
mu <- predict(modpl,type="response")
u <- (gala$Species-mu)/mu + coef(modpl)[2]*log(gala$Area)
postscript("Class.MLE/Images/gala3.ps",width=7,height=5)
par(mfrow=c(1,2),mar=c(6,4,1,1),oma=c(1,1,1,1),cex.lab=1.3,bg="white")
plot(u ~ log(Area), gala,ylab="Partial Residual")
abline(0,coef(modpl)[2])
z <- predict(modpl)+(gala$Species-mu)/mu
plot(z ~ predict(modpl), xlab="Linear predictor", ylab="Linearized Response")
dev.off()
```

Galápagos Third Figure

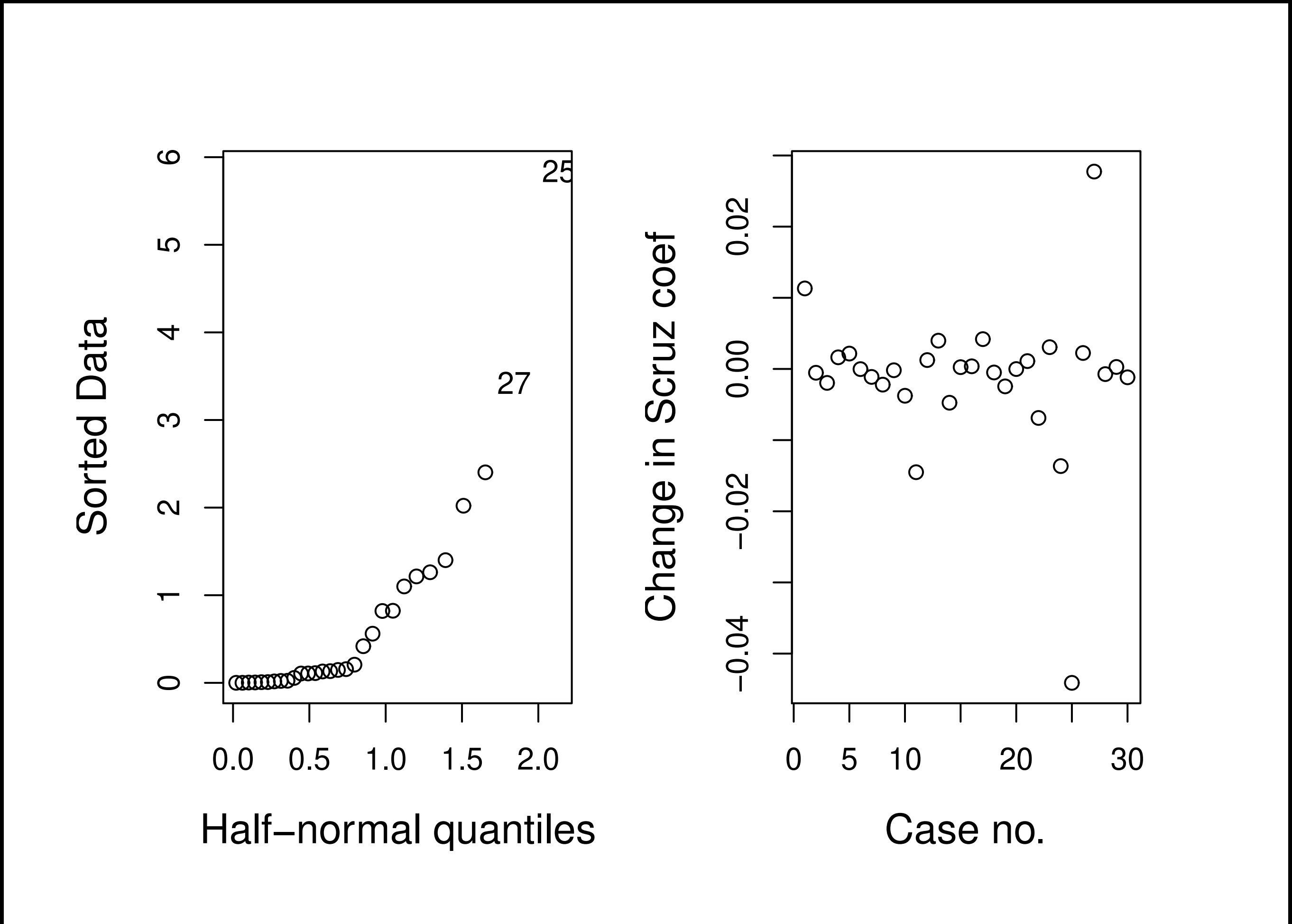


GLM Diagnostics, Unusual Points

- Return to standard ideas; halfnormal plots (ordered absolute values of residuals from a generalized linear model against expected values of normal order statistics), Cook's D plots, and jackknifing out suspect cases.

```
postscript("Class.MLE/Images/gala4.ps",width=7,height=5)
par(mfrow=c(1,2),mar=c(6,4,1,1),oma=c(1,1,1,1),cex.lab=1.3,bg="white")

gali <- influence(modpl)
halfnorm(gali$hat)

plot(gali$coef[,5],ylab="Change in Scruz coef",xlab="Case no.")
dev.off()
```

Galápagos Fourth Figure



GLM Diagnostics, Unusual Points

- A model that drops case number 25 (Santa Cruz)

```
modplr <- glm(Species ~ log(Area) + log(Elevation) + log(Nearest)
                      + log(Scruz+0.1) + log(Adjacent),
              family=poisson, gala, subset=-25)

cbind(coef(modpl),coef(modplr))
(Intercept)       3.287941  3.050699
log(Area)         0.348445  0.334530
log(Elevation)    0.036421  0.059603
log(Nearest)     -0.040644 -0.052548
log(Scruz + 0.1) -0.030045  0.015919
log(Adjacent)    -0.089014 -0.088516
```

GLM Diagnostics, Unusual Points

- A smaller model and rescaling the dispersion:

```
modpla <- glm(Species ~ log(Area)+log(Adjacent), family=poisson, data=gala)
(Intercept)    3.27668    0.04413    74.2   <2e-16
log(Area)      0.37503    0.00802    46.7   <2e-16
log(Adjacent) -0.09575    0.00612   -15.7   <2e-16
(Dispersion parameter for poisson family taken to be 1)

dp <- sum(residuals(modpla,type="pearson")^2)/modpla$df.res
summary(modpla,dispersion=dp)
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)     3.2767     0.1794   18.26  < 2e-16
log(Area)       0.3750     0.0326   11.50  < 2e-16
log(Adjacent)  -0.0957     0.0249   -3.85  0.00012
(Dispersion parameter for poisson family taken to be 16.527)

