Advanced Statistical Modelling III (second term)

Department of Mathematical Sciences at Durham University

2024-03-14

Contents

General Information

- These are the lecture notes for the second term of the module MATH3411 – Advanced Statistical Modelling III of Durham University's degree for Mathematics and Statistics.
- **Acknowledgements**: This material is based on the lecture notes in previous modules taught by Dr Samuel Jackson, Dr Jochen Einbeck, Dr Ric Crossman, Dr Emmanuel Ogundimu, Dr Cuong Nguyen, Dr Ian Jermyn, Dr Louis Aslett, and Dr Reza Drikvandi.

Chapter 1

Review of Generalised Linear Models

Definition. A GLM is specified through the following components:

- A *linear predictor*: $\eta = \boldsymbol{\beta}^T \boldsymbol{x}$.
- An *injective response function* h, such that $\mu = \mathrm{E}[Y|\boldsymbol{x}, \boldsymbol{\beta}] = h(\eta) = h(\boldsymbol{\beta}^T \boldsymbol{x})$. Equivalently, one can write $g(\mu) = \boldsymbol{\beta}^T \boldsymbol{x}$, where $g = h^{-1}$ is the *link* function.
- The *distributional assumption*: $P(Y|\boldsymbol{x}, \boldsymbol{\beta})$ is an EDF, that is:

  $$P(y|\boldsymbol{x}, \boldsymbol{\beta}) = P(y|\theta(\boldsymbol{x}, \boldsymbol{\beta}), \phi(\boldsymbol{x}, \boldsymbol{\beta})) = \exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi)\right) \tag{1.1}$$

  Thus, the mean and variance of this distribution are:

  $$\begin{aligned} \mathrm{E}[Y|\theta, \phi] &= \mu = b'(\theta) && (1.2) \\ \mathrm{Var}[Y|\theta, \phi] &= \phi\, b''(\theta) = \phi\, b''((b')^{-1}(\mu)) = \phi\, \mathcal{V}(\mu) && (1.3) \end{aligned}$$

- We also assume *independent data*, that is:

  $$P(\{y_i\} \,|\, \{\boldsymbol{x}_i\}, \boldsymbol{\beta}) = \prod_{i=1}^{n} P(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}) \tag{1.4}$$

  where $\{y_i, i = 1, ..., n\}$ are response data given the $\{\boldsymbol{x}_i, i = 1, ..., n\}$.

The Natural/Canonical Link. Recall that we have both:

$$\begin{aligned} \mu &= \mathrm{E}[Y|\theta, \phi] \ = b'(\theta) && (1.5) \\ \mu &= \mathrm{E}[Y|\boldsymbol{x}, \boldsymbol{\beta}] = h(\boldsymbol{\beta}^T \boldsymbol{x}) = h(\eta) && (1.6) \end{aligned}$$

with Equation (1.5) holding as a result of $P(y|\theta, \phi)$ following an EDF distribution, and Equation (1.6) holding by definition for a GLM.

The *natural link* is the choice $h = b'$, or equivalently $g = (b')^{-1}$, resulting in the equation

$$\theta = \boldsymbol{\beta}^T \boldsymbol{x} = \eta. \tag{1.7}$$

Chapter 2

Estimation

2.1 Likelihood

Consider the grouped data setup where we have predictors and data with possible replicates $\{(\boldsymbol{x}_i, y_{ir_i})\}_{i\in[1..n],r_i\in[1..m_i]}$. Recall that under a GLM, given predictors $\{\boldsymbol{x}_i\}_{i\in[1..n]}$, each response y_{ir_i} is independent of the other y_{jr_j}, and of the values of all predictors $\boldsymbol{x}_j$ with $j \neq i$, so that the joint probability of the data — that is, the likelihood — is given by

$$L(\boldsymbol{\beta}) = P(\{y_{ir_i}\} \,|\, \{\boldsymbol{x}_i\}, \boldsymbol{\beta}) = P(\{y_{ir_i}\} \,|\, \{\theta_i\}, \phi) = \prod_{i=1}^{n} \prod_{r_i=1}^{m_i} P(y_{ir_i}|\theta_i, \phi) \tag{2.1}$$

where

$$P(y_{ir_i}|\theta_i, \phi) = \exp\left(\frac{y_{ir_i}\theta_i - b(\theta_i)}{\phi} + c(y_{ir_i}, \phi)\right) \tag{2.2}$$

with

$$\theta_i = (b')^{-1}(\mu_i) = (b')^{-1}(h(\eta_i)) = (b')^{-1}(h(\boldsymbol{\beta}^T\boldsymbol{x}_i)) \tag{2.3}$$

2.2 Log-Likelihood

The log probability of the data — or log-likelihood — is thus given by

$$\begin{align} l(\boldsymbol{\beta}) &= \log P(\{y_{ir_i}\} \,|\, \{\theta_i\}, \phi) \tag{2.4}\\ &= \sum_i \sum_{r_i} \left(\frac{y_{ir_i}\theta_i - b(\theta_i)}{\phi} + c(y_{ir_i}, \phi)\right) \tag{2.5}\\ &= \sum_i \left(m_i \frac{y_i\theta_i - b(\theta_i)}{\phi} + \sum_{r_i} c(y_{ir_i}, \phi)\right) \tag{2.6}\\ &= \sum_i l_i \tag{2.7} \end{align}$$

where

$$l_i = \frac{y_i\theta_i - b(\theta_i)}{\phi_i} + \sum_{r_i} c(y_{ir_i}, \phi) \tag{2.8}$$

$$\phi_i = \phi / m_i \tag{2.9}$$

and

$$y_i = \frac{1}{m_i} \sum_{r_i} y_{ir_i} \tag{2.10}$$

2.3 Score Function and Equation

The *score function* is given by

$$\boldsymