(Further) Statistical Methods

Log-Linear Models, Hilary Term, 2015

Marco Scutari

scutari@stats.ox.ac.uk
Department of Statistics
University of Oxford

July 27, 2015

Course Information

Lectures

Week 1: Monday 11am, Friday 10am

Week 2: Monday 11am, Friday 10am

Practical

Week 1: Friday 2pm or 4pm (not assessed)

Week 3: Friday 2pm or 4pm $\longrightarrow$ submit by 10am Monday Week 4

Reference Books (further references in the next slide)

AA Agresti A (2013). Categorical Data Analysis. Wiley, 3rd edition.

MN McCullagh P, Nelder AJ (1989). Generalized Linear Models. Chapman & Hall, 2nd edition.

VR Venables WN, Ripley BD (2002). Modern Applied Statistics with S. Springer, 4th edition.

Other Useful Books on Generalised Linear Models

- Christensen R (1997). Log-Linear Models and Logistic Regression. Springer, 2nd edition.
- Davison AC (2008). Statistical Models. Cambridge University Press.
- Faraway (2006). Extending the Linear Model with R. Chapman & Hall.
- Hastie T, Tibshirani R, Friedman J (2009). The Elements of Statistical Learning. Springer, 2nd edition.
- Kleinbaum DG, Klein M (2010). Logistic Regression: A Self-Learning Text. Springer, 3rd edition.
- von Eye A, Mun E-Y (2013). Log-Linear Models: Concepts, Interpretation and Application. Wiley.
- Wood SN (2006). Generalized Additive Models: An Introduction with R. Chapman & Hall.

Overview

1. Definition and Notations [MN 2; AA 4]
2. Logistic Regression [MN 4; AA 6; VR 7]
3. Log-Linear Regression [MN 6; VR 7]
4. Advanced Models [MN 9; AA 13]

Generalised Linear Models

Recap: In the Previous Episode

Assuming a continuous response and normally-distributed errors constrains the kinds of response variables we can model without transforming them. A general class of models that tackles this problem is generalised linear models (GLMs), which assume the response has a distribution for the exponential family and regresses its expected value through a link function:

$$g(\mathrm{E}(y_i)) = \eta_i \qquad \text{where} \qquad \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p. \tag{1}$$

Some possible choices for the response are:

- the Gaussian distribution to obtain classic linear models;
- the Gamma distribution for non-negative continuous responses, $y_i \in \mathbb{R}^+$;
- the binomial distribution for binary responses, $y_i \in \{0, 1\}$;
- the Poisson distribution for count data, $y_i \in \mathbb{N}$.

We will concentrate on the last two which are by far the most popular non-Gaussian GLMs.

Generalised Linear Models: Binomial Response

For a binary response, the natural assumption is the binomial distribution. So

$$\mathrm{E}(y_i) = \pi_i \qquad \text{and} \qquad g(\pi_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{2}$$

and as a link function we need a $g : [0, 1] \to \mathbb{R}$. Popular candidates that are implemented in R are:

- the logistic (logit) function or log-odds

$$g(\pi) = \log \frac{\pi}{1 - \pi}; \tag{3}$$

- the probit function

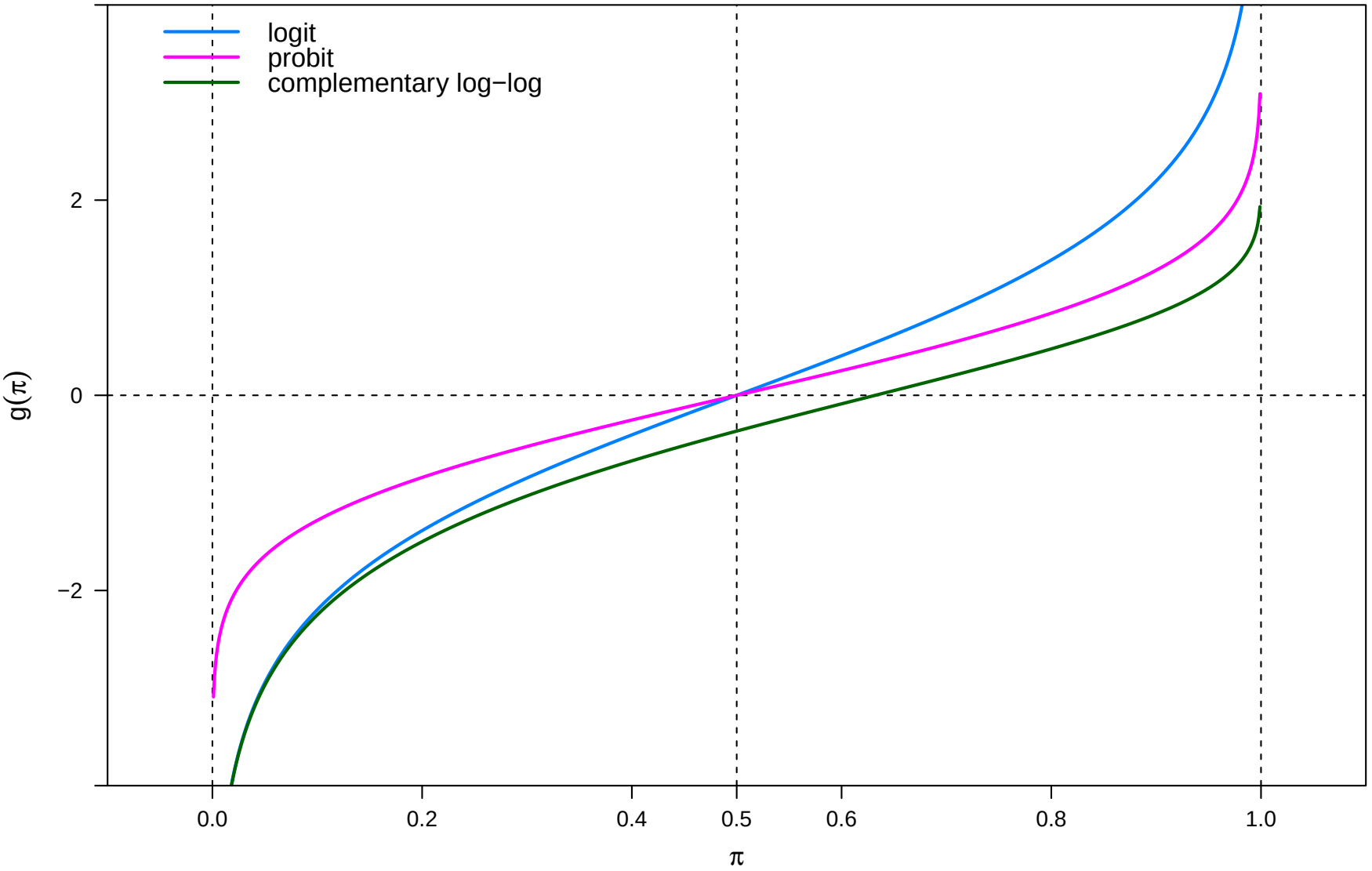
$$g(\pi) = \Phi^{-1}(\pi), \qquad \text{where } \Phi() \text{ is the Normal CDF;} \tag{4}$$

- and the complementary log-log function

$$g(\pi) = \log(-\log(1 - \pi)). \tag{5}$$

Hence the name logistic regression.

Link Functions: Logit, Probit and Complementary Log-Log



Why These Link Functions?

All the link functions have similar shapes. They converge asymptotically to zero for large negative values and to one for large positive values.

The logit and probit functions are symmetric around $\pi = 0.5$, and are almost perfectly linearly related in $[0.1, 0.9]$. Therefore, the logit function is preferable because it is in closed form and it is easy to invert:

$$\text{logit}^{-1}(\eta) = \frac{e^{\eta}}{1 + e^{\eta}}. \tag{6}$$

The complementary log-log function is not symmetric around 0.5, which means that the roles of success and failure are not interchangeable. It is the inverse of the (log-)Weibull CDF; that distribution is used to model extreme values data. It is also useful in some models because it is related to hazard functions in survival data analysis.

Generalised Linear Models: Poisson Response

For count data, the natural assumption is the Poisson distribution. So

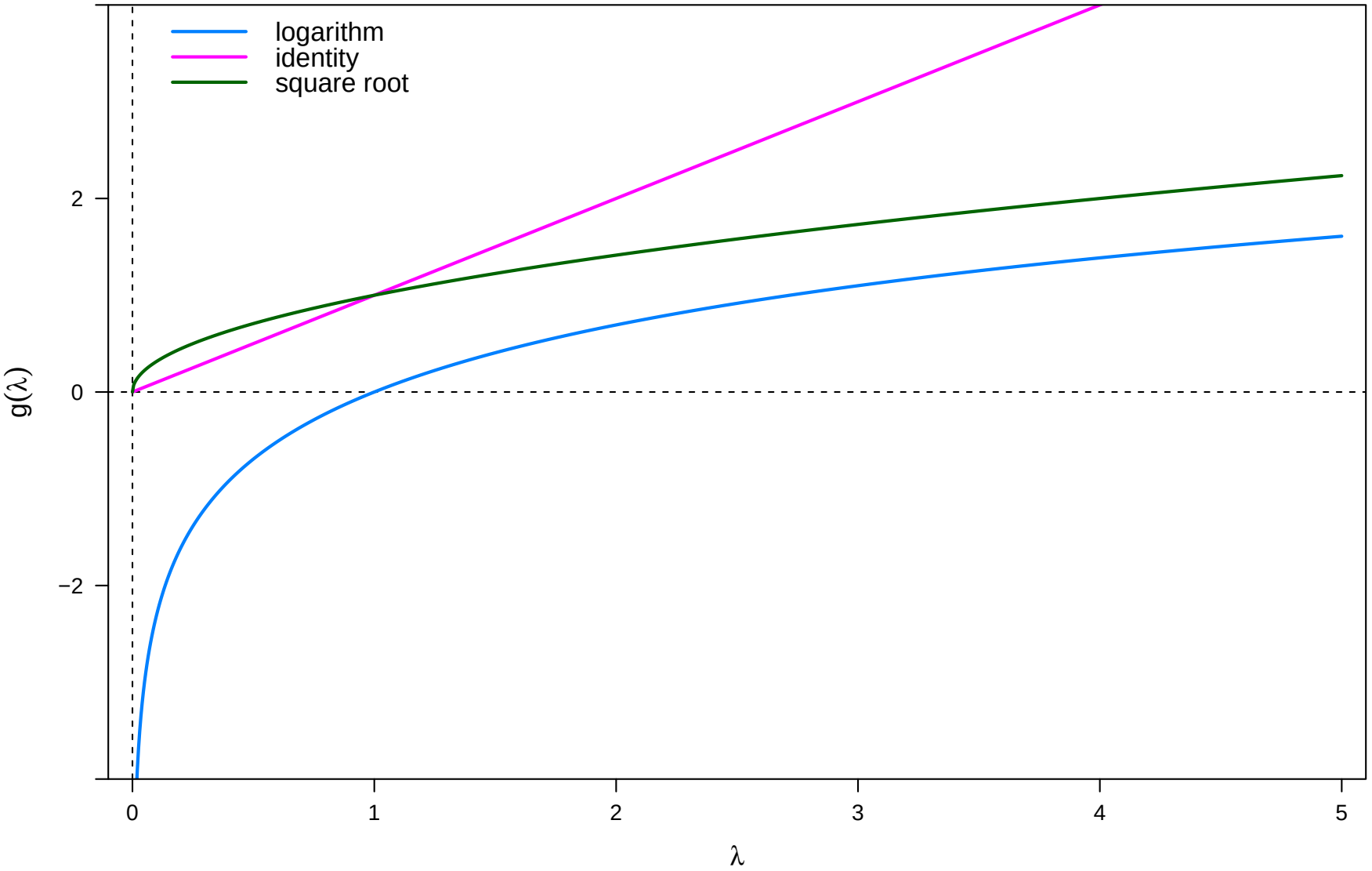
$$\mathrm{E}(y_i) = \lambda_i \qquad \text{and} \qquad g(\lambda_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p$$

and as a link function we need a $g : \mathbb{R}^+ \to \mathbb{R}$. Popular candidates that are implemented in R are:

- the natural logarithm
$$g(\lambda) = \log(\lambda); \tag{7}$$
- the identity function
$$g(\lambda) = \lambda; \tag{8}$$
- the square root
$$g(\lambda) = \sqrt{\lambda}. \tag{9}$$

Hence the name log-linear regression.

Link Functions: Logarithm, Identity and Square Root



Why These Link Functions?

The logarithm is a simple and mathematically elegant transform from $\mathbb{R}^+$ to $\mathbb{R}$, and it has an equally simple and elegant inverse in the exponential.

The identity is a suitable link for a "large enough" because the Poisson distribution is asymptotically normal due to the central limit theorem:

$$Pois(\lambda) \to N(\lambda, \lambda) \qquad \text{as } \lambda \to \infty. \tag{10}$$

In that case the responses will be very far from zero and we can structure the GLM as an ordinary linear model, knowing that the residuals may be strongly heteroschedastic (the variance is λ same as the mean).

The square root is an approximate variance-stabilising transformation (i.e. to make the variability of the values not related to their expectation, as they would not be in a normal distribution). The original is called the Anscombe transform: for $Y \sim Pois(\lambda)$ and

$$g : y \to 2\sqrt{y + \frac{3}{8}} \quad \text{we have} \quad g(Y) \dot{\sim} N\left(2\sqrt{\lambda + \frac{3}{8}} - \frac{1}{4\sqrt{\lambda}}, 1\right). \tag{11}$$

Canonical Link Functions

Distributions in the exponential family have the form

$$f(y;\theta) = \exp\left\{\frac{y\theta - b(\theta)}{a(\psi)} + c(y,\psi)\right\} \tag{12}$$

where θ is called the canonical parameter. ψ is a nuisance parameter which may or may not be known. A link function is called canonical when it links the linear predictor η to the canonical parameter.
For instance, we can re-write the binomial distribution as

$$f(y;\pi) \propto \pi^y(1-\pi)^{n-y} = \exp\left\{y\log\frac{\pi}{1-\pi} + n\log(1-\pi)\right\} \tag{13}$$

so the logit function is the canonical link. For the Poisson,

$$f(y;\lambda) \propto \lambda^y e^{-\lambda} = \exp\left\{y\log\lambda - \lambda\right\}, \tag{14}$$

so the natural logarithm function is the canonical link.

Why Do We Prefer Canonical Link Functions?

When using a canonical link function:

- $\mathbf{X}^T\mathbf{y}$ is the sufficient statistic for θ.
- The residuals will sum up to zero, though they are not likely to be symmetric unless the response variable is normal.
- Deriving maximum likelihood estimates is much simpler than with non-canonical link functions (more below): Newton-Raphson and Fisher scoring coincide.
- If observations come with weights, they are conserved between the raw data y_i and the estimated data $\hat{y}_i$.
- Interpretation of the regression is typically intuitive (for some values of "intuitive"): think odds (for the Binomial) and multiplicative effects (for the Poisson).

Mean and Variance in Exponential Families

The log-likelihood function from (12) is

$$l(\theta; y) = \frac{y\theta - b(\theta)}{a(\psi)} + c(y, \psi); \tag{15}$$

and the first and second order derivatives are

$$l'(\theta; y) = \frac{y - b'(\theta)}{a(\psi)} \qquad \text{and} \qquad l''(\theta; y) = -\frac{b''(\theta)}{a(\psi)}. \tag{16}$$

Since we know that $\mathrm{E}[l'(\theta; Y)] = 0$, if we substitute we have that

$$\mathrm{E}\left[\frac{Y - b'(\theta)}{a(\psi)}\right] = 0 \quad \Rightarrow \quad \frac{\mathrm{E}(Y) - b'(\theta)}{a(\psi)} = 0 \quad \Rightarrow \quad \mathrm{E}(Y) = b'(\theta). \tag{17}$$

We also know that $\mathrm{E}[l''(\theta; Y)] + \mathrm{E}[l'(\theta; Y)]^2 = 0$, so

$$\mathrm{E}\left[-\frac{b''(\theta)}{a(\psi)}\right] + \mathrm{E}\left[\frac{y - b'(\theta)}{a(\psi)}\right]^2 = 0 \Rightarrow -\frac{b''(\theta)}{a(\psi)} + \frac{\mathrm{VAR}(Y)}{a(\psi)^2} = 0$$

$$\Rightarrow \mathrm{VAR}(Y) = b''(\theta)a(\psi). \tag{18}$$

The Dispersion Parameter

The variance of the data in the model is expressed as

$$\mathrm{VAR}(Y) = b''(\theta)a(\psi) \tag{19}$$

and $a(\psi)$ is commonly of the form $a(\psi_i) = \psi/w_i$, where

- ψ is called the dispersion parameter and
- w is set of weights that may or may not be different for different observations, depending on the distribution.

NOTE: for the distributions which do not have a dispersion parameter separate from the expectation (the normal does, the binomial and the Poisson do not), fitting a generalised linear model may result in overdispersion or underdispersion when does not display the right amount of variability for its mean value.

A Breakdown of the Gaussian Distribution

The exponential family form is

$$f(\pi; y) = \exp\left\{-\frac{(y^2 - 2y\mu + \mu^2)}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right\} \tag{20}$$

so the various components are

$$a(\psi) = \sigma^2, \qquad \theta = \mu, \qquad b(\theta) = \frac{1}{2}\theta^2, \tag{21}$$

$$c(y; \psi) = -\frac{1}{2\psi}y^2 - \frac{1}{2}\log(2\pi\sigma^2). \tag{22}$$

Then the canonical link function and the variance are

$$\mathrm{E}(Y) = \mu = \theta \quad \Rightarrow \quad \text{the identity link, and} \tag{23}$$

$$\mathrm{VAR}(Y) = \sigma^2 \quad \Rightarrow \quad \text{the dispersion (scale) parameter.} \tag{24}$$

A Breakdown of the Binomial Distribution

The exponential family form is

$$f(\pi; y, n) = \exp\left\{ y \log \frac{\pi}{1-\pi} + n \log(1-\pi) + \log \binom{n}{y} \right\} \tag{25}$$

so the various components are

$$a(\psi) = 1, \quad \theta = \log \frac{\pi}{1-\pi}, \quad b(\theta) = n \log(1 + e^{\theta}), \quad c(y; \psi) = \log \binom{n}{y}. \tag{26}$$

Then the canonical link function and the variance are

$$\mathrm{E}(Y) = n\pi = n \frac{e^{\theta}}{1 + e^{\theta}} \quad \Rightarrow \quad \theta = \log \frac{\pi}{1-\pi} \quad \text{and} \tag{27}$$

$$\mathrm{VAR}(Y) = \frac{e^{\theta}}{(1 + e^{\theta})^2} \quad \Rightarrow \quad \frac{1}{a(\psi)} \mathrm{VAR}(Y) = n\pi(1-\pi). \tag{28}$$

A Breakdown of the Poisson Distribution

The exponential family form is

$$f(\lambda; y) = \exp\{y \log \lambda - \lambda - \log y!\} \tag{29}$$

so the various components are

$$a(\psi) = 1, \quad \theta = \log \lambda, \quad b(\theta) = e^{\theta}, \quad c(y; \psi) = -\log y! \tag{30}$$

Then the canonical link function and the variance are

$$\mathrm{E}(Y) = \lambda = e^{\theta} \qquad \Rightarrow \qquad \theta = \log \lambda \tag{31}$$

and

$$\mathrm{VAR}(Y) = e^{\theta} = \lambda. \tag{32}$$

Exponential Family and Maximum Likelihood Estimation

Maximum likelihood estimates for $\hat{\boldsymbol{\beta}}$ can be derived through iteratively (re-)weighted least squares (IWLS). Say $\mathrm{E}(Y_i) = \mu_i$. Then we change variable a few times and string the resulting derivatives to compute $l'(\beta_j)$, $j = 1, \ldots, p$

$$\frac{dl(\beta_j)}{d\beta_j} = \frac{dl(\theta)}{d\theta}\frac{d\theta(\mu)}{d\mu}\frac{d\mu(\eta)}{d\eta}\frac{d\eta(\beta_j)}{d\beta_j} = \sum_{i=1}^{n} \frac{W}{a(\psi)}(y_i - \mu_i)\frac{d\eta}{d\mu}x_{ij}. \quad (33)$$

Then we compute Fisher's information

$$\mathrm{E}\left(-\frac{d^2 l}{d\beta_r d\beta_s}\right) = \mathbf{A} \qquad \text{that has} \qquad \mathbf{A}_{rs} = \sum_{i=1}^{n} W_i x_{ir} x_{is} \quad (34)$$

and use the two order of derivatives to iteratively update the estimates of β_j with Newton-Raphson or Fisher scoring. The weights W are updated in each iteration along with the estimates.

Goodness of Fit: The Deviance

The main measure of goodness of fit is the deviance, which is twice the difference between the log-likelihoods of two nested models. It is defined as

$$D = 2\left(l(\hat{\boldsymbol{\beta}}) - l(\tilde{\boldsymbol{\beta}})\right) = 2\sum_{i=1}^{n} \frac{y_i(\hat{\theta} - \tilde{\theta}) - b(\hat{\theta}) + b(\tilde{\theta})}{a(\psi_i)} \tag{35}$$

which in practice is a log-likelihood ratio test statistic and is asymptotically distributed as a χ^2 distribution whose degrees of freedom are given by the difference in number of the free parameters. If we assume that $a(\psi_i) = \psi/w_i$ then we can define the (un)scaled deviance

$$D^* = \psi D = \sum_{i=1}^{n} 2w_i \left(y_i(\hat{\theta} - \tilde{\theta}) - b(\hat{\theta}) + b(\tilde{\theta})\right). \tag{36}$$

for distribution with a meaningful dispersion parameter.

The Null and Residual Deviance

The two most basic forms of deviance used in model selection are:

- The null deviance

$$D_N = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_0)\right] \sim \chi^2_{n-1}, \tag{37}$$

comparing the saturated model $\mathcal{M}_S$ and the empty model $\mathcal{M}_0$. It is useful as a term of comparison and to give the scale for other deviances.

- The residual deviance

$$D_R = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_L)\right] \sim \chi^2_{n-p-1}, \tag{38}$$

comparing the the saturated model $\mathcal{M}_S$ and the model $\mathcal{M}_L$ estimated from the data. For Gaussian GLMs, the residual deviance is (surprise!) the scaled residual sum of squares

$$D_R = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{\sigma^2} \sim \chi^2_{n-p-1}. \tag{39}$$

Analysis of Deviance

Like the analysis of variance, analysis of deviance can be used to decompose the deviance of a full fitted model into independent components associated with the explanatory variables. Starting from the null deviance we can split out the residual deviance

$$D_N = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_0)\right] = 2\left[\underbrace{l(\mathcal{M}_S) - l(\mathcal{M}_L)}_{\text{residual deviance } D_R} + \underbrace{l(\mathcal{M}_L) - l(\mathcal{M}_0)}_{\text{model deviance}}\right], \quad (40)$$

and then we can split the component related to each explanatory variable:

$$D_N = D_R + 2\left[\underbrace{l(\mathcal{M}_L) - l(\mathcal{M}_{\beta_{p-1}})}_{\text{component for } \beta_p} + \underbrace{l(\mathcal{M}_{\beta_{p-1}}) - l(\mathcal{M}_{\beta_{p-2}})}_{\text{component for } \beta_{p-1}} + \right.$$

$$\left. + \ldots + \underbrace{l(\mathcal{M}_{\beta_1}) - l(\mathcal{M}_0)}_{\text{component for } \beta_1}\right]. \quad (41)$$

Other Model Selection Criteria

Aside from using deviance instead of residual variance, model selection is mostly the same as for Gaussian GLMs.

- We can build tables like ANOVA tables with deviance contributions and χ^2 tests.
- We can use AIC and BIC to compare models that are not necessarily nested; but we can also use p-values from residual deviance to the same effect.
- For predictive models, we can use cross-validation to compute predictive correlations (for continuous responses), true positives & negatives (for binary responses) or classification errors (for categorical responses).

Model assumptions and goodness-of-fit should be checked in the process of selecting a model, to make sure the selected model makes sense.

Residuals

The definition of the residuals is more ambiguous than in the case of classic linear models because of the link function and because their standard error is not necessarily a relevant quantity. Two common takes are:

- Pearson's residuals,

$$r_P = \frac{y_i - \mu_i}{\sqrt{\mathrm{V}(\mu_i)}} \qquad \text{where } \mathrm{V}(\mu_i) = \mathrm{VAR}(Y_i)/a(\psi) = b''(\theta), \tag{42}$$

  which are classic residuals standardised by the variance adjusted for the dispersion parameter. Note that if Y_i is Poisson then

$$\sum_{i=1}^{n} r_P^2 = \text{Pearson's } X^2. \tag{43}$$

- the deviance residuals,

$$r_D = \mathrm{sign}(y_i - \mu_i)\sqrt{d_i} \qquad \text{where} \qquad \sum_{i=1}^{n} d_i = D \tag{44}$$

  and D is the deviance of the model.

Properties of the Residuals

- The residuals of a generalised linear model are not normally distributed. If the response is continuous but non-normal, Pearson's residuals residuals are skewed in general. If the response is discrete, residuals usually appear in stripe patterns, with one stripe for each level of the response.
- For both definitions, the sum of the squared residuals is approximately distributed as a χ^2.
- Pearson's residuals have approximately zero mean and constant variance $a(\psi)$ but they can be quite skewed.
- Deviance residuals are more likely to look like they are normally distributed. If all parameters for the model are known, $D \sim \chi^2_n$, $d_i \sim \chi^2_1$ i.i.d. and therefore $\sqrt{d_i} \sim N(0, 1)$. Plus, they can be interpreted as the contribution of the ith observation to D.
- Model selection often tries to minimise deviance residuals, (almost) never Pearson's residuals.

Residuals vs Fitted Values: an Example

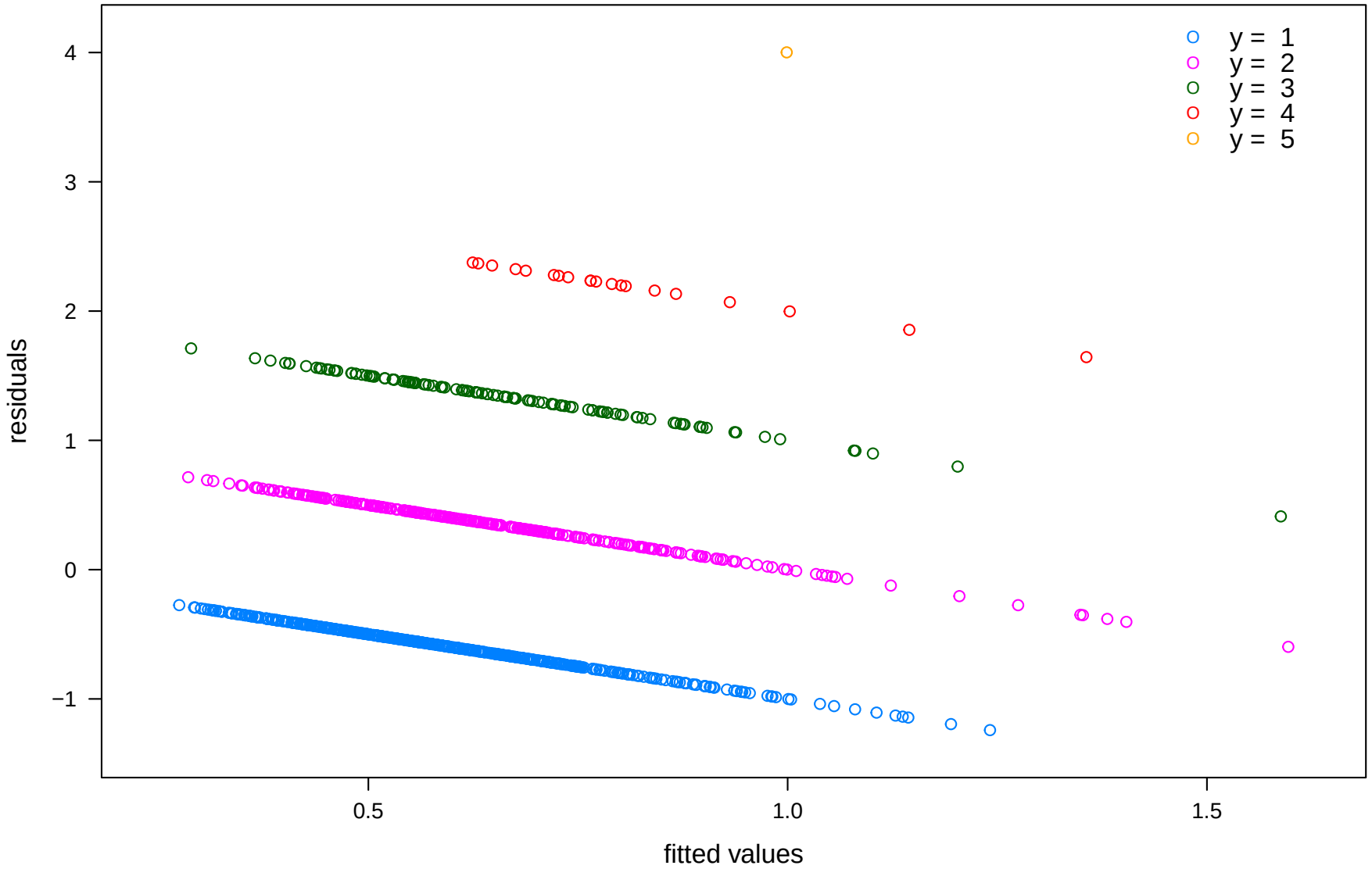
The stripe pattern in the residuals appears even if the model is perfectly specified, as the Poisson GLM below.

```
> set.seed(123)
> n = 10^3
> k = 5
> beta = rnorm(k, sd = 0.2)
> x = matrix(rnorm(n * k), ncol = k)
> y = rpois(n, lambda = exp(-0.5 + x %*% beta + 0.1 * rnorm(n)))
> m = glm(y ~ x, family = poisson)
```

To highlight which residual corresponds to which value of `y`, we can produce a custom plot with one colour for each observed value of the response.

```
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[1:max(y)]
> xyplot(y - fitted(m) ~ fitted(m), group = y,
+   xlab = "fitted values", ylab = "residuals",
+   key = list(points = list(col = col, pch = "O"), corner = c(0.95, 0.97),
+              text = list(paste("y = ", as.character(1:max(y))))),
+   col = col, scales = list(tck = c(1, 0)))
```

Residuals vs Fitted Values: an Example



Estimating the Dispersion Parameter

For the binomial and Poisson GLMs, $a(\psi)$ is a constant and therefore there is nothing to estimate. For other distributions, the dispersion parameter is a nuisance parameter that estimated as:

$$a(\psi) = \frac{D_R}{n-p-1} = \frac{1}{n-p-1}\sum_{i=1}^{n} d_i, \tag{45}$$

which equivalent to the mean of the squared deviance residuals.

For Gaussian GLMs, we obtain the unbiased estimate of the residual's variance:

$$a(\psi) = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{n-p-1} = \sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{n-p-1} = \hat{\sigma}_\varepsilon^2. \tag{46}$$

Overdispersion

The downside of not having a dispersion to estimate is that the variance $\mathrm{VAR}(Y)$ is a function of the same parameter as $\mathrm{E}(Y)$; but model estimation only cares about the latter. Therefore, the data may have more than the allotted amount of variability (overdispersion) or less than that (underdispersion).

The estimates of the regression coefficients are not influenced by either, because the μ_i do not depend on $a(\psi)$. But all the goodness-of-fit statistics are biased unless the asymptotic χ^2 distributions are rescaled with an inflation factor, e.g. $D_R \sim \psi^* \chi^2_{n-p-1}$.

There are several model that extend GLMs to handle such data sets under more relaxed assumptions: the beta-binomial model, the gamma-poisson model, quasi-likelihood models, random-effects models, etc.

How Do We Discover Overdispersion?

The common way of assessing overdispersion is to compare the residual deviance against its degrees of freedom, because the two quantities should be similar (asymptotically, for large $n\pi_i$ and λ_i the binomial and the Poisson distributions are almost Gaussian).

In practice, using the sum of the squared Pearson's residuals to check whether

$$\frac{1}{n-p-1}\sum_{i=1}^{n}(r_P^2)_i \simeq 1 \tag{47}$$

as opposed to using the deviance residuals r_D^2 has much less bias.

The Anolis Lizards Data Set

This small data set is from Fienberg's (1980) book on categorical data analysis.

```
> lizards = read.table("lizards.txt", header = TRUE)
> str(lizards)
'data.frame':   409 obs. of  3 variables:
 $ Species : Factor w/ 2 levels "Sagrei","Distichus": 1 1 1 1 1 1 1 ...
 $ Diameter: Factor w/ 2 levels "narrow","wide": 1 1 1 1 1 1 1 ...
 $ Height  : Factor w/ 2 levels "high","low": 2 2 2 2 2 2 2 ...
```

For a sample of 409 lizards, the following variables were recorded:

- the species, which can be either `Sagrei` or `Distichus`;
- the height of the branch they were perched on, discretised in two categories `narrow` ($\leqslant 4$in) and `wide` (> 4in);
- the diameter of that same branch, discretised in two categories `high` (> 4.75ft) and `low` ($\leqslant 4.75$ft).

Fitting GLMs: the `glm()` Function

The `glm()` function is the analogous of `lm()` for GLMs, and has a similar syntax. The main difference lies in the family argument, which specifies the distribution we are assuming for the response and (optionally) the link function. The default is the canonical link.

```
> m = glm(Species ~ Diameter + Height, data = lizards,
+         family = binomial(link = logit))
> m

Call:  glm(formula = Species ~ Diameter + Height,
    family = binomial(link = logit), data = lizards)

Coefficients:
 (Intercept)  Diameterwide     Heightlow
     -0.1437        0.8029        0.7511

Degrees of Freedom: 408 Total (i.e. Null);  406 Residual
Null Deviance:      550.8
Residual Deviance: 526.6        AIC: 532.6
```

Models From glm() and summary()

```
> summary(m)

Call:
glm(formula = Species ~ Diameter + Height, family = binomial(link = logit),
    data = lizards)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8048  -1.1170   0.6609   0.9326   1.2390

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.1437     0.1503  -0.956 0.338972
Diameterwide   0.8029     0.2198   3.652 0.000260 ***
Heightlow      0.7511     0.2242   3.350 0.000807 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 550.85  on 408  degrees of freedom
Residual deviance: 526.57  on 406  degrees of freedom
AIC: 532.57
```

summary(m): Regression Coefficients

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.1437     0.1503  -0.956 0.338972
Diameterwide   0.8029     0.2198   3.652 0.000260 ***
Heightlow      0.7511     0.2242   3.350 0.000807 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The p-values for the Wald tests are computed using z-scores (as opposed to the t-scores used for `lm()` models), which are defined as

$$z_{\beta_i} = t^2_{\beta_i} = \frac{(\hat{\beta}_i - \beta_0)^2}{\mathrm{VAR}(\hat{\beta}_i)} \sim \chi^2_1 \qquad \text{(asymptotic)} \qquad (48)$$

and $\mathrm{VAR}(\hat{\beta}_i)$ comes from the IWLS. For instance, for `Diameter`:

```
> pchisq((0.8029 / 0.2198)^2, 1, lower.tail = FALSE)
[1] 0.0002593293
```

summary(m): Goodness of Fit

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 550.85  on 408  degrees of freedom
Residual deviance: 526.57  on 406  degrees of freedom
AIC: 532.57
```

The null and residual deviance are reported with the respective degrees of freedom. If $l(\mathcal{M}_S) = 0$, as in this case, we have

$$D_R = -2l(\mathcal{M}_L) \quad \text{which means} \quad \text{AIC} = D_R + 2(p+1). \tag{49}$$

R^2 is not reported, because even though a few pseudo-R^2 coefficients have been defined they are difficult to interpret. One example is the generalised R^2 from Cox & Snell:

$$R^2_{\text{GLM}} = 1 - \left[\frac{L(\mathcal{M}_0)}{L(\mathcal{M}_L)}\right]^{2/n} \in [0, R^2_{\max}] \tag{50}$$

which needs to be rescaled for logistic regression to be defined in $[0, 1]$.

Key Quantities Pre-Computed by `glm()`

- The fitted values $\hat{\mu}_i$, obtained by transforming the linear predictors by the inverse of the link function, i.e. $g^{-1}(\eta_i)$.

```
> fitted(m)
      1       2       3       4       5       6
0.64733 0.64733 0.64733 0.64733 0.64733 0.64733 [...]
```

- The residuals $\hat{\varepsilon}_i$ from the final iteration of the IWLS algorithm.

```
> resid(m)
      1       2       3       4       5       6
-1.44377 -1.44377 -1.44377 -1.44377 -1.44377 -1.44377 [...]
```

  Note that, unlike fitted values, they are on the scale of the link function so they are different from `y - fitted(m)`.

- The intercept and regression coefficients $\hat{\boldsymbol{\beta}}$.

```
> coef(m)
 (Intercept) Diameterwide    Heightlow
  -0.1437035    0.8028584    0.7510606
```

The `predict()` Function

`predict()` produces predicted or fitted values (if no `newdata` is passed to the function) on two scales:

- on the scale of the linear predictors, i.e. the $\hat{\eta}_i$;

```
> predict(m, type = "link")
      1       2       3       4       5       6
0.60735 0.60735 0.60735 0.60735 0.60735 0.60735 [...]
```

- on the scale of the response, i.e. $\hat{\mu}_i$;

```
> predict(m, type = "response")
      1       2       3       4       5       6
0.64733 0.64733 0.64733 0.64733 0.64733 0.64733 [...]
```

For example, for a logistic regression the former returns

$$\hat{\eta}_i = \log \frac{\hat{\pi}_i}{1 - \hat{\pi}_i} \tag{51}$$

while the latter returns $\hat{\pi}_i$.

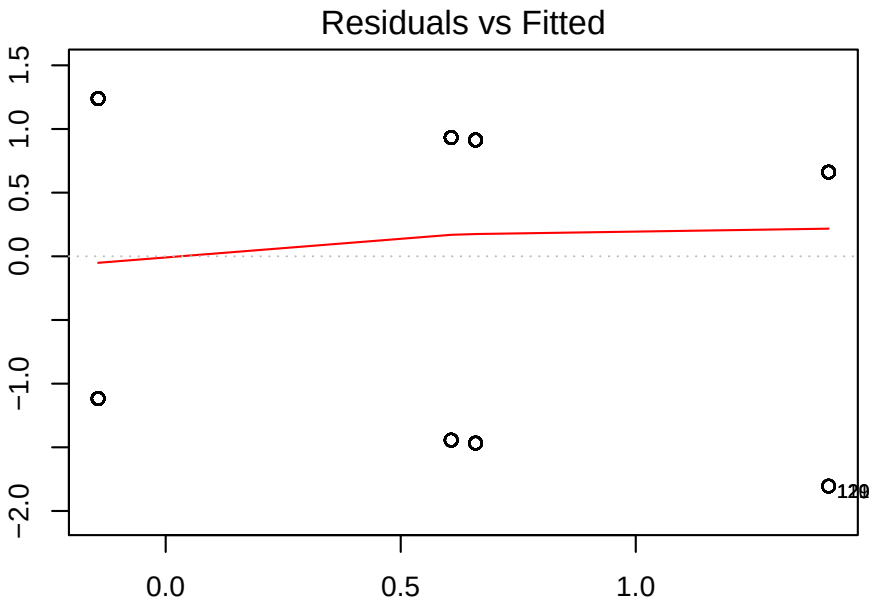
Deviance Tables from `anova()`

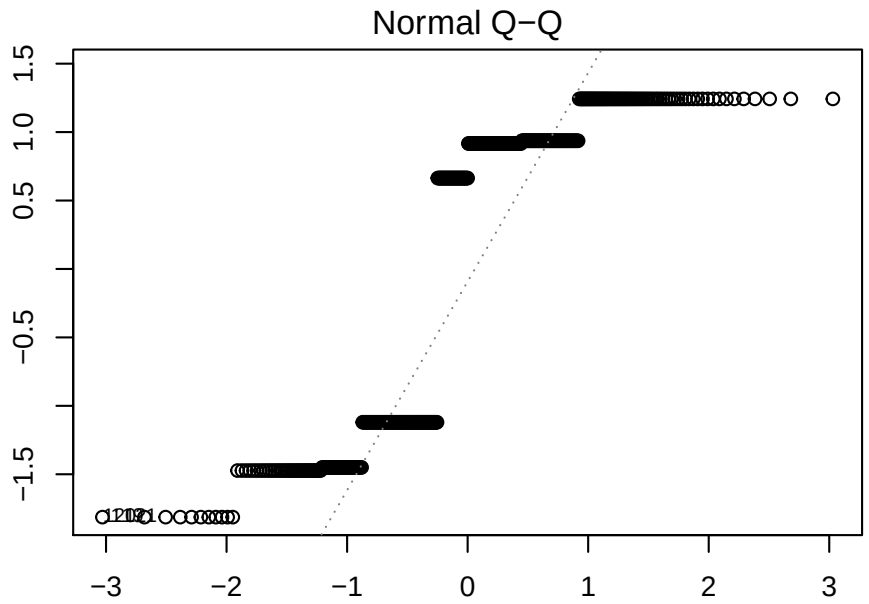
`anova()` returns a table with the decomposition of the deviance, starting from the empty model $\mathcal{M}_0$ and adding each explanatory in turn in the order in which they were specified in the call to `glm()`.

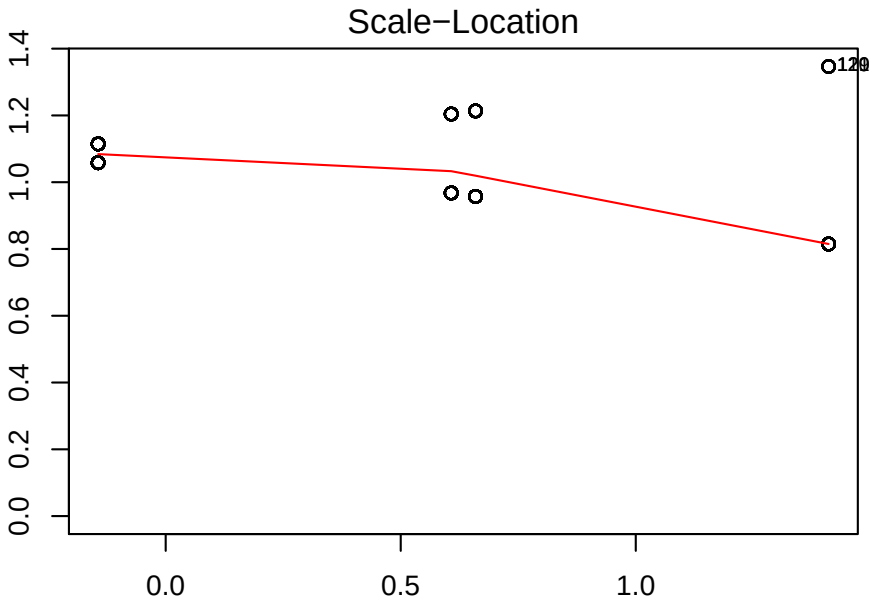
```
> anova(m)
Analysis of Deviance Table

Model: binomial, link: logit

Response: Species

Terms added sequentially (first to last)


         Df Deviance Resid. Df Resid. Dev
NULL                       408     550.85
Diameter  1   12.606        407     538.24
Height    1   11.674        406     526.57
```

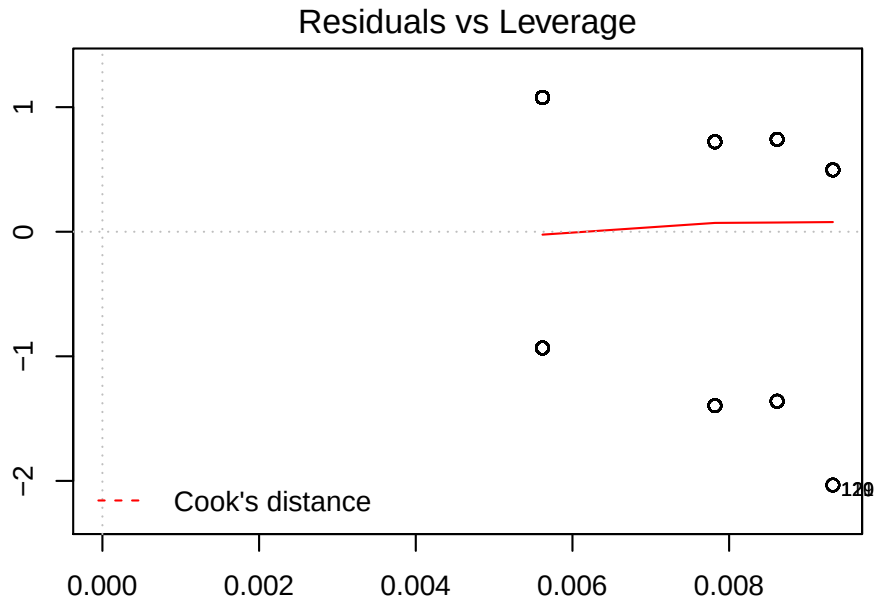
The first entry is the null deviance (i.e. the residual deviance of $\mathcal{M}_0$) and the last is the residual deviance (i.e. of $\mathcal{M}_L$) from `summary()`.

Graphical Diagnostics: `plot(m)`









Model Checking

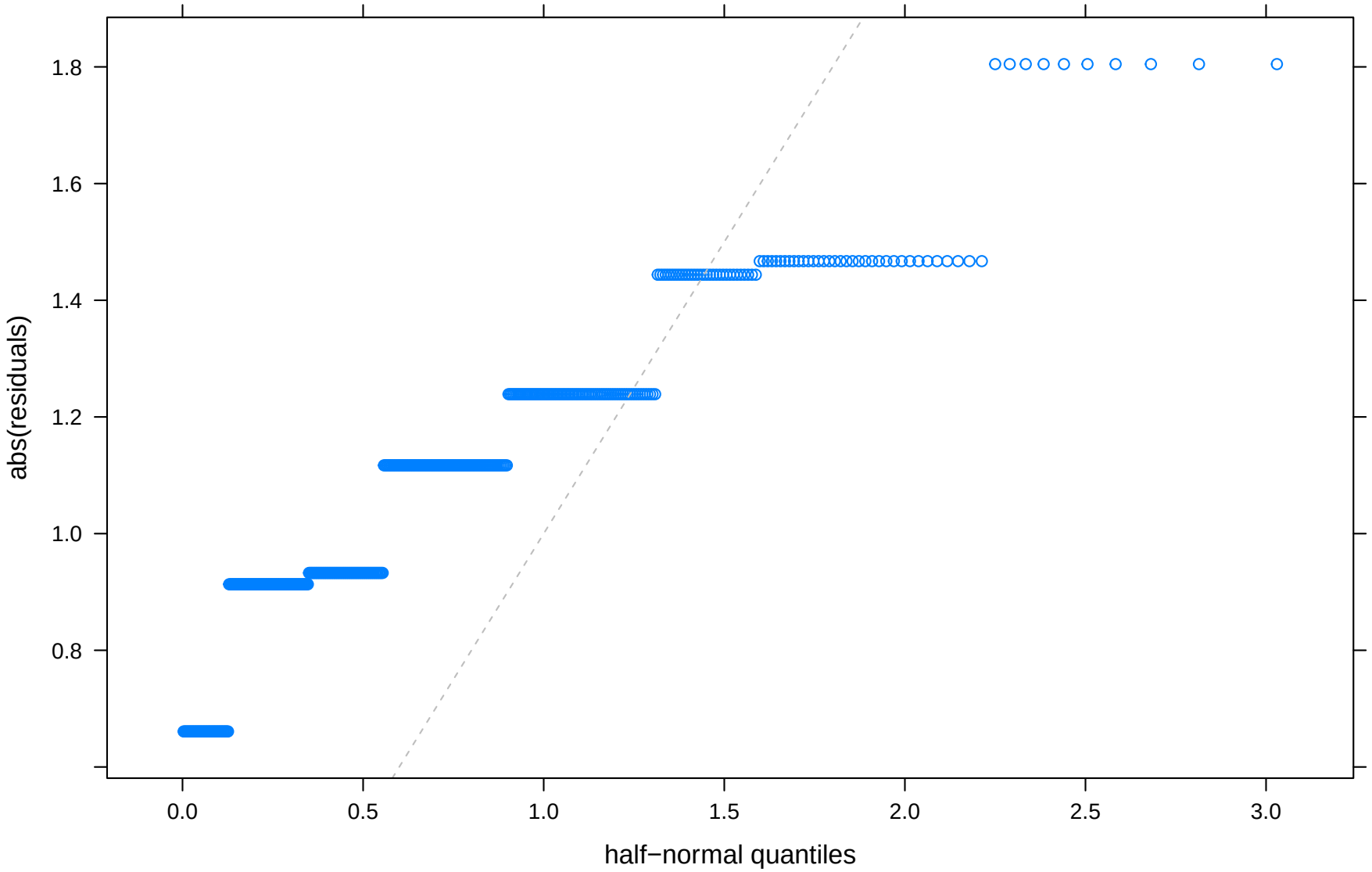
Checking whether the model is problematic is much more difficult for a GLM than for a classic linear model. Diagnostic plots can look very different depending on the nature of the response and of the explanatory variables, particularly the residuals' quantile-quantile plot.

An additional plot that may be useful to check for potential outliers is the half-normal quantile-quantile plot of the absolute residuals:

```
> library(lattice)
> n = nrow(lizards)
> xyplot(sort(abs(resid(m))) ~ qnorm((1:n + n)/(2 * n)),
+   xlab = "half-normal quantiles", ylab = "abs(residuals)",
+   panel = function(...) {
+     panel.xyplot(...)
+     panel.abline(0, 1, col = "grey", lty = 2)
+ })
```

which may be easier to read than the default two-tailed plot when the residuals are skewed.

Graphical Diagnostics: Half-Normal Plot



Leverage and Cook's Distance

The residual variance of a linear model is replaced the dispersion $a(\psi)$ and the hat matrix becomes

$$\mathbf{H} = \mathbf{W}^{\frac{1}{2}}\mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}^{\frac{1}{2}} \tag{52}$$

and in turn it can be shown that

$$\frac{\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}}{\sqrt{\mathrm{VAR}(\boldsymbol{\mu})}} \simeq \mathbf{H}\mathbf{V}^{\frac{1}{2}}(\mathbf{Y} - \boldsymbol{\mu}) \tag{53}$$

which means that $\mathbf{H}$ measures the influence of $\mathbf{Y}$ on $\boldsymbol{\mu}$ on the scale of Studentised residuals. So we can have the standardised residuals

$$r_P^* = \frac{(y_i - \mu_i)}{\sqrt{\hat{\psi}V(\hat{\mu}_i)(1 - h_{ii})}}, \qquad r_D^* = \frac{r_D}{\hat{\psi}(1 - h_{ii})} \tag{54}$$

and Cook's distance (which is still confusingly denoted as d_i or D_i) to identify outliers.

Logistic Regression

Model Formulation

Logistic regression is a binomial GLM with the canonical logit link

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{55}$$

which means that for each observation

$$\pi_i = \frac{\exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p)}{1 + \exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p)}. \tag{56}$$

The relationship is linear between the logarithm of the odds of success and the regressors. In other words, each regression coefficient represents the logarithm of the estimated change in the odds for a unit change of the corresponding explanatory variable.

Prostatic Cancer: an Epidemiological Study

This data set from Collett's "Binary Data Modelling" book describes an epidemiological study on the diagnosis of nodal involvement in prostatic cancer based on non-invasive methods. The study includes 53 patients and 5 explanatory variables:

- Age: the age of the patient, in years.
- Acid: the level of serum acid phosphate.
- X-ray: the result of x-ray examination, `positive` or `negative`.
- Size: tumour size, `small` or `large`.
- Grade: tumour grade, `less` or `more` serious.