    Null deviance: 3510.73  on 29  degrees of freedom   # THIS PART IS THE
Residual deviance:  395.54  on 27  degrees of freedom   # SAME FOR BOTH
AIC: 562.4                                              # MODELS
```

Estimation Background

- MLEs produced with *iterative weighted least squares* (IWLS).

- IWLS works for any GLM based on an exponential family form (and for some others).

- IWLS proposed by Nelder and Wedderburn (1972) in the founding article and implemented in the GLIM package by Baker and Nelder 1978.

- All professional-level statistic computing implementations now employ IWLS to find maximum likelihood estimates for generalized linear models.

- Overall strategy: Newton-Raphson with Fisher Scoring applied iteratively to the modified normal equations.

- For detailed analyses and extensions of this procedure, see articles by Green (1984, JRSS-B) and del Pino (1989, StatSci).

Review of Newton-Raphson

- The problem of finding a mode (MLE) can often be reduced to root finding with a derivatives: algorithmically rather than analytically.

- If we visualize the problem of numerical maximum likelihood estimation as that of finding the top of a unimodal function in the k-dimensional parameter space, then its easy to see that this is equivalent to finding the parameter value where the derivative of the likelihood function is equal to zero: the tangent line is horizontal.

- Newton's method is based on a Taylor series expansion about some given point, x_1, relative to a constant, x_0:

$$f(x_1) = f(x_0) + (x_1 - x_0)f'(x_0) + \frac{1}{2}(x_1 - x_0)^2 f''(x_0) + \frac{1}{3!}(x_1 - x_0)^3 f'''(x_0) + \ldots .$$

Review of Newton-Raphson

- Infinite precision is achieved with infinite application of the series and is therefore unobtainable. We generally only care about the first two terms are required as a step in an iterative process.

- Suppose we are interested in finding the point, x_1, such that $f(x_1) = 0$. This is a root of $f()$ in the sense that it provides a solution to the polynomial expressed by the function.

- It can also be thought of as the point where the function crosses the x-axis in a graph of x versus $f(x)$.

- We could find this point from x_0 using the Taylor series expansion in one step if we had an infinite precision calculator:

$$0 = f(x_0) + (x_1 - x_0)f'(x_0) + \frac{1}{2}(x_1 - x_0)^2 f''(x_0) + \frac{1}{3!}(x_1 - x_0)^3 f''(x_0) + \ldots.$$

Review of Newton-Raphson

- Lacking that resource, it is clear from the additive nature of the Taylor series expansion that we could use some subset of the terms on the right hand side to at least get *closer* to the desired point:
$$0 \cong f(x_0) + (x_1 - x_0)f'(x_0).$$

- This shortcut is referred to as the Gauss-Newton method because it is based on Newton's algorithm, but leads to a least squares solution in multivariate problems.

- To emphasize the iterative nature, re-label x_0 as $x^{(j)}$ and x_1 as $x^{(j+1)}$.

Review of Newton-Raphson

- Newton's method is rearranged to produce at the $(j+1)^{\text{th}}$ step:

  $$x^{(j+1)} = x^{(j)} - \frac{f(x^{(j)})}{f'(x^{(j)})}$$

  so that progressively improved estimates are produced until $f(x^{(j+1)})$ is sufficiently close to zero: $x^{(j)} \approx x^{(j+1)}$.

- This method converges rapidly (quadratically in fact) to a solution provided that the selected starting point is reasonably close to the solution. However, the results can be disastrous if this condition is not met.

- If we know for certain that there exists one unique maxima, then the method of "steepest ascent" can be used:

  $$x^{(j+1)} = x^{(j)} - f(x^{(j)}).$$

Review of Newton-Raphson

- Suppose that we wanted a numerical routine for finding the square root of a number, μ.
- This is equivalent to finding the root of the simple equation $f(x) = x^2 - \mu = 0$, with first derivative $\frac{\partial}{\partial x} f(x) = 2x$.
- If we insert these functions into $(j+1)^{\text{th}}$ step:

$$x^{(j+1)} = x^{(j)} - \frac{f(x^{(j)})}{f'(x^{(j)})},$$

we get:

$$\begin{aligned} x^{(j+1)} &= x^{(j)} - \frac{(x^{(j)})^2 - \mu}{2x^{(j)}} \\ &= x^{(j)} - \frac{1}{2}x^{(j)} - \frac{1}{2}\mu(x^{(j)})^{-1} \\ &= \frac{1}{2}(x^{(j)} + \mu(x^{(j)})^{-1}) \end{aligned}$$

Review of Newton-Raphson

- A simple R function:

```
newton.raphson.ex <- function(mu,x,iterations)  {
    for (i in 1:iterations)
         x <- 0.5*(x + mu/x)
     return(x)
}
```

with parameters for the target, the starting point, and the number of iterations.

- Running for different numbers of iterations:

```
> newton.raphson.ex(99,2,3)
[1] 10.74386
> newton.raphson.ex(99,2,6)
[1] 9.949874
```

Gauss-Newton and Root Finding

- The Newton-Raphson algorithm when applied ML mode finding treats the score function as $f()$ to produce *iterative* estimates from the Taylor series:

$$\beta^{(j+1)} = \beta^{(j)} - \frac{\frac{\partial}{\partial\beta}\ell(\beta^{(j)}|\mathbf{y})}{\frac{\partial^2}{\partial\beta\partial\beta}\ell(\beta^{(j)}|\mathbf{y})}.$$

- Generalize this by allowing multiple coefficients:

$$\boldsymbol{\beta}^{(j+1)} = \boldsymbol{\beta}^{(j)} - \left(\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})\right)^{-1}\frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y}).$$

- Where the “Hessian” matrix is:

$$\mathbf{H} = \frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y}).$$

- For exponential family distributions and natural link functions, the observed and expected Hessian matrix are identical (Fahrmeir and Tutz, 1994, p.39; Lehmann and Casella, 1998, pp.124-8).

Gauss-Newton and Root Finding

- So it is common to replace this calculation with forms that are equivalent for the exponential family, such as the Fisher information:

$$\mathbf{A}_F = -E\left(\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})\right)$$

  (Fisher 1925), or the square of the score function:

$$\mathbf{A}_B = E\left[\frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})'\frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})\right].$$

  sometimes called the *BHHH* method (Berndt, Hall, Hall, and & Hausman 1974).

- At each step of the Newton-Raphson algorithm there is a system of multivariate normal equations:

$$\underbrace{\mathbf{A}}_{\text{angle}}\ \underbrace{(\boldsymbol{\beta}^{(j+1)}-\boldsymbol{\beta}^{(j)})}_{\text{direction: }\delta\boldsymbol{\beta}} = \underbrace{\frac{\partial}{\partial\boldsymbol{\beta}^{(j)}}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})}_{\text{size of direction: }\mathbf{u}},$$

  builds "linear structure in the parameter vector" analogous to what we did before for diagnostic purposes: $z = \eta + (y-\mu)\frac{\partial\eta}{\partial\mu}$.

Gauss-Newton and Root Finding

- This creates a linear system of equations according to:

$$\underbrace{\mathbf{A}}_{\text{angle}} \underbrace{(\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})}_{\text{direction: } \delta\boldsymbol{\beta}} = \underbrace{\frac{\partial}{\partial \boldsymbol{\beta}^{(j)}} \ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})}_{\text{size of direction: } \mathbf{u}}, \longrightarrow \underset{(p\times 1)}{\delta\boldsymbol{\beta}} = \underset{(p\times p)}{\mathbf{A}^{-1}} \underset{(p\times 1)}{\mathbf{u}}$$

  where we obtain new estimates by:

$$\boldsymbol{\beta}^* = \boldsymbol{\beta} + \delta\boldsymbol{\beta} = \boldsymbol{\beta} + \mathbf{A}^{-1}\mathbf{u}$$

- Given this system of equations, it is computationally convenient to solve on each iteration by *weighted least squares*.

- Therefore the problem of mode finding reduces to a repeated weighted least squares application in which the inverse of the diagonal values of $\mathbf{A}$ are the appropriate weights.

Review of Weighted Least Squares

- A standard technique for compensating for non-constant error variance in LMs is to insert a diagonal matrix of weights, $\mathbf{\Omega}$, into the calculation of $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ such that the heteroscedasticity is mitigated.

- The $\mathbf{\Omega}$ matrix is created by taking the error variance of the i^{th} case (estimated or known), v_i, and assigning the inverse to the i^{th} diagonal: $\mathbf{\Omega}_{ii} = \frac{1}{v_i}$. The idea is that large error variances are reduced by multiplication of the reciprocal.

- Starting with $\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \epsilon_i$, observe that there is heteroscedasticity in the error term so: $\epsilon_i = \epsilon v_i$, where the shared (minimum) variance is ϵ (i.e. non-indexed), and differences are reflected in the v_i term.

- Really simple example: a heteroscedastic error vector: $\mathbf{E} = [1, 2, 3, 4]$. Then $\epsilon = 1$, and the $\mathbf{v}$ vector is $[1, 2, 3, 4]$. So by the logic above, the $\mathbf{\Omega}$ matrix for this example is:

$$\mathbf{\Omega} = \begin{bmatrix} \frac{1}{v_1} & 0 & 0 & 0 \\ 0 & \frac{1}{v_2} & 0 & 0 \\ 0 & 0 & \frac{1}{v_3} & 0 \\ 0 & 0 & 0 & \frac{1}{v_4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}.$$

Review of Weighted Least Squares

- Premultiply each term by the square root of the $\mathbf{\Omega}$ matrix (a Cholesky factorization given that $\mathbf{A}$ is a positive definite, but greatly simplified here since $\mathbf{\Omega}$ is diagonal).

$$\mathbf{\Omega}^{\frac{1}{2}}\mathbf{Y} = \mathbf{\Omega}^{\frac{1}{2}}\mathbf{X}\boldsymbol{\beta} + \mathbf{\Omega}^{\frac{1}{2}}\boldsymbol{\epsilon}.$$

- So if the heteroscedasticity in the error term is expressed as the diagonals of a matrix: $\boldsymbol{\epsilon} \sim (0, \sigma^2\mathbf{V})$, then this gives: $\boldsymbol{\epsilon} \sim (0, \mathbf{\Omega}\sigma^2\mathbf{V}) = (0, \sigma^2)$, and the heteroscedasticity is “removed.”

- Now instead of minimizing

$$(\mathbf{Y} - \boldsymbol{X}\boldsymbol{\beta})'(\mathbf{Y} - \boldsymbol{X}\boldsymbol{\beta}),$$

we minimize

$$(\mathbf{Y} - \boldsymbol{X}\boldsymbol{\beta})'\mathbf{\Omega}(\mathbf{Y} - \boldsymbol{X}\boldsymbol{\beta}),$$

and the weighted least squares estimator is found by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{\Omega}\mathbf{X})^{-1}\mathbf{X}'\mathbf{\Omega}\mathbf{Y}.$$

GLS in R