```
> cancer = read.table("prostatic.cancer.txt", header = TRUE)
> head(cancer)
  Age Acid     Xray  Size Grade Nodal
1  66 0.48 negative small  less    no
2  68 0.56 negative small  less    no
3  66 0.50 negative small  less    no
4  56 0.52 negative small  less    no
5  58 0.50 negative small  less    no
6  60 0.49 negative small  less    no
```

The Importance of Factor Coding

The coding of the factors involved in the logistic regression is crucial in interpreting both model validation results and regression coefficients. In the case of the response variable, swapping cases and controls changes the signs of all the regression coefficients (and the intercept) because

$$\log\left(\frac{\pi}{1-\pi}\right) = -\log\left(\frac{1-\pi}{\pi}\right) = \log\left(\frac{\psi}{1-\psi}\right) \quad \text{with } \psi = 1-\pi. \tag{57}$$

The first level should correspond to the controls and the second to the cases.

As for the explanatory variables, contrasts are build using the first level as a reference so the regression coefficients may or may not be easily interpreted depending on which is chosen. We can take care of that with the `relevel()` function.

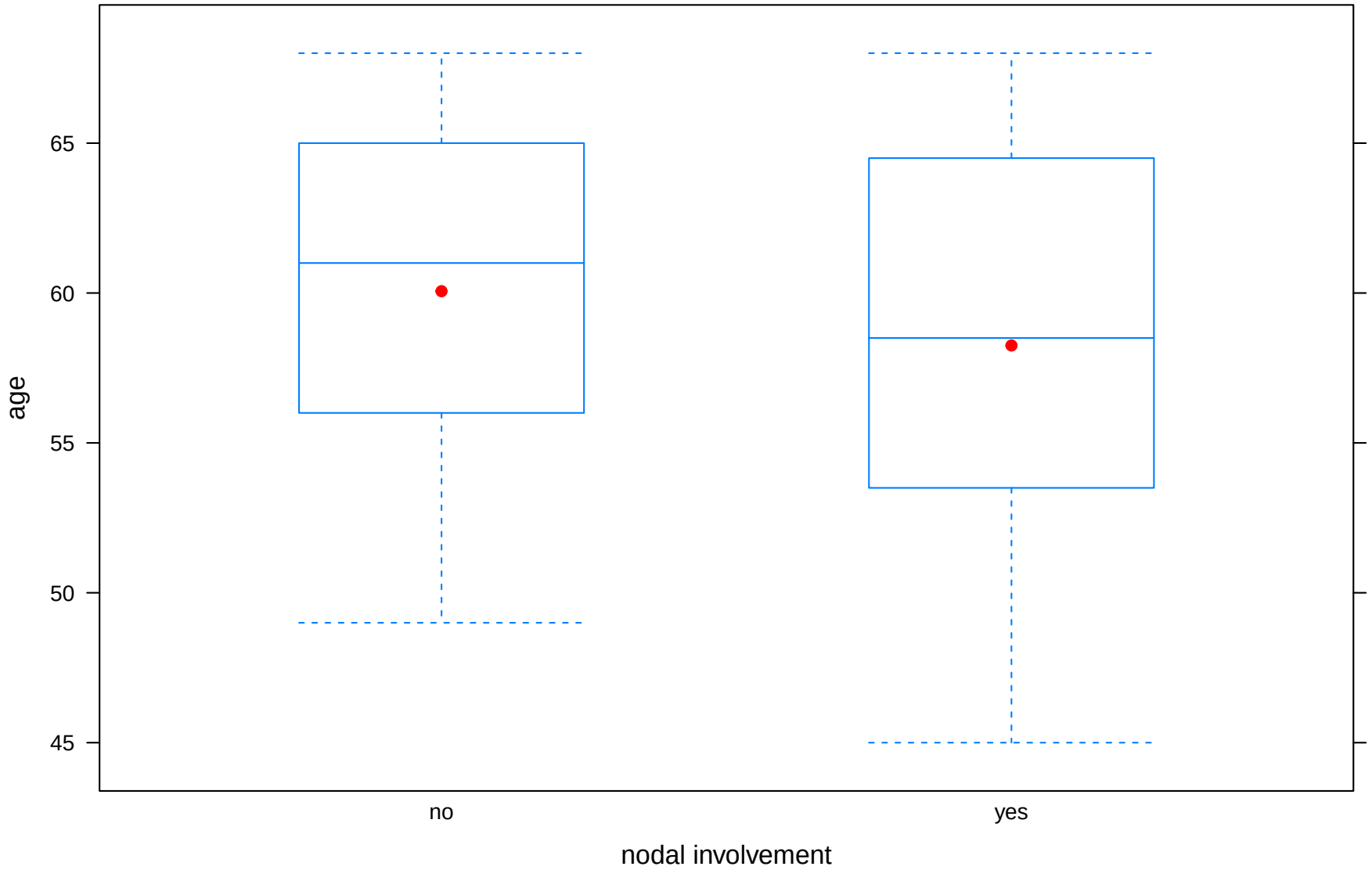
```
> cancer$Nodal = relevel(cancer$Nodal, ref = "no")
> cancer$Grade = relevel(cancer$Grade, ref = "less")
> cancer$Size = relevel(cancer$Size, ref = "small")
> cancer$Xray = relevel(cancer$Xray, ref = "negative")
```

Graphical Methods for Exploratory Analysis

Graphical exploratory analysis techniques developed for classic linear models are unsuited to logistic regression because they implicitly assume the response is continuous. Three alternatives are:

- Plotting each continuous explanatory variable against the response using boxplots.
- Plotting each categorical explanatory variable against the response using stacked barplots.
- Plotting the first principal component for the explanatory variables against the second in a principal components plot. Cases and controls are in different colours and should ideally cluster.

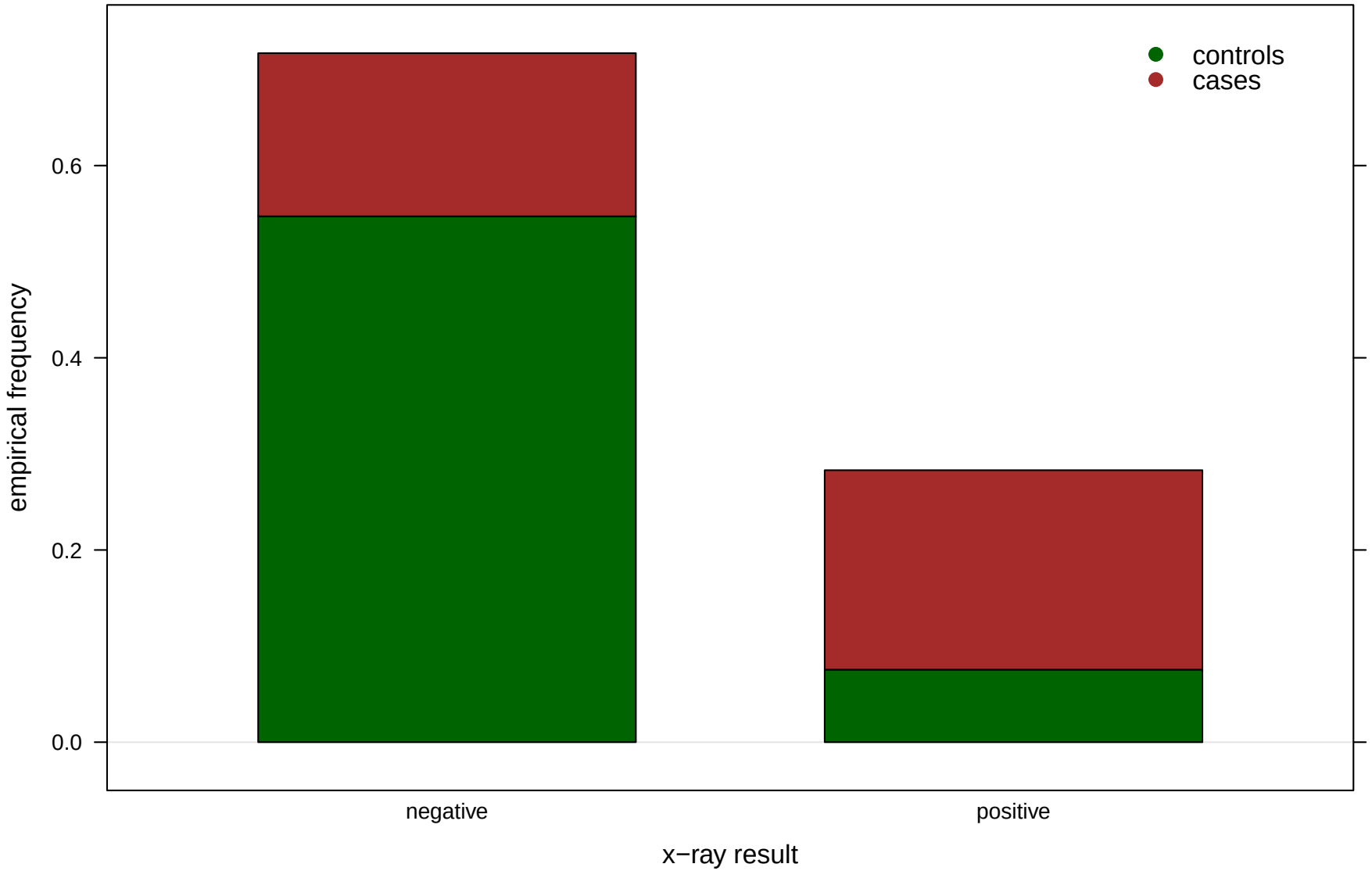
Boxplots for Continuous Explanatory Variables



Boxplots for Continuous Explanatory Variables (R Code)

```
> library(lattice)
> bwplot(Age ~ Nodal, data = cancer,
+   xlab = "nodal involvement", ylab = "age",
+   panel = function(x, y, ...) {
+
+     panel.bwplot(x, y, ..., pch = "|")
+     panel.points(1, mean(y[x == "no"]), pch = 19, col = "red", ...)
+     panel.points(2, mean(y[x == "yes"]), pch = 19, col = "red", ...)
+
+ })
```

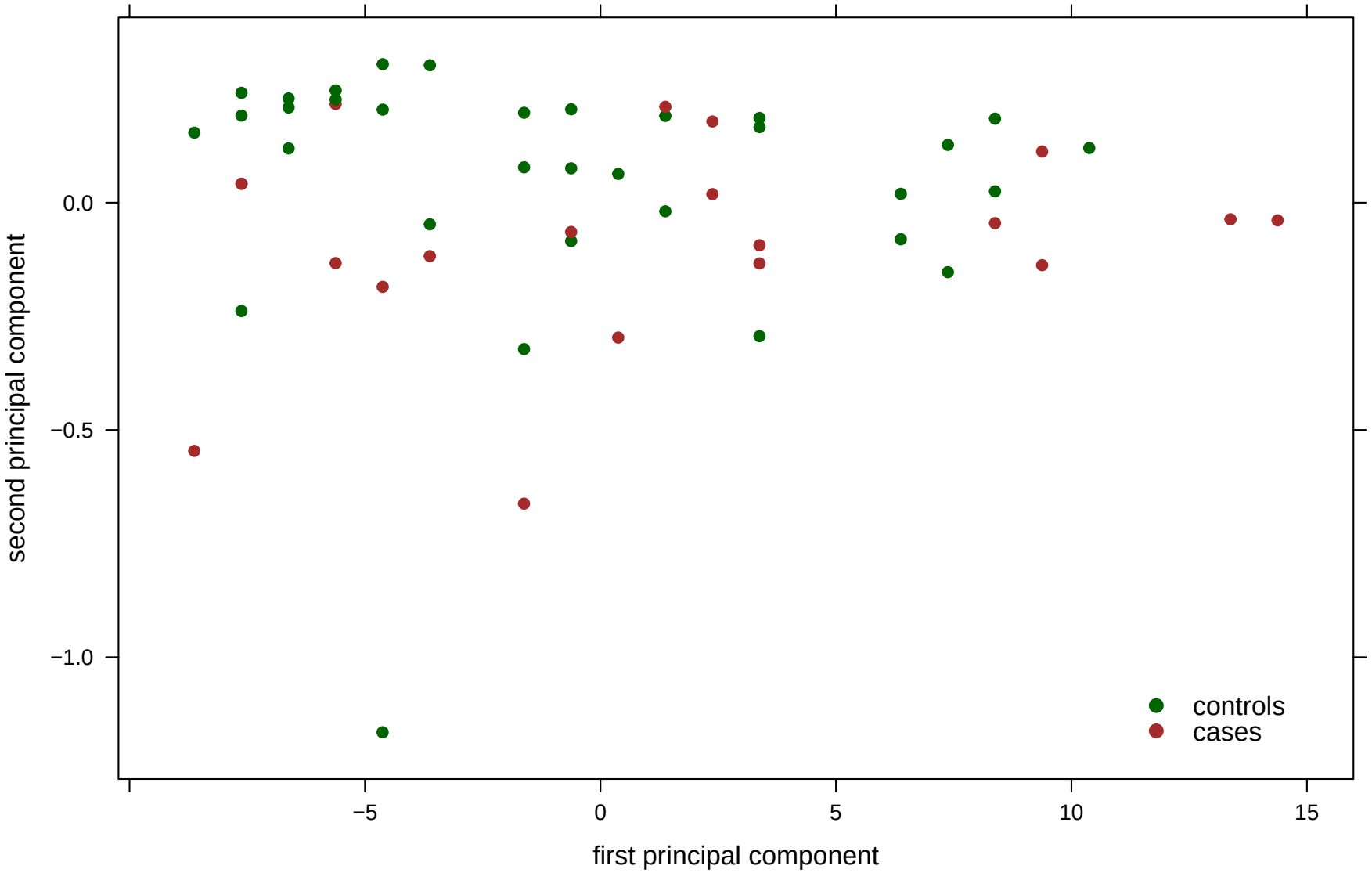
Stacked Barplots for Categorical Explanatory Variables



Stacked Barplots (R Code)

```
> tab = table(cancer[, c("Xray", "Nodal")])
>
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[c(3, 7)]
>
> barchart(prop.table(tab), horizontal = FALSE, col = col,
+   xlab = "x-ray result", ylab = "empirical frequency",
+   key = list(points = list(pch = c(19, 19), col = col),
+       corner = c(0.95, 0.95),
+       text = list(c("controls", "cases"))))
```

Principal Components Plot



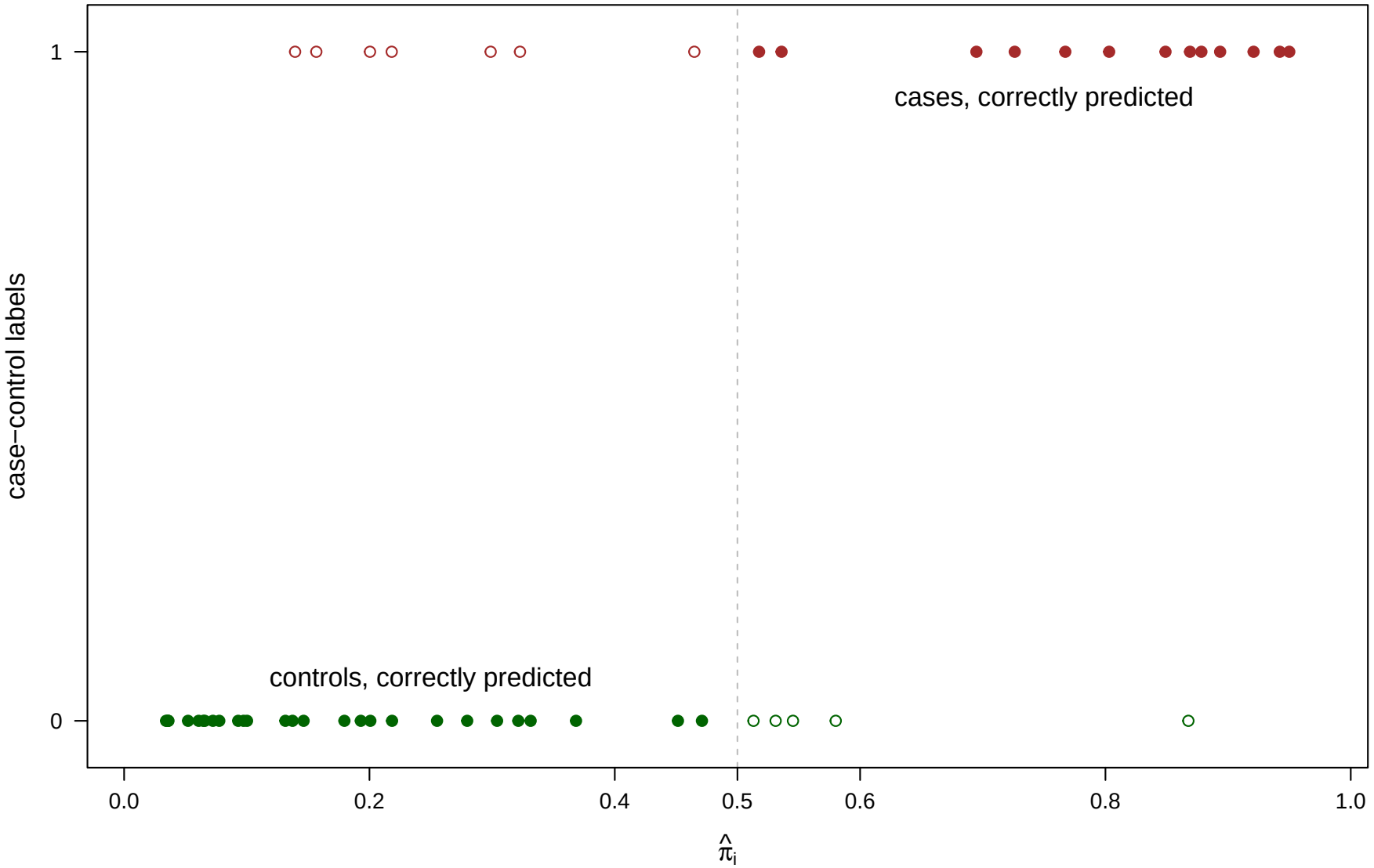
Principal Components Plot (R Code)

```
> pc = princomp(as.matrix(cancer[, c("Age", "Acid")]))$scores
>
> col = trellis.par.get()$superpose.symbol$col[c(3, 7)]
> names(col) = c("no", "yes")
>
> library(lattice)
> xyplot(Comp.2 ~ Comp.1, data = as.data.frame(pc),
+   col = col[cancer$Nodal], pch = 19,
+   xlab = "first principal component",
+   ylab = "second principal component",
+   key = list(points = list(pch = c(19, 19), col = col),
+           corner = c(0.95, 0.05),
+           text = list(c("controls", "cases"))))
```

Fitting the Logistic Regression Model

```
> m  = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = cancer,
+        family = binomial(link = "logit"))
> summary(m)
[...]
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0110  -0.7020  -0.3654   0.5723   1.9852

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.62590    3.45981   0.470   0.6384
Age          -0.06926    0.05788  -1.197   0.2314
Acid          2.43445    1.31583   1.850   0.0643 .
Xraypositive  2.04534    0.80718   2.534   0.0113 *
Sizesmall    -1.56410    0.77401  -2.021   0.0433 *
Grademore     0.76142    0.77077   0.988   0.3232
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 48.126  on 47  degrees of freedom
AIC: 60.126
```

Plotting Fitted and Observed Responses



Plotting Fitted and Observed Responses (R Code)

```
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[c(3, 7)]
>
> xyplot(as.numeric(cancer$Nodal) - 1 ~ fitted(m),
+   xlab = expression(hat(pi)[i]) , ylab = "case-control labels",
+   scales = list(x = list(at = c(0:5 * 0.2, 0.5)),
+                 y = list(at = c(0, 1)), tck = c(1, 0)),
+   panel = function(x, y, ...) {
+
+     panel.xyplot(x, y, col = col[y + 1],
+       pch = c(19, 1)[(((y == 0) & (x > 0.5)) | ((y == 1) & (x < 0.5))) + 1])
+     panel.abline(v = 0.5, col = "grey", lty = 2)
+     panel.text(x = 0.25, y = 0.1, pos = 1, "controls, correctly predicted")
+     panel.text(x = 0.75, y = 0.9, pos = 3, "cases, correctly predicted")
+
+ })
```

predict(m): Classify Cases and Controls

The information in the figure can be produced with `predict()`, either from the data used to fit the model `m` or from new data.