```
diastolic.pressure.df <-
    read.table("http://jgill.wustl.edu/data/bloodpressure.data",header=FALSE)
dimnames(diastolic.pressure.df)[[2]] <- c("age","pressure")
summary(diastolic.pressure.df)
```

```
        age            pressure
Min.   :20.00   Min.   : 63.00
1st Qu.:30.25   1st Qu.: 71.00
Median :40.00   Median : 77.00
Mean   :39.57   Mean   : 79.11
3rd Qu.:49.00   3rd Qu.: 85.75
Max.   :59.00   Max.   :109.00
```

GLS in R

```
attach(diastolic.pressure.df)
unweighted.lm <- lm(pressure˜age)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.15693    3.99367  14.061  < 2e-16
age          0.58003    0.09695   5.983 2.05e-07
---
Residual standard error: 8.146 on 52 degrees of freedom
Multiple R-Squared: 0.4077,     Adjusted R-squared: 0.3963
F-statistic: 35.79 on 1 and 52 DF,  p-value: 2.05e-07

plot(age,pressure,pch=3)
abline(unweighted.lm)
```

GLS in R

```
# REGRESS ABSOLUTE VALUE RESIDUALS ON PREDICTOR -> SD FUNCTION
resid.fit <- lm(abs(unweighted.lm$residuals)~age)

# OBTAIN FITTED VALUES FOR THE WEIGHTS
weights.fit <- 1/(resid.fit$fitted.values)^2

# USE THESE WEIGHTS FOR A GLS REGRESSION
weighted.lm <- lm(pressure~age,weights=weights.fit)
```

GLS in R

```
summary(weighted.lm)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0230 -0.9939 -0.0327  0.9250  2.2008