```
> PRED = ifelse(predict(m, type = "response") >= 0.5, "yes", "no")
> table(OBS = cancer$Nodal, PRED)
     PRED
OBS   no yes
  no  28   5
  yes  7  13
```

The four cells in the table above, which is called a confusion matrix, indicate how many observations are correctly identified as cases or controls by the model:

- Cases with a predicted $\hat{\pi}_i \geqslant 0.5$ are true positives (TP).
- Cases with $\hat{\pi}_i < 0.5$ are false negatives (FN).
- Controls with $\hat{\pi}_i \geqslant 0.5$ are false positives (FP).
- Controls with $\hat{\pi}_i < 0.5$ are true negatives (TN).

Accuracy, Sensitivity and Specificity

The goodness of fit and predictive ability of logistic regression (as well as other binary classification models) are measured using various functions of TP, TN, FP and FN. Say the number of cases is $\mathrm{P} = \mathrm{TP} + \mathrm{FN}$ and the number of controls is $\mathrm{N} = \mathrm{TN} + \mathrm{FP}$.

The first of these measures is the accuracy, the proportions of observations that are correctly classified:

$$\mathrm{ACCURACY} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\text{observations that are correctly classified}}{\text{sample size}}. \quad (58)$$

Then there are sensitivity,

$$\mathrm{SENSITIVITY} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\text{observations correctly classified as cases}}{\text{number of cases}}; \quad (59)$$

and specificity

$$\mathrm{SPECIFICITY} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\text{observations correctly classified as controls}}{\text{number of controls}}. \quad (60)$$

Precision and Recall

Another set of measures are precision

$$\text{PRECISION} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{observations correctly classified as cases}}{\text{observations classified as cases}} \tag{61}$$

and recall, which is another name for sensitivity.

To add to the confusion, sensitivity is also called the true positive rate (TPR) and specificity is also called the true negative rate (TNR). This naming convention is the same as in multiple testing adjustment, where we try to control the false positive rate (FPR) through the false discovery rate (FDR):

$$\text{FPR} = \frac{\text{FP}}{\text{N}} \quad \text{and} \quad \text{FDR} = \frac{\text{FP}}{\text{TP} + \text{FP}} = 1 - \text{PRECISION}. \tag{62}$$

Computing Them From the Confusion Matrix

So, in the confusion matrix we have

```
> tab = table(OBS = cancer$Nodal, PRED)
> TN = tab[OBS = "no", PRED = "no"]
> FN = tab[OBS = "yes", PRED = "no"]
> FP = tab[OBS = "no", PRED = "yes"]
> TP = tab[OBS = "yes", PRED = "yes"]
```

and then we can compute

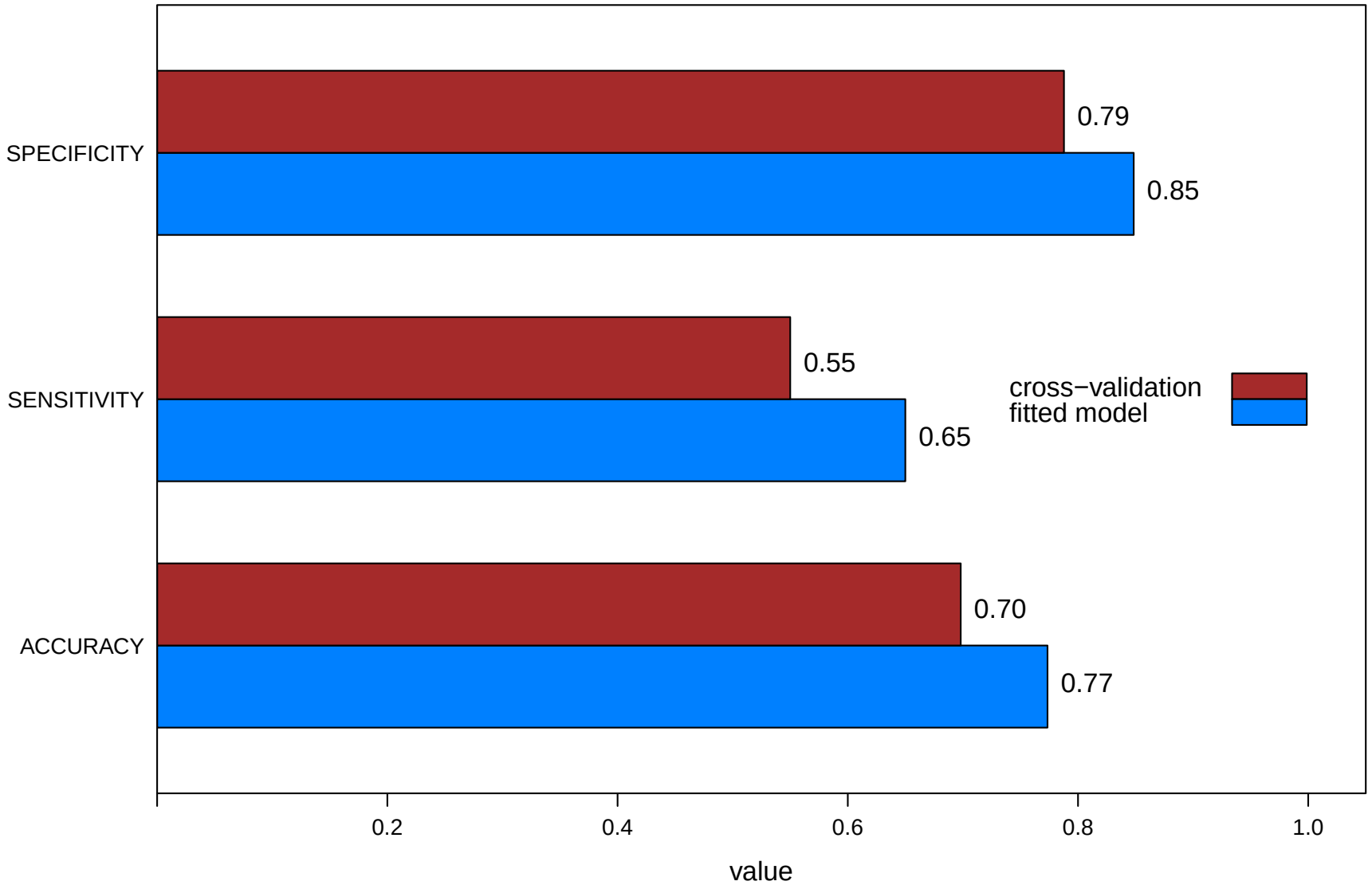
```
> accuracy = (TP + TN) / nrow(cancer)
> accuracy
[1] 0.7735849
> sensitivity = TP / (TP + FN)
> sensitivity
[1] 0.65
> specificity = TN / (TN + FP)
> specificity
[1] 0.8484848
```

All these measures are defined in $[0, 1]$, and high values are assigned to good models which fit or predict the data well.

Computing Them Using Cross-Validation

```
> # shuffle the data to get unbiased splits.
> kcv = split(sample(nrow(cancer)), seq_len(10))
>
> predicted = lapply(kcv, function(test) {
+
+   # split training and test.
+   dtraining = cancer[-test, ]
+   dtest = cancer[test, ]
+   # fit the model on the training data.
+   model = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = dtraining,
+             family = binomial(link = "logit"))
+   # predict the data in the test data.
+   PRED = ifelse(predict(model, newdata = dtest, type = "response") >= 0.5,
+             "yes", "no")
+   # return the observed-predicted pairs.
+   return(data.frame(OBS = dtest$Nodal, PRED = PRED))
+
+ })
>
> # collate all the predictions from the different folds.
> predicted = do.call("rbind", predicted)
> table(predicted)
```

Comparing Goodness of Fit and Predictive Power



Comparing Goodness of Fit and Predictive Power (R Code)

```
> d = data.frame(
+       MEASURE = rep(c("ACCURACY", "SENSITIVITY", "SPECIFICITY"), 2),
+       MODEL = c(rep("FITTED", 3), rep("XVAL", 3)),
+       VALUE = c(0.7735, 0.65,0.8484,0.6981,0.55,0.7878))
>
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[c(1, 7)]
>
> barchart(MEASURE ~ VALUE, group = MODEL, data = d,
+   xlim = c(0, 1.05), xlab = "value", scales = list(tck = c(1, 0)),
+   auto.key = list(corner = c(0.95, 0.5), points = FALSE, rectangles = TRUE,
+                   text = c("fitted model", "cross-validation"),
+                   reverse.rows = TRUE),
+   par.settings = simpleTheme(col = col),
+   panel = function(x, y, groups, ...) {
+
+     panel.barchart(x, y, groups = groups, ...)
+     panel.text(x = x,
+       y = as.numeric(y) + ((as.numeric(groups) - 1.5) * 2) * 0.15,
+       labels = sprintf("%.2f", x), pos = 4)
+
+ })
```

Predictive Power and The Bias-Variance Trade-Off

Logistic regression is susceptible to overfitting, same as classic linear models. Symptoms are markedly reduced values for sensitivity, specificity and accuracy in cross-validation compared to the model fitted on the whole data. Some caution is needed in reasoning on these quantities.

Some caution is needed in reasoning on these quantities. For example, note that when the sample is very unbalanced (i.e. very few cases compared to controls):

- specificity is inflated, because there are so many N that all models will have a high TN and thus $\mathrm{TN}/\mathrm{N} \to 1$;
- accuracy is similarly inflated, because TN dominates TP so $\mathrm{TN} + \mathrm{TP} \simeq \mathrm{TN}$ for all models;
- sensitivity loses discriminatory power, because TP/P is defined in increments of $1/\mathrm{P}$.

ROC Curves

A graphical, synthetic summary of such classification goodness-of-fit measure is the receiver operating characteristic (ROC) curve. Model performance is represented as a curve of sensitivity (the true positive rate) against $1 - \text{SPECIFICITY}$ (the false positive rate). The curve is bounded in $[0, 1] \times [0, 1]$, the ROC space:

- A perfect classification model would be in $(0, 1)$ because it has sensitivity 1 (no false negatives) and specificity 1 (no false positives).
- A model that is equivalent to a random guess would be on the diagonal.
- Models above the diagonal are good classifiers, models below are poor classifiers.

The curve is produced by varying the discrimination threshold that determines whether an observation is classified as a case or not; for logistic regression the default is $\hat{\pi}_i \geqslant 0.5$. This can be done either in the context of model fitting or in cross-validation.

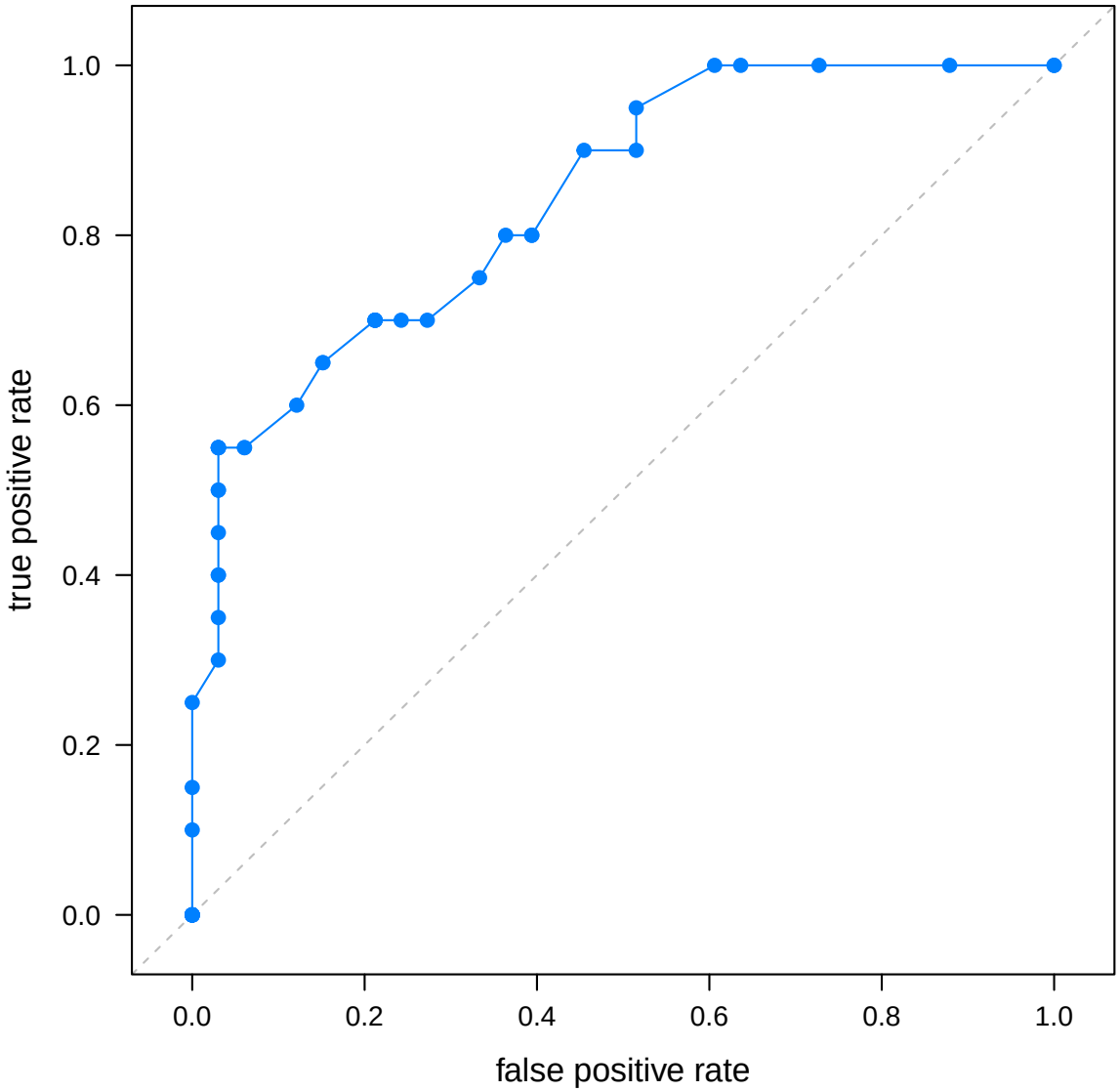
Building a ROC Curve (R Code)

Building a ROC curve on the whole data entails fitting the model once, building the confusion matrix and then varying the value of $\hat{\pi}_i$.

```
> m  = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = cancer,
+        family = binomial(link = "logit"))
> roc = data.frame(x = numeric(41), y = numeric(41))
> # 40 thresholds in 2.5% increments.
> thr = seq(from = 0, to = 1, by = 0.025)
> for (i in seq_along(thr)) {
+   PRED = ifelse(predict(m, type = "response") >= thr[i], "yes", "no")
+   PRED = factor(PRED, levels = c("no", "yes"))
+   tab = table(OBS = cancer$Nodal, PRED, useNA = "always")
+   # compute false positive rate.
+   roc[i, "x"] =  tab[OBS = "no", PRED = "yes"] / sum(tab[OBS = "no", ])
+   # compute true positive rate.
+   roc[i, "y"] =  tab[OBS = "yes", PRED = "yes"] / sum(tab[OBS = "yes", ])
+ }#FOR
```

Doing the same from cross-validated predictions works in the same way; unless multiple runs of cross-validation can be used to produce averaged coordinates for each `thr` for improved stability.

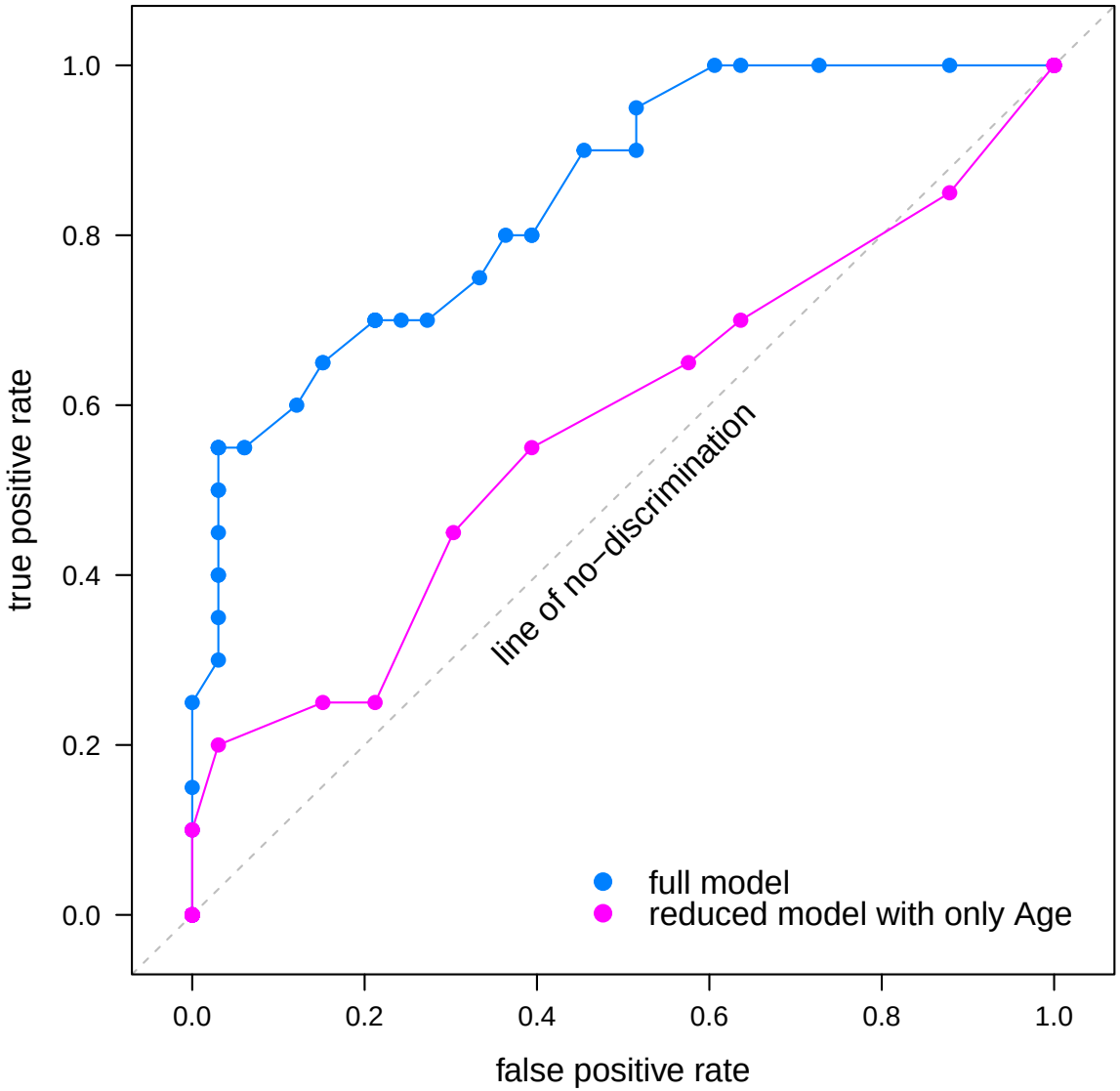
The ROC Curve for the Logistic Regression Model



Building and Interpreting a ROC Curve

- It is tricky to guess how many values of the threshold are needed to obtain a smooth-ish curve, because neither axis is a direct function of the threshold. This is important if the model takes time to fit and/or cross-validation is run many times.
- Models can be compared but it is unlikely any of them will strictly dominate the others over the whole ROC space. The closer a ROC curve is to the left and upper bounds of the ROC space, the better classifier is the corresponding model.
- All curves start in $(0, 0)$ and end up $(0, 1)$, and any realistic model for binary responses produces curves that are strictly above the diagonal.

Comparing ROC Curves



A Summary Statistic for ROC Curves

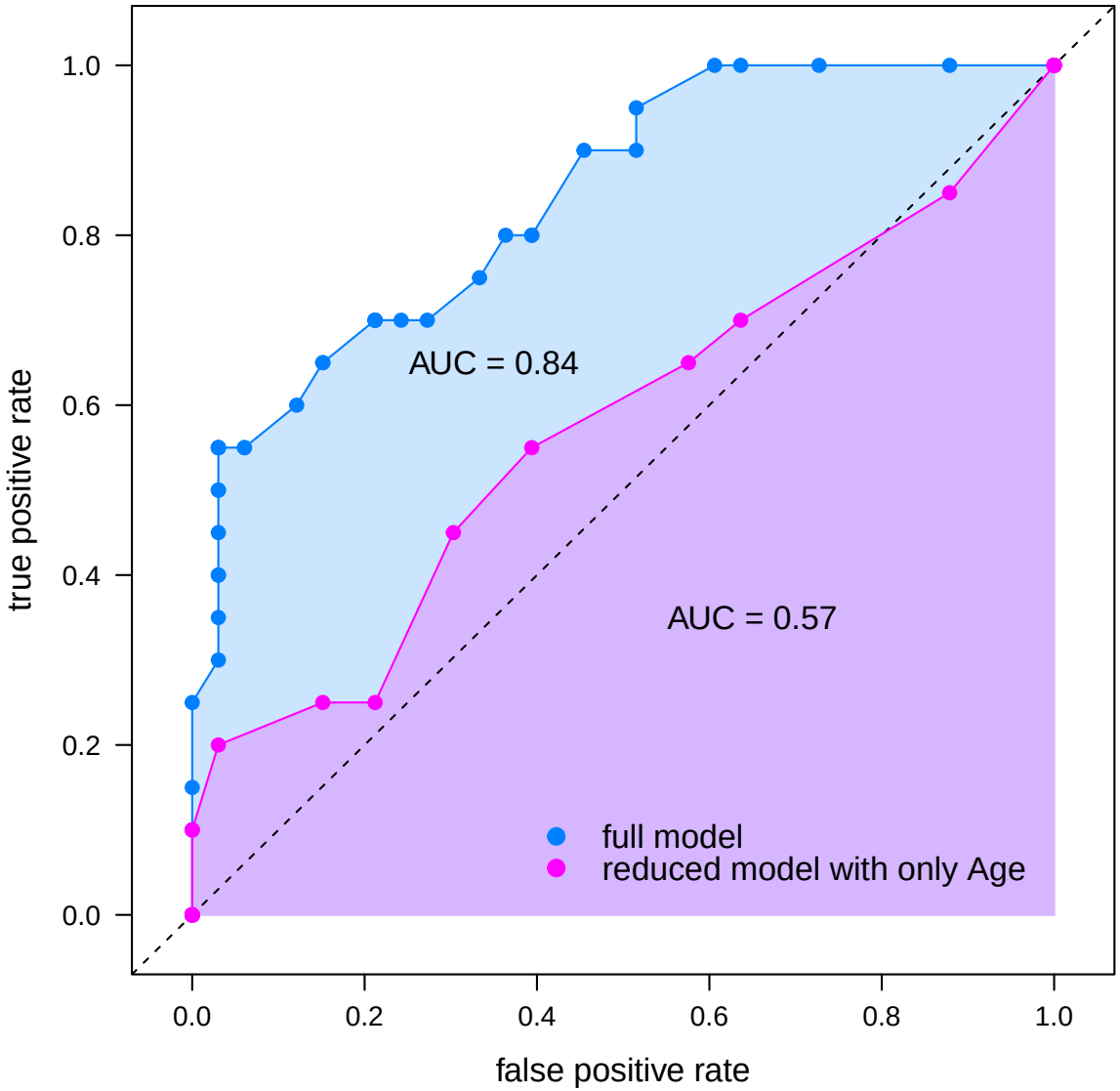
Clearly comparing models through their ROC curves is a principled approach, but it does not scale well to large number of models and it is ambiguous when the curves overlap and cross each other.

A popular summary statistic for a ROC curve is the area under the curve (AUC). If the curve is above the diagonal it ranges from 0.5 (e.g. the model does not perform any better than picking at random) and 1 (e.g. perfect classifier). An informal evaluation scale is:

from	**to**	**interpretation**
0	0.60	Bad
0.61	0.70	Acceptable
0.71	0.80	Good
0.81	1	Excellent

and 0.75 is a rough threshold for classification accuracy on cross-validated predictions.

Comparing AUC Values



Comparing AUC Values (R Code)