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.56577    2.52092  22.042  < 2e-16
age          0.59634    0.07924   7.526 7.19e-10
---
Residual standard error: 1.213 on 52 degrees of freedom
Multiple R-Squared: 0.5214,     Adjusted R-squared: 0.5122
F-statistic: 56.64 on 1 and 52 DF,  p-value: 7.187e-10
```

Iteratively Weighted Least Squares

- Suppose that the individual variances used to make the reciprocal diagonal values for $\boldsymbol{\Omega} = \boldsymbol{A}^{-1}$ are unknown and cannot be easily estimated.

- It is known that they are a function of the mean of the outcome variable: $v_i = f(E[Y_i])$.

- So if the expected value of the outcome variable, $E[Y_i] = \mu$, and the form of the relation function, $f()$, are known then this is a straightforward estimation procedure.

- Unfortunately, it is not always possible to know the exact form of the relationship between the mean function and the variance structure.

- A solution to this problem is to iteratively estimate the weights, improving the estimate on each cycle using the mean function:

$$\min(\mathbf{A}^{-1}\mathbf{u} - \delta\boldsymbol{\beta})'(\mathbf{A}^{-1}\mathbf{u} - \delta\boldsymbol{\beta})$$

from the linear system of equations:

$$\underset{(p\times 1)}{\delta\boldsymbol{\beta}} = \underset{(p\times p)}{\mathbf{A}^{-1}}\underset{(p\times 1)}{\mathbf{u}} = \mathbf{A}^{-1}\frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta}^{(j)}|\mathbf{y})$$

Iteratively Weighted Least Squares

- Since $\boldsymbol{\mu} = g^{-1}(\boldsymbol{X}\boldsymbol{\beta})$, then the coefficient estimate, $\hat{\boldsymbol{\beta}}$, provides a mean estimate and vice versa.

- Under very general conditions, satisfied by the exponential family of distributions, the iterative weighted least squares procedure finds the mode of the likelihood function, thus producing the maximum likelihood estimate of the unknown coefficient vector, $\hat{\boldsymbol{\beta}}$.

- Furthermore, the matrix produced by: $\hat{\sigma}^2(\mathbf{X}'\mathbf{\Omega}\mathbf{X})^{-1}$ converges in probability to the variance matrix of $\hat{\boldsymbol{\beta}}$ as desired.

Derivation of IWLS

- Define the starting point of the linear predictor by (later done in iterations): $\underset{(n\times 1)}{\hat{\mathbf{E}}_0} = \underset{(n\times p)(p\times 1)}{\mathbf{X}'\boldsymbol{\beta}_0}$ with fitted value $\hat{\boldsymbol{\mu}}_0$ from applying the link function $\hat{\boldsymbol{\mu}}_0 = g^{-1}(\hat{\mathbf{E}}_0)$, where $\boldsymbol{\beta}_0$is some starting point.

- Form the "adjusted dependent variable" according to:

$$\underset{(n\times 1)}{z_0} = \underset{(n\times 1)}{\hat{\mathbf{E}}_0} + \underset{\text{diag}(n\times n)}{\left(\frac{\partial \eta}{\partial \mu}\bigg|_{\hat{\boldsymbol{\mu}}_0}\right)} \underset{(n\times 1)}{(\mathbf{y} - \hat{\boldsymbol{\mu}}_0)}$$

  which is a linearized form of the link function applied to the data. As an example of this derivative function, the Poisson form looks like:

$$\eta = \log(\mu) \implies \frac{\partial \eta}{\partial \mu} = \frac{1}{\mu}$$

- Form the *quadratic weight matrix*, which is the variance of z:

$$\underset{(n\times n)}{w_0^{-1}} = \left(\frac{\partial \eta}{\partial \mu}\bigg|_{\hat{\boldsymbol{\mu}}_0}\right)^2 v(\mu)|_{\hat{\boldsymbol{\mu}}_0}$$

  where $v(\mu)$ is the variance function: $\frac{\partial}{\partial\theta}b'(\theta) = b''(\theta)$.

Derivation of IWLS

- Note that this process is necessarily iterative because both z and w depend on the current fitted value, $\boldsymbol{\mu}_0$.

- Recall the following (at the jth iteration):

$$b'(\theta) = \mu$$

$$b''(\theta) = v(\mu) = \frac{\partial}{\partial\theta}\mu$$

$$\beta_j \mathbf{x}_j = \eta_j \implies \frac{\partial\eta}{\partial\beta_j} = \mathbf{x}_j$$

$$\ell(y_i, \theta) = \frac{y_i\theta - b(\theta)}{\phi} + c(y_i, \phi)$$

with the simplifications: $a(\phi) = \phi$, no grouping.

Derivation of IWLS

- General Scheme:

1. First construct z, w. Regress z on the covariates with weights to get a new interim estimate:

$$\underset{(p\times 1)}{\hat{\boldsymbol{\beta}}_1} = (\underset{(p\times n)}{\mathbf{X}'}\underset{(n\times n)}{w_0}\underset{(n\times p)}{\mathbf{X}})^{-1}\underset{(p\times n)}{\mathbf{X}'}\underset{(n\times n)}{w_0}\underset{(n\times 1)}{z_0}$$

2. Use the coefficient vector estimate to update the linear predictor:

$$\hat{\mathbf{E}}_1 = \mathbf{X}'\hat{\boldsymbol{\beta}}_1$$

3. Iterate:

$$\begin{aligned} z_1, w_1 &\Longrightarrow \hat{\boldsymbol{\beta}}_2, & \hat{\mathbf{E}}_2 &\Longrightarrow z_2, w_2 \\ z_2, w_2 &\Longrightarrow \hat{\boldsymbol{\beta}}_3, & \hat{\mathbf{E}}_3 &\Longrightarrow z_3, w_3 \\ z_3, w_3 &\Longrightarrow \hat{\boldsymbol{\beta}}_4, & \hat{\mathbf{E}}_4 &\Longrightarrow z_4, w_4 \end{aligned}$$

and so on.

What relevant controls and diagnostics exist in R?

- Default parameters:

```
glm.control()
$epsilon
[1] 1e-04

$maxit
[1] 25

$trace
[1] FALSE
```

What relevant controls and diagnostics exist in R?

- Run a simple example from Gelman, etal. Chapters 3 and 4:

```
bioassay.df <- data.frame(freq=c(5,4,1,2,3,5),
                          dose=c(-0.863,-0.296,-0.296,-0.053,-0.053,0.727),
                          death=c(0,0,1,0,1,1))

bioassay.logit.fit <- glm(death~dose,weights=freq,family=binomial(link=logit),
                 maxit=20,trace=T,epsilon=1e-10,data=bioassay.df,start=c(1,1))
Deviance = 12.54704 Iterations - 1
Deviance = 11.96351 Iterations - 2
Deviance = 11.79770 Iterations - 3
Deviance = 11.78159 Iterations - 4
Deviance = 11.78145 Iterations - 5
Deviance = 11.78145 Iterations - 6
Deviance = 11.78145 Iterations - 7
Deviance = 11.78145 Iterations - 8
```

What relevant controls and diagnostics exist in R?

- Look at results:

```
summary.glm(bioassay.logit.fit)
Deviance Residuals:
      1        2        3        4        5        6
-0.1609  -1.2874   1.8308  -1.9450   1.7176   0.1151

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.8734     1.0401   0.840    0.401
dose          7.9128     5.0620   1.563    0.118

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.526  on 5  degrees of freedom
Residual deviance: 11.781  on 4  degrees of freedom
AIC: 15.781

Number of Fisher Scoring iterations: 8
```

What relevant controls and diagnostics exist in R?

- Test for dropping dose in nested model:

```
anova(bioassay.logit.fit,test="Chisq")

     Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                     5    27.5256
dose  1  15.7441         4    11.7814    0.0001

1-pchisq(15.7441,1)
[1] 7.251374e-05
```

- Look at last iteration of the working residuals, the working response minus final linear predictor: $\mathbf{z}_8 - \eta_8 = (\mathbf{y} - \mu)\frac{d\eta}{d\mu}$:

```
residuals(bioassay.logit.fit,type="working")
        1         2         3         4         5         6
-1.002592 -1.230214  5.343782 -2.574698  1.635042  1.001325
```

What relevant controls and diagnostics exist in R?

- Obtain the Cook's values from the last linear iteration to use as an influence approximation:

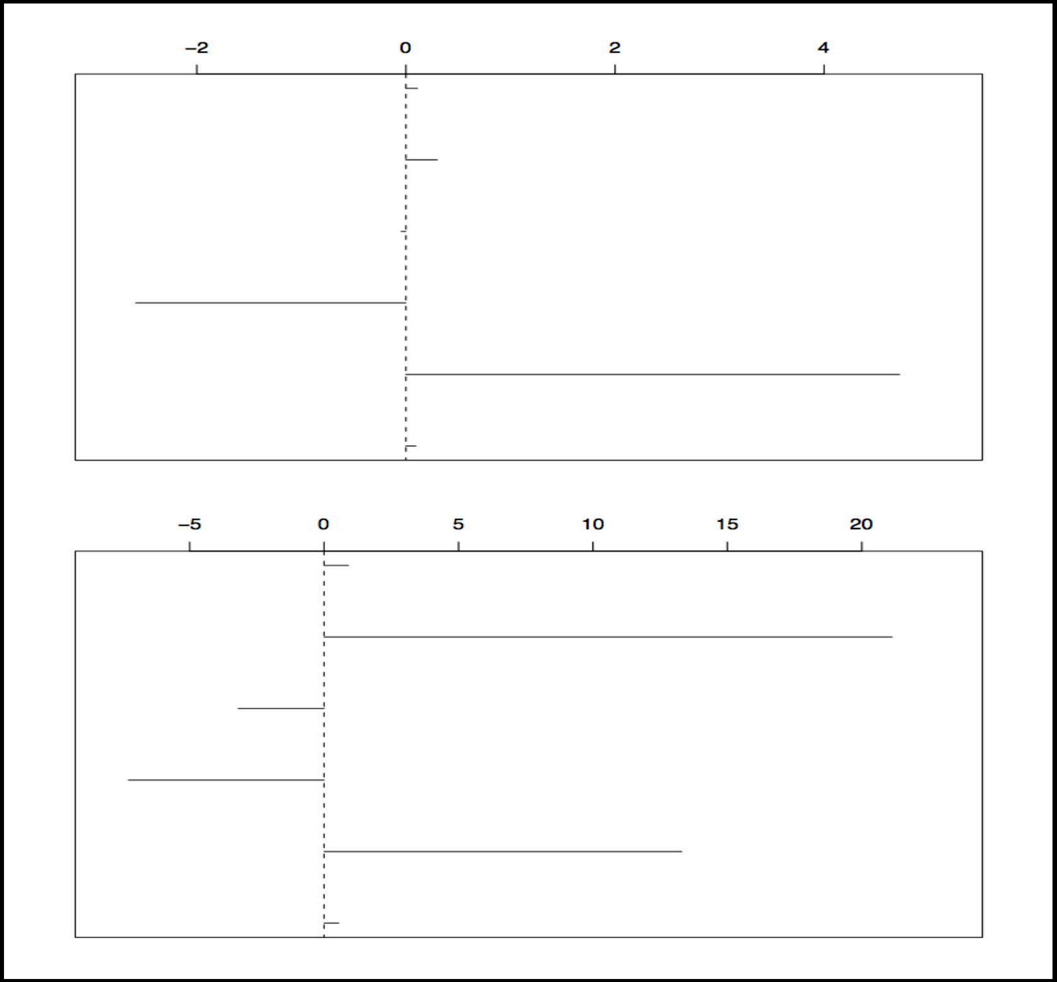
  ```
  cooks.vals <- lm.influence(bioassay.logit.fit)
  par(mfrow=c(1,2),mar=c(2,3,2,2))
  ```

  recalling that `coefficients` is a matrix whose *i*th row contains the change in the estimated model coefficients when the *i*th case is dropped from the data.

- There are only two coefficients to look at (intercept and dose):

  ```
  plot(1:6,rep(0,6),ylim=range(cooks.vals$coefficients[,1])*1.1,
          type="n",xaxt="n",xlab="",ylab="")
  for (i in 1:6) segments(i,0,i,cooks.vals$coefficients[i,1])
  abline(h=0,lty=2)

  plot(1:6,rep(0,6),ylim=range(cooks.vals$coefficients[,2])*1.1,
          type="n",xaxt="n",xlab="",ylab="")
  for (i in 1:6) segments(i,0,i,cooks.vals$coefficients[i,2])
  abline(h=0,lty=2)
  ```

What relevant controls and diagnostics exist in R?



GLMs for Big Data

```
library(biglm)

make.data<-function(urlname, chunksize,...){
      conn<-NULL
     function(reset=FALSE){
     if(reset){
       if(!is.null(conn)) close(conn)
       conn<<-url(urlname,open="r")
     } else{
       rval<-read.table(conn, nrows=chunksize,...)
       if (nrow(rval)==0) {
            close(conn)
            conn<<-NULL
            rval<-NULL
       }
       return(rval)
     }
  }
}
```

GLMs for Big Data

```
airpoll<-make.data("http://faculty.washington.edu/tlumley/NO2.dat",
        chunksize=150,
        col.names=c("logno2","logcars","temp","windsp",
                    "tempgrad","winddir","hour","day"))
dim(airpoll)
NULL

# COULD ALSO USE SQLiteConnection

b <- bigglm(exp(logno2)~logcars+temp+windsp,
         data=airpoll, family=Gamma(log),
         start=c(2,0,0,0),maxit=10)
```

GLMs for Big Data

```
summary(b)

Large data regression model: bigglm(exp(logno2) ~ logcars + temp + windsp,
    data = airpoll, family = Gamma(log), start = c(2, 0, 0, 0), maxit = 100)
Sample size =  500
             Coef  (95%   CI)   SE p
(Intercept)  1.68  1.32  2.04 0.18 0
logcars      0.37  0.32  0.42 0.03 0
temp        -0.03 -0.04 -0.02 0.00 0
windsp      -0.14 -0.17 -0.11 0.02 0
```