```
> m1  = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = cancer,
+        family = binomial(link = "logit"))
> m2  = glm(Nodal ~ Age, data = cancer, family = binomial(link = "logit"))
> roc = data.frame(x = numeric(82), y = numeric(82),
+         model = c(rep("M1", 41), rep("M2", 41)))
[...]
> xyplot(y ~ x, groups = model, data = roc, type = "b",
+   scales = list(tck = c(1, 0)),
+   xlab = "false positive rate", ylab = "true positive rate", pch = 19,
+   key = list(points = list(pch = 19, col = col), corner = c(0.85, 0.10),
+           text = list(c("full model", "reduced model with only Age"))),
+   panel = function(x, y, groups, ...) {
+
+     panel.polygon(x = c(1, x[groups == "M1"]), y = c(0, y[groups == "M1"]),
+       col = col[1], border = col[1], alpha = 0.2)
+     panel.polygon(x = c(1, x[groups == "M2"]), y = c(0, y[groups == "M2"]),
+       col = col[2], border = col[2], alpha = 0.2)
+     panel.abline(0, 1, lty = 2, col = "black")
+     panel.xyplot(x, y, groups, ...)
+     panel.text(x = 0.65, y = 0.35, "AUC = 0.57")
+     panel.text(x = 0.35, y = 0.65, "AUC = 0.84")
+
+ })
```

Ranking Models by AUC Values

The AUC for a ROC curve can be easily approximated using the trapezoid method, which is a one-liner from the `roc` data frame:

```
> r1 = roc[roc$model == "M1", ]
> r2 = roc[roc$model == "M2", ]
> sum(abs(r1$x[2:41] - r1$x[1:40]) * (r1$y[2:41] + r1$y[1:40]) / 2)
[1] 0.8424242
> sum(abs(r2$x[2:41] - r2$x[1:40]) * (r2$y[2:41] + r2$y[1:40]) / 2)
[1] 0.5742424
```

We can then rank the models by AUC and use it as we would use AIC or BIC to select the best classifier. Unlike deviance tests, models need not to be nested. As usual, AUC has to be computed under cross-validation to select the best predictive model.

Note, however, that telling whether two AUC values are significantly different is much of an open problem without an widespread, accepted solution.

predict(m): Interval Predictions

In addition to class labels, `predict(m)` also provides the fitted or predicted $\hat{\eta}_i$ complete with the respective standard errors.

```
> pred = predict.glm(m, newdata = cancer, se.fit = TRUE, type = "link")
```

They are useful to compute confidence intervals for:

- the odds for each observation,

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) \in \left[\hat{\eta}_i \pm z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}\right] \Rightarrow$$

$$\frac{\pi_i}{1-\pi_i} \in \left[e^{\hat{\eta}_i \pm z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}}\right]; \quad (63)$$

- and the corresponding probability of success (i.e. `yes` for the `cancer` data),

$$\pi_i \in \left[\frac{e^{\hat{\eta}_i - z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}}}{1+e^{\hat{\eta}_i - z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}}}, \frac{e^{\hat{\eta}_i + z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}}}{1+e^{\hat{\eta}_i + z_{1-\alpha/2}\sqrt{\text{VAR}(\hat{\eta}_i)}}}\right]. \quad (64)$$

Interpreting Odds: Point and Interval Estimates

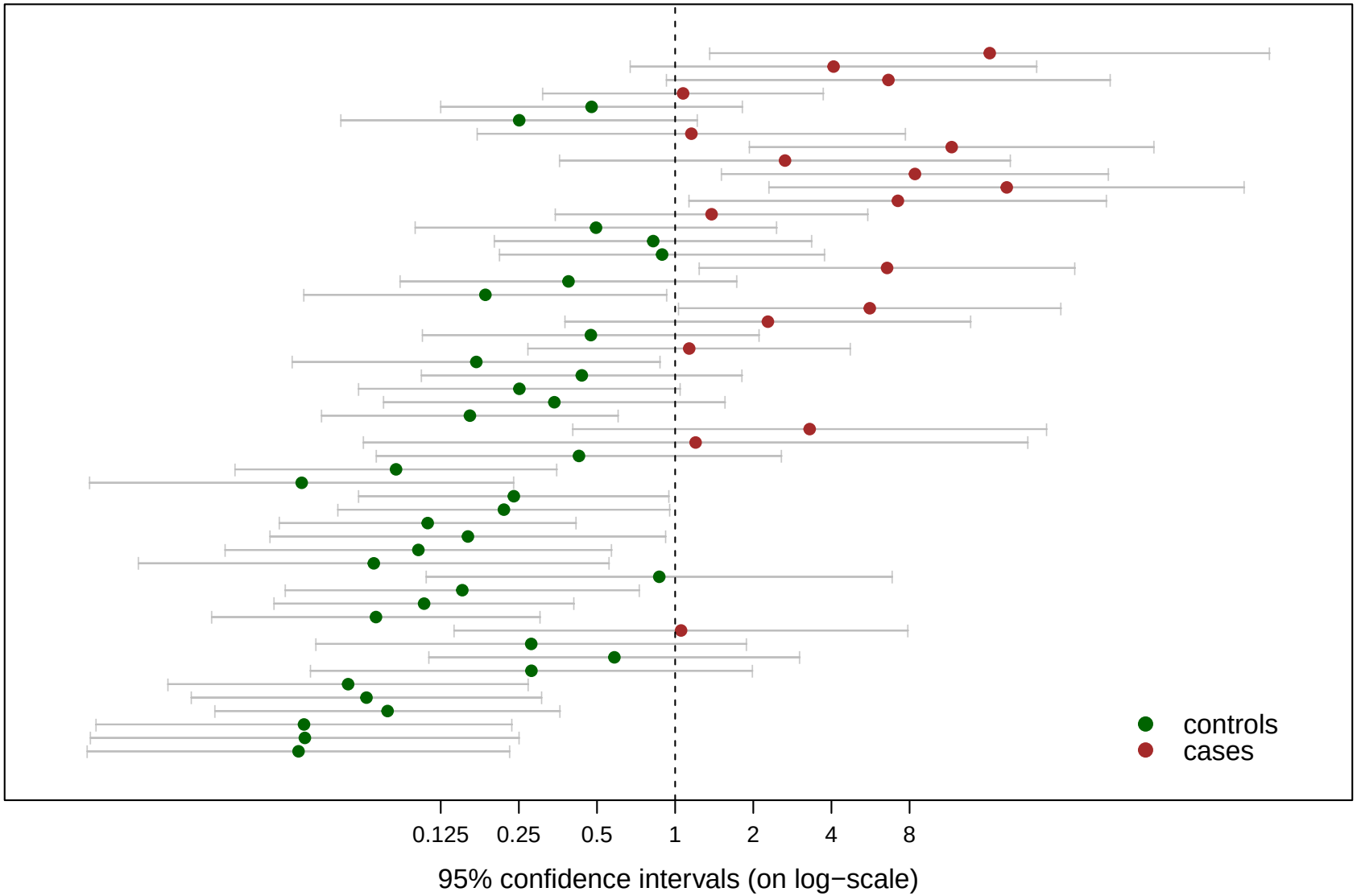
Approximate confidence intervals are easy to obtain from `pred`,

```
> fitted  exp(pred$fit)
> low = exp(pred$fit + qnorm(0.025) * pred$se.fit)
> high = exp(pred$fit + qnorm(0.975) * pred$se.fit)
```

and can be plotted in what is called an odds (ratio) plot. If zero falls within the confidence interval for a certain observation, then the odds are not significantly different from one, or equivalently $\hat{\pi}_i$ is not significantly different from 0.5. Given the long tradition of using logistic regression in epidemiological studies, practitioners have established conventional qualitative descriptions for odds and odds-ratios that look like this:

odds ratio	$\hat{\pi}_i$	susceptibility to disease
up to 0.3	up to 0.23	strong benefit (low risk of disease)
0.4 to 0.5	0.24 to 0.33	moderate benefit
0.6 to 0.8	0.34 to 0.44	weak benefit
0.9 to 1.1	0.45 to 0.52	no effect (no particular risk factors)
1.2 to 1.6	0.53 to 0.62	weak hazard
1.7 to 2.5	0.63 to 0.71	moderate hazard
2.6 or more	0.72 or more	strong hazard (high risk of disease)

Odds (Ratio) Plot



Odds (Ratio) Plot (R Code)

```
> fit = pred$fit
> low = pred$fit + qnorm(0.025) * pred$se.fit
> high = pred$fit + qnorm(0.975) * pred$se.fit
> col = trellis.par.get()$superpose.symbol$col[c(3, 7)]
> ticks = c(1/8, 1/4, 1/2, 1, 2, 4, 8)
> xyplot(seq(nrow(cancer)) ~ fit + low + high,
+   xlab = "95% confidence intervals (on log-scale)", ylab = "", col = col,
+   scales = list(x = list(at = log(ticks), label = as.character(ticks)),
+                 y = list(at = numeric(0)), tck = c(1, 0)),
+   key = list(points = list(pch = c(19, 19), col = col),
+            corner = c(0.95, 0.05), text = list(c("controls", "cases"))),
+   panel = function(x, y, groups, ...) {
+
+     panel.segments(x0 = x[groups == "low"], x1 = x[groups == "high"],
+       y0 = y, y1 = y, col = "grey")
+     panel.xyplot(x[groups == "fit"], y[groups == "fit"], pch = 19,
+       col = col[(x[groups == "fit"] > 0) + 1])
+     panel.xyplot(x[groups != "fit"], y[groups != "fit"], pch = "|",
+       col = "grey")
+     panel.abline(v = 0, col = "black", lty = 2)
+
+ })
```

Success Probabilities: Point and Interval Estimates

Success probabilities $\hat{\pi}_i$ and odds can be used interchangeably to investigate the same problems, because they are deterministic functions of each other:

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{\hat{\eta}_i} \qquad \Longleftrightarrow \qquad \hat{\pi}_i = \frac{e^{\hat{\eta}_i}}{1+e^{\hat{\eta}_i}}. \tag{65}$$

One may be more common than the other depending on the field. In plotting odds are more common, and often printed on a log-scale to make the confidence intervals symmetric. They are also used to compare the overall success probability for the same response across different models (on the same data) or across different data (with the same model).

A Note: The Hauck-Donner Phenomenon

Testing regression coefficients with Wald tests with deviance tests is more reliable than using the corresponding Wald tests due to a paradox called the Hauck - Donner phenomenon. What happens is that as the distance between the $\hat{\beta}_j$ and the null value increases, the test statistic t_{β_j} decreases to 0 (and so does z_{β_j}). This means, counter-intuitively, that we might fail to reject the null hypothesis because the effect of an explanatory variable is "too significant"!

This can happen when there is perfect or near-perfect separation of successes and failures in terms of an explanatory variable. Then the $\sqrt{\text{VAR}(\hat{\beta}_j)} \to \infty$ faster than $\hat{\beta}_j \to \infty$ which means that $t_{\beta_j} \to 0$.

Log-Linear Regression

Model Formulation

Log-linear regression is a Poisson GLM with the canonical logarithmic link

$$\log(\lambda_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{66}$$

which means that for each observation

$$\lambda_i = \exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p) \tag{67}$$

The relationship is linear between the logarithm of the intensity (i.e. the expected number of arrivals) and the regressors. In other words, each regression coefficient introduces a multiplicative contribution linked to changes in the corresponding explanatory variable:

$$\lambda_i = \exp(\beta_0 + [x_{i1} + 1]\beta_1) = \exp(\beta_0)\exp(x_{i1}\beta_1)\exp(\beta_1). \tag{68}$$

A Link Between Logistic and Log-Linear Models

If all the explanatory variables are categorical, the response can be summarised as a count for each of their configurations and then modelled as a Poisson random variable.

```
> library(doBy)
> cancer$Nodal = as.numeric(cancer$Nodal == "yes")
> counts = summaryBy(Nodal ~ Size + Grade + Xray, data = cancer,
+   FUN = c(sum, length))
> counts
   Size Grade     Xray Nodal.sum Nodal.length
1 large  less negative         4            9
2 large  less positive         3            3
3 large  more negative         2            8
4 large  more positive         6            7
5 small  less negative         1           17
6 small  less positive         1            4
7 small  more negative         2            4
8 small  more positive         1            1
> glm(Nodal.sum ~ Size + Grade + Xray, data = counts,
+   family = poisson(link = "log"), offset = log(Nodal.length))
```

(More on the `offset` argument later.)

Species in the Galapagos: an Example from Ecology

This data set from Ramsey & Schafer's "Statistical Sleuth" book describes the number of native and non-native species in relation to:

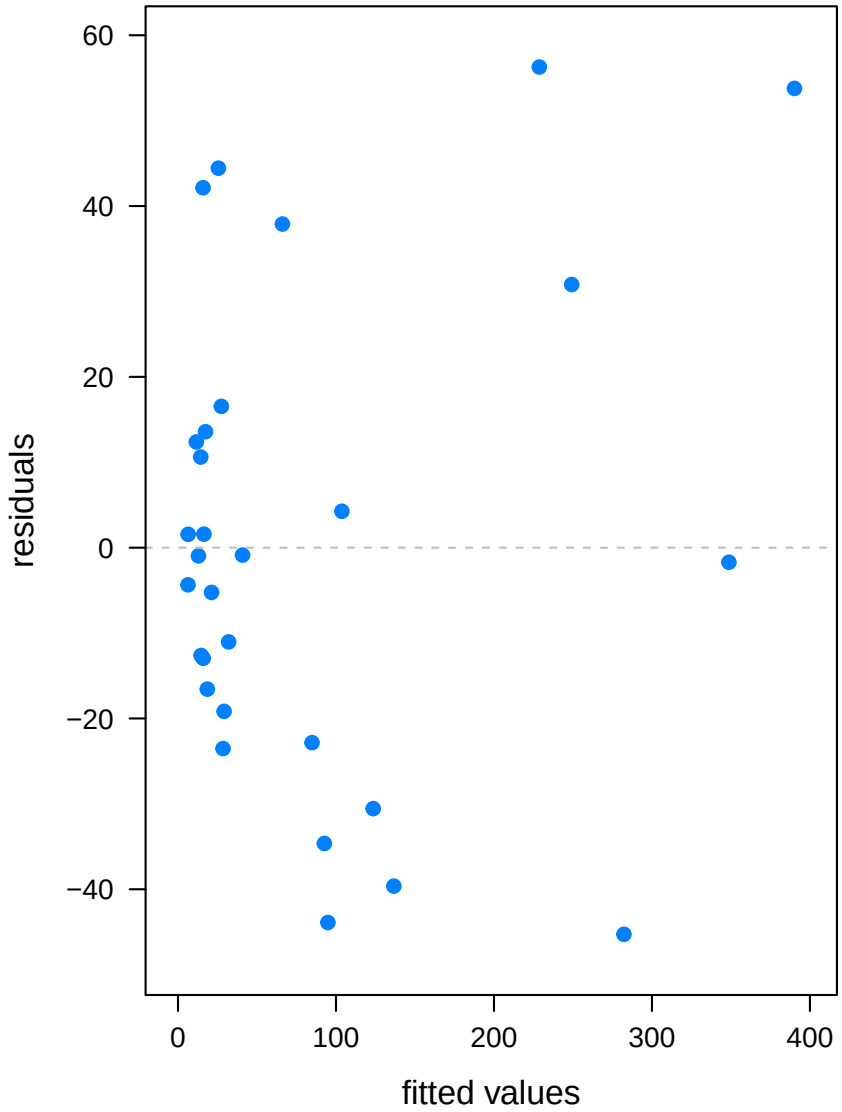
- Island: name of the island.
- Total number of species and number of native species.
- Area (km^2) and Elevation (m).
- Distance from the nearest island (DistNear) and from Santa Cruz (DistSc).
- AreaNear: area of the nearest island.

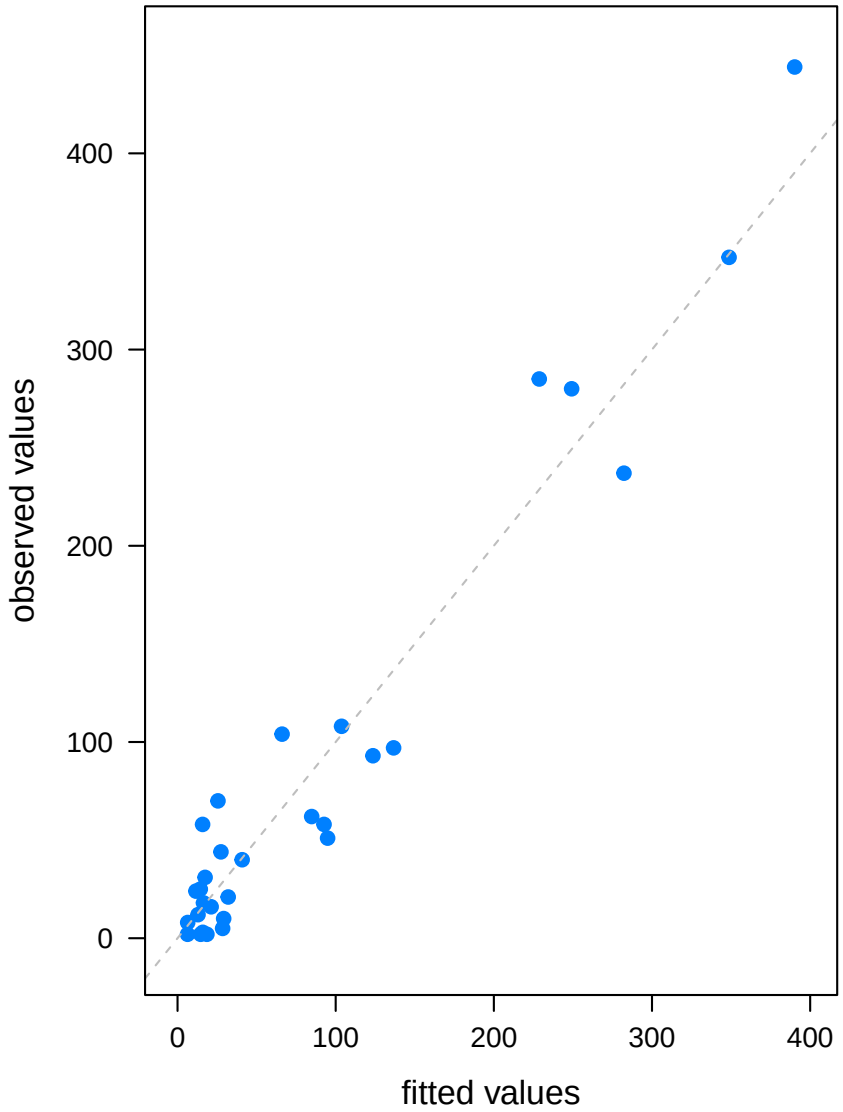
```
> galapagos = read.table("galapagos.txt", header = TRUE)
> head(galapagos)
        Island Total Native  Area Elev DistNear DistSc AreaNear
1       Baltra    58     23 25.09  332      0.6    0.6     1.84
2    Bartolome    31     21  1.24  109      0.6   26.3   572.33
3     Caldwell     3      3  0.21  114      2.8   58.7     0.78
4     Champion    25      9  0.10   46      1.9   47.4     0.18
5      Coamano     2      1  1.05  130      1.9    1.9   903.82
6 Daphne Major    18     11  0.34  119      8.0    8.0     1.84
```

Modelling the Total Number of Species

```
> m = glm(Total ~ 1, data = galapagos, family = poisson(link = "log"))
> m = step(m, scope = ~ log(Area) + log(Elev) + log(DistNear) + log(AreaNear))
Start:  AIC=3673.56; Total ~ 1
[...]
Step:  AIC=552.09;   Total ~ log(Area) + log(AreaNear) + log(DistNear)

                Df Deviance    AIC
<none>               383.3  552.1
+ log(Elev)      1    382.4  553.2
- log(DistNear)  1    409.8  576.7
- log(AreaNear)  1    656.8  823.6
- log(Area)      1   3320.2 3487.0

Call:  glm(formula = Total ~ log(Area) + log(AreaNear) + log(DistNear),
    family = poisson(link = "log"), data = galapagos)

Coefficients:
  (Intercept)      log(Area)  log(AreaNear)  log(DistNear)
      3.37832        0.36626       -0.09916       -0.05982

Degrees of Freedom: 29 Total (i.e. Null);  26 Residual
Null Deviance:      3511
Residual Deviance: 383.3        AIC: 552.1
```

Fitted Values and Residuals





Fitted Values and Residuals (R Code)

```
> library(lattice)
> library(gridExtra)
>
> p2 = xyplot(Total ~ pred, data = galapagos, pch = 19,
+         xlab = "fitted values", ylab = "observed values",
+         scales = list(tck = c(1, 0)),
+         panel = function(...) {
+           panel.xyplot(...)
+           panel.abline(0, 1, col = "grey", lty = 2)
+ })
>
> p1 = xyplot(Total - pred ~ pred, data = galapagos, pch = 19,
+         xlab = "fitted values", ylab = "residuals",
+         scales = list(tck = c(1, 0)),
+         panel = function(...) {
+           panel.xyplot(...)
+           panel.abline(h = 0, col = "grey", lty = 2)
+ })
>
> grid.arrange(p1, p2, ncol = 2)
```

Residuals Look Good But the Residual Deviance is Huge?

There is no apparent pattern in the residuals, which is contrary to expectations. This can happen when there are few or no duplicate values in the response, so each point in the residuals plots is essentially its own pattern.
Residuals look as if they came from a Gaussian GLM because the $\hat{\lambda}_i$ are large enough that $Pois(\lambda) \to N(\lambda, \lambda)$ but are spread out in such a way that heteroschedasticity does not show too much.
Then why is the residual deviance (383.26) is large compared to the residual degrees of freedom (26)? It is due to overdispersion:

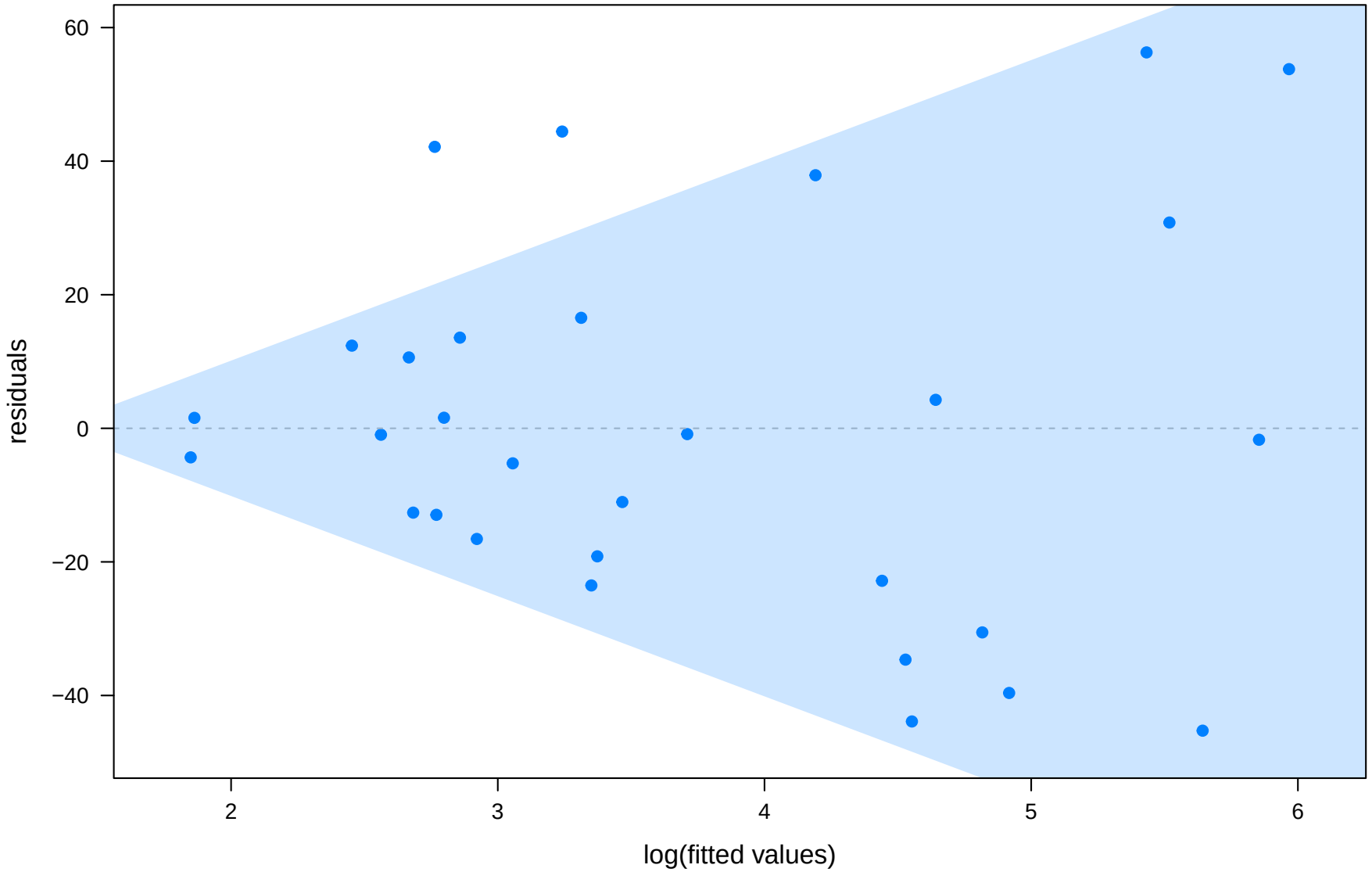
```
> psi = sum(residuals(m, type = "pearson")^2)/m$df.res
> psi
[1] 16.16604
```

Rescaling the residual deviance by the dispersion parameter $\hat{\psi}$ gives a value that is much closer to the residual degrees of freedom,

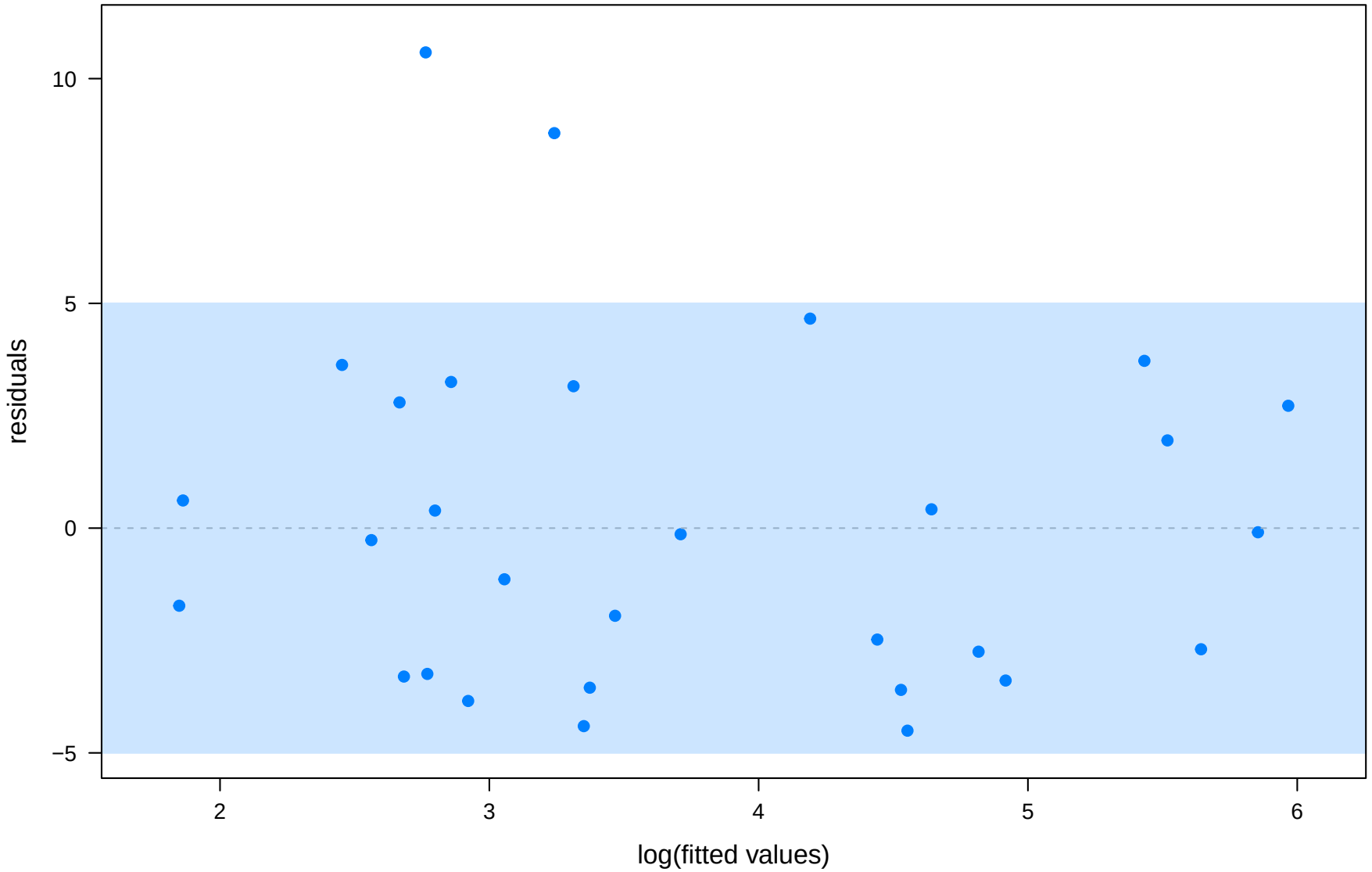
```
> 383.26 / 16.16
[1] 23.71658
```

which are the expected value for the underlying distribution, after all.

Yes, Raw Residuals Really Are Heteroschedastic



But Pearson's Residuals Are Not



Residuals Plots (R Code)

```
> pred = predict(m, type = "response")
> # raw residuals.
> xyplot(Total - pred ~ log(pred), data = galapagos, pch = 19,
+   xlab = "log(fitted values)", ylab = "residuals",
+   scales = list(tck = c(1, 0)),
+   panel = function(...) {
+     panel.xyplot(...)
+     panel.abline(h = 0, col = "grey", lty = 2)
+     panel.polygon(x = c(20/15, 7, 7), y = c(0, 85, -85),
+       col = col, border = col, alpha = 0.2)
+ })
> # scaled residuals.
> xyplot(residuals(m, type = "pearson") ~ log(pred), data = galapagos,
+   xlab = "log(fitted values)", ylab = "residuals", pch = 19,
+   scales = list(tck = c(1, 0)),
+   panel = function(...) {
+     panel.xyplot(...)
+     panel.abline(h = 0, col = "grey", lty = 2)
+     panel.polygon(x = c(0, 0, 7, 7), y = c(-5, 5, 5, -5),
+       col = col, border = col, alpha = 0.2)
+  })
```

Modelling Native Species: Offsets

Now, Poisson random variables are usually characterised as modelling the "number of events occurring in a fixed interval of time and/or space." This is not the case either for the `cancer` data (i.e. the number n_i of patients for each configuration of the explanatory variables is different) or for the `galapagos` data (i.e. each island has a different number n_i of species that may be or may not be native). To make up for this we can introduce $\beta_0^* = n_i \beta_0$ in

$$\log(\lambda_i) - \log(n_i) = \beta_0^* + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{69}$$

to have an offset that expresses the exposure e.g. the length of the time interval in which events were counted or the number of subjects in the study. It must be a known constant.

The Poisson GLM then models an set of observation-specific intensities,

$$\log\left(\frac{\lambda_i}{n_i}\right) = \beta_0^* + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p. \tag{70}$$

The Wrong Model, Without Offsets

```
> summary(glm(Native ~ log(Area) + log(AreaNear) + log(DistNear),
+   family = poisson(link = "log"), data = galapagos)
[...]

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4748  -1.6666   0.1745   0.6280   3.8165

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.59389    0.07240  35.826  < 2e-16 ***
log(Area)     0.27002    0.01255  21.521  < 2e-16 ***
log(AreaNear) -0.05023    0.01072  -4.684 2.82e-06 ***
log(DistNear) -0.06099    0.02102  -2.902  0.00371 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 700.717  on 29  degrees of freedom
Residual deviance:  96.448  on 26  degrees of freedom
AIC: 241.74
```

A Better Model, With Offsets

```
> summary(glm(Native ~ log(Area) + log(AreaNear) + log(DistNear),
+   family = poisson(link = "log"), data = galapagos, offset = log(Total)))
[...]

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6640  -0.4904   0.2651   0.9648   2.2015

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.83891    0.07245 -11.579  < 2e-16 ***
log(Area)    -0.08306    0.01207  -6.879 6.04e-12 ***
log(AreaNear)  0.03622    0.01045   3.467 0.000526 ***
log(DistNear)  0.01230    0.02103   0.585 0.558465
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 85.563  on 29  degrees of freedom
Residual deviance: 30.982  on 26  degrees of freedom
AIC: 176.27
```

Model Selection and Validation

The model can equivalently be specified putting the offset variable in the formula.

```
> summary(glm(Native ~ log(Area) + log(AreaNear) + log(DistNear) +
+   offset(log(Total)), family = poisson(link = "log"),
+   data = galapagos))
```

Then we can use stepwise selection to look for a better offsets model.

```
> m = glm(Native ~ 1 + offset(log(Total)), family = poisson(link = "log"),
+       data = galapagos)
> m = step(m, scope = ~ log(Area) + log(Elev) + log(DistNear) +
+       log(AreaNear) +  offset(log(Total)))
> formula(m)
Native ~ log(Area) + log(AreaNear) + offset(log(Total))
```

That model has a better AIC than the original non-offsets model, and on top of that it does not suffer from overdispersion (i.e. $\hat{\psi}$ is close to 1).

```
> AIC(m)
[1] 174.6107
> sum(residuals(m, type = "pearson")^2)/m$df.res
[1] 1.195945
```

Advanced Models

Quasi-Likelihood Models

From (33) we have that the likelihood equations for the β_j can be written as

$$\frac{dl(\beta_j)}{d\beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)} \frac{d\mu_i}{d\eta_i} x_{ij} = 0 \qquad \text{for all } j = 1, \ldots, p \qquad (71)$$

to emphasise that the GLM depends on the distribution assumed for the response only through μ_i and $V(\mu_i)$; and that assumption determines how the two are linked.
So, if we do without any distributional assumption and we switch to a semi-parametric model in which we only assume

$$\text{VAR}(Y_i) = V(\mu_i), \qquad (72)$$

we obtain a quasi-likelihood model that is more flexible but at the same time reverts back to a classic GLM if we add the distributional assumptions back (e.g. the $\hat{\beta}_j$ coincide with the maximum likelihood estimates). Here we assume implicitly that $a(\psi) = 1$.

Quasi-Binomial and Quasi-Poisson

In the Poisson GLM the variance is determined by

$$V(\mu_i) = \mu_i \tag{73}$$

but in a quasi-likelihood model it more convenient to have

$$V(\mu_i) = \psi\mu_i \tag{74}$$

to model overdispersion (typically, it assumed that $\psi \geqslant 1$). Similarly, in a binomial quasi-likelihood model we can assume

$$V(\mu_i) = \psi n\pi_i(1 - \pi_i). \tag{75}$$

In both cases ψ drops out the likelihood equations, much like the residual variance does in Gaussian regression models, so model estimation is identical to that of Poisson and binomial GLMs. An estimated $\hat{\psi}$ can be plugged in afterwards for the purpose of rescaling deviance.

A Quasi-Poisson Model for the Galapagos

The syntax of `glm()` has `family = quasipoisson` instead of `poisson`, and the AIC is `NA` because the likelihood is undefined. The deviance is not rescaled by $\hat{\psi} = 1.1959$ in the output, we must do that by hand.

```
>  summary(glm(Native ~ log(Area) + log(AreaNear) + offset(log(Total)),
+    family = quasipoisson(link = "log"), data = galapagos))
[...]

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.82113    0.07158 -11.471 6.92e-12 ***
log(Area)    -0.08428    0.01295  -6.507 5.63e-07 ***
log(AreaNear)  0.03570    0.01140   3.132  0.00415 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasipoisson family taken to be 1.195946)

    Null deviance: 85.563  on 29  degrees of freedom
Residual deviance: 31.323  on 27  degrees of freedom
AIC: NA
```

A Quasi-Binomial Model for Prostatic Cancer

It is actually possible to have a $\hat{\psi} < 1$, but in general it is implicitly expected that $\hat{\psi} \geqslant 1$. Here $\hat{\psi} = 1$ for practical purposes.

```
> summary(glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = cancer,
+   family = quasibinomial(link = "logit")))
[...]

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.62590    3.45209   0.471   0.6398
Age          -0.06926    0.05775  -1.199   0.2364
Acid          2.43445    1.31289   1.854   0.0700 .
Xraypositive  2.04534    0.80538   2.540   0.0145 *
Sizesmall    -1.56410    0.77228  -2.025   0.0485 *
Grademore     0.76142    0.76905   0.990   0.3272
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasibinomial family taken to be 0.9955437)

    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 48.126  on 47  degrees of freedom
AIC: NA
```

Penalised Generalised Linear Models

Penalised regression in its classic form was introduced for linear models as penalised least squares optimisation, e.g.

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda_2 \sum_{i=0}^{p} \beta_i^2 \right\}, \qquad \lambda_2 \geqslant 0; \tag{76}$$

but in that context it can be equivalently formulated as a penalised maximum (log-)likelihood problem. In that latter form it may also be applied to GLMs as e.g.

$$\underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{n} l(g(\mathbf{x}_i\boldsymbol{\beta}), y_i) - \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}, \qquad \lambda_2 \geqslant 0 \tag{77}$$

to produce the ridge, lasso and elastic net equivalents of logistic and log-linear regression models.

Gene Association Study: Crohn's Disease

The data used in the unassessed practical are part of a case-control study on Crohn's disease from the ENCODE project (ENCyclopedia Of DNA Elements) comprising 101 individuals (59 cases, 42 controls). For each of them we have a marker profile containing the expression of 5 candidate genes.

The original study contains 13K gene expressions, which makes it impossible to fit logistic regression in its basic form. Suitable models are:

- a logistic ridge regression (next slide);
- a random-effects model (more on that in the hierarchical model course);
- a collection of single-gene logistic regressions to obtain the $\hat{\beta}_j$, which are then treated as independent and used jointly (ugly but sometimes the data are so big there is no other sensible way).

Penalised GLMs with `penalized()` and `glmnet()`

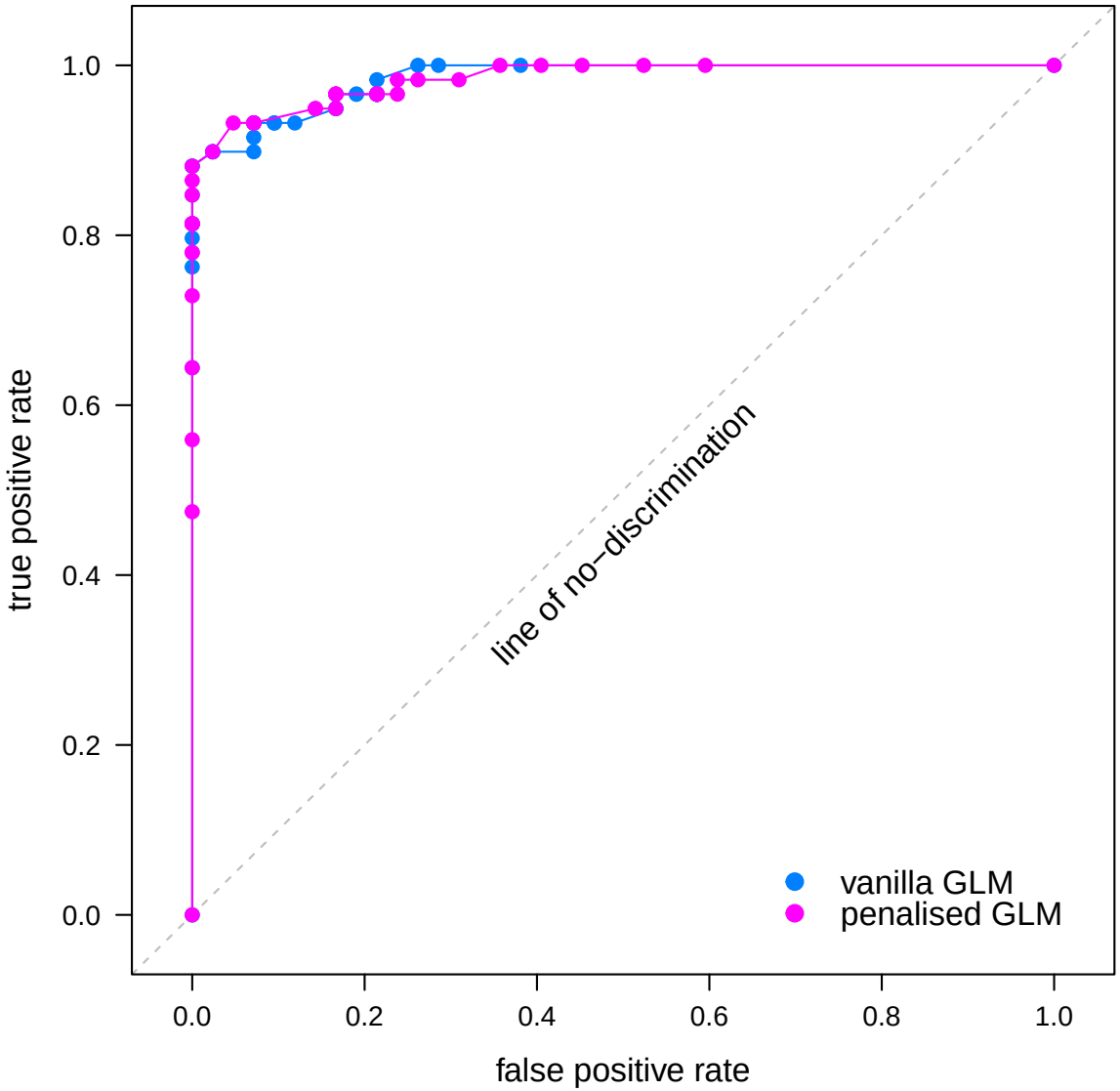
The `penalized` and `glmnet` packages can be used to fit such models using the same functions as for classic linear models. For instance:

```
> opt.lambda2 = optL2(response = expr$CASE,
+                     penalized = expr[, c("G1", "G2", "G3", "G4", "G5")],
+                     model = "logistic")
> penalized(response = expr$CASE,
+   penalized = expr[, c("G1", "G2", "G3", "G4", "G5")],
+   model = "logistic", lambda2 = opt.lambda2$lambda)
Penalized logistic regression object
6 regression coefficients

Loglikelihood =  -18.2756
L2 penalty =     3.282452       at lambda2 =  175.3674
```

An equivalent model can be fitted with `glmnet()` using the `family = "binomial"` argument.

For log-linear regression, use `model = "poisson"` in `penalized()` and `family = "poisson"` in `glmnet()`.

ROC Curves for Penalised and Vanilla GLMs



Bayesian GLMs: the Zero-Inflated Poisson Model

Suppose you have a response variable representing counts, which would ideally be well fitted by a Poisson GLM. However, the data contain an abnormal number of $y_i = 0$ due to the nature of the underlying phenomenon.

To account for this we can re-formulate the model as

$$Y_i \sim \begin{cases} 0 & \text{with probability } \pi \\ Pois(\lambda) & \text{with probability } 1 - \pi. \end{cases} \tag{78}$$

or equivalently

$$\mathrm{P}(Y_i = 0) = \pi + (1 - \pi)e^{-\lambda}, \quad \mathrm{P}(Y_i = y_i) = (1 - \pi)\frac{\lambda^{y_i} e^{-\lambda}}{y_i!}. \tag{79}$$

This is known as zero-inflated Poisson (ZIP) model.

Mixture Model, or Bayesian Model?

From a frequentist point of view, this model is a mixture model combining a Poisson distributions and a Dirichlet mass at zero. Each observation is generated by one of these two distributions (the components of the mixture), but we do not which. Therefore, model estimation is treated as a missing data problem in which there is an unobserved auxiliary dummy variable encoding which that missing information.

From a Bayesian point of view, we have a Poisson likelihood and we assign a prior distribution to λ with π as hyperparameter:

$$f(\lambda; \pi) = 0 \cdot \pi + \lambda \cdot (1 - \pi). \tag{80}$$

That is a spike-and-slab prior, because is combines a diffuse probability distribution (the $Pois(\lambda)$) with a point probability mass (the spike at zero).

Estimates for λ and π

The methods of moments gives closed form estimates for both λ and π:

$$\hat{\lambda}_{\text{MO}} = \frac{s^2 + \bar{y}^2 - \bar{y}}{\bar{y}} \qquad \text{and} \qquad \hat{\pi}_{\text{MO}} = \frac{s^2 - \bar{y}}{s^2 + \bar{y}^2 - \bar{y}} \tag{81}$$

where m and s^2 are the sample mean and variance.

The maximum likelihood solution is not in closed form, but λ can be estimated numerically by solving

$$\bar{y}(1 - e^{-\lambda}) = \lambda \left(1 - \frac{n_0}{n}\right) \tag{82}$$

where n_0 is the number of observed zeros; and then that estimate can be plugged in

$$\hat{\pi}_{\text{ML}} = 1 - \frac{\bar{y}}{\hat{\lambda}_{\text{ML}}}. \tag{83}$$

to estimate π.

Zero-Inflated Poisson Regression

In the context of regression, the ZIP model takes the form

$$\lambda_i = 0 \cdot \pi + \exp(x_{i1}\beta_1 + \ldots + x_{ip}\beta_p) \cdot (1 - \pi_i) \tag{84}$$

with

$$\log(\lambda_i) = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{85}$$

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \gamma_0 + z_{i1}\gamma_1 + \ldots + z_{iq}\gamma_q \tag{86}$$

where the explanatory variables $x_{i1}, \ldots, x_{ip}$ for λ_i may or many not be the same as (or overlap with) the explanatory variables $z_{i1}, \ldots, z_{ip}$ for π_i. Estimation is performed as a missing data problem with the expectation-maximisation (EM) algorithm.

Articles Produced by Ph.D. Students

This data set comprises 915 biochemistry Ph.D. students and was collected to investigate differences in the career paths between men and women:

- art: articles published in last 3 years of Ph.D.;
- fem: gender of the student, `Men` or `Women`;
- mar: marital status of the student, `Single` or `Married`;
- kid5: number of children aged 5 or younger;
- phd: prestige of the department;
- ment articles published by the student's mentor in the last 3 years.

```
> library(pscl)
> data(bioChemists)
> str(bioChemists)
'data.frame': 915 obs. of  6 variables:
 $ art : int  0 0 0 0 0 0 0 0 0 0 ...
 $ fem : Factor w/ 2 levels "Men","Women": 1 2 2 1 2 2 2 1 1 2 ...
 $ mar : Factor w/ 2 levels "Single","Married": 2 1 1 2 1 2 1 2 1 2 ...
 $ kid5: num  0 0 0 1 0 2 0 2 0 0 ...
 $ phd : num  2.52 2.05 3.75 1.18 3.75 ...
 $ ment: int  7 6 6 3 26 2 3 4 6 0 ...
```

Bad: The Plain Poisson GLM Model

```
> summary(glm(art ~ ., data = bioChemists, family = poisson))
[...]

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.304617   0.102981   2.958   0.0031 **
femWomen    -0.224594   0.054613  -4.112 3.92e-05 ***
marMarried   0.155243   0.061374   2.529   0.0114 *
kid5        -0.184883   0.040127  -4.607 4.08e-06 ***
phd          0.012823   0.026397   0.486   0.6271
ment         0.025543   0.002006  12.733  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1817.4  on 914  degrees of freedom
Residual deviance: 1634.4  on 909  degrees of freedom
AIC: 3314.1
```

Nearly as Bad: A Quasi-Likelihood Model

```
> summary(glm(art ~ ., data = bioChemists, family = quasipoisson))
[...]

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.304617   0.139273   2.187 0.028983 *
femWomen    -0.224594   0.073860  -3.041 0.002427 **
marMarried   0.155243   0.083003   1.870 0.061759 .
kid5        -0.184883   0.054268  -3.407 0.000686 ***
phd          0.012823   0.035700   0.359 0.719544
ment         0.025543   0.002713   9.415  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for quasipoisson family taken to be 1.829006)

    Null deviance: 1817.4  on 914  degrees of freedom
Residual deviance: 1634.4  on 909  degrees of freedom
AIC: NA
```

Better: The Zero-Inflated Poisson Model

```
> summary(zeroinfl(art ~ . | ment, data = bioChemists))
[...]

Count model coefficients (poisson with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.630168   0.113099   5.572 2.52e-08 ***
femWomen    -0.218471   0.058793  -3.716 0.000202 ***
marMarried   0.133420   0.066170   2.016 0.043768 *
kid5        -0.162958   0.043371  -3.757 0.000172 ***
phd         -0.006518   0.028533  -0.228 0.819312
ment         0.018298   0.002261   8.093 5.84e-16 ***

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.6837     0.2053  -3.331 0.000866 ***
ment         -0.1303     0.0402  -3.240 0.001194 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 16
Log-likelihood: -1606 on 8 Df (so AIC = 3244)
```

Another Model for Overdispersion: the Beta-Binomial

The beta-binomial model is the frequentist take on the classic Bayesian conjugate model

$$f(y \mid \pi) \sim Bi(n, \pi_i) \qquad \text{with} \qquad \pi \sim Beta(\alpha, \beta) \tag{87}$$

resulting in the compound model

$$f(y \mid n, \alpha, \beta) = \binom{y}{n} \frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{88}$$

after integrating out π. The expected value and variance are

$$\mathrm{E}(Y) = \frac{n\alpha}{\alpha+\beta} = n\pi, \tag{89}$$

$$\mathrm{VAR}(Y) = \frac{n\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)} = n\pi(1-\pi)\frac{\alpha+\beta+n}{\alpha+\beta+1}. \tag{90}$$

The Dispersion Parameter in the Beta-Binomial

The variance of the beta-binomial can be re-written as

$$\mathrm{VAR}(Y) = n\pi(1-\pi)\frac{\alpha+\beta+n}{\alpha+\beta+1} = n\pi(1-\pi)[1+(n-1)\rho]. \quad (91)$$

where ρ is effectively an over-dispersion parameter equal to

$$\rho = \frac{1}{\alpha+\beta+1}. \quad (92)$$

- If $\rho \to 0$ then $\alpha \to +\infty$ or $\beta \to +\infty$, and there is no overdispersion because $\mathrm{VAR}(Y) = n\pi(1-\pi)$. The beta-binomial simplifies into a plain binomial random variable.
- If $\rho \to 1$ is large because $\alpha \to 0$ and $\beta \to 0$, then $\mathrm{VAR}(Y) \gg n\pi(1-\pi)$.

Note that it is not possible to model under-dispersion in this way.

Beta-Binomial as a Mixture Model

The beta-binomial model can also be seen as a random-effects or a mixture model combining multiple Bernoulli trials.

- For example, if $\alpha = \beta = 1$ then $Beta(\alpha, \beta) = U(0, 1)$ and all Bernoulli trials have random π_i from a non-informative prior distribution.
- At the other end of the spectrum, if

$$\frac{1}{\alpha + \beta} \to 0 \tag{93}$$

then the variance of the Beta distribution converges to zero and so all Bernoulli trials have the same exact π.

Beta-Binomial and Quasi-Likelihood

It is possible to use a logistic link function to write beta-binomial regression model akin to binomial GLM,

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{94}$$

and estimate the α and β hyperparameters through π_i and the dispersion parameter ψ. However, even though the model can be fitted maximising the likelihood numerically, it does not have all the favourable properties of GLMs because the beta-binomial distribution is not part of the exponential family.

For this reason the quasi-likelihood approach is preferred, as it has a similar formulation but more useful theoretical properties.

That's All Folks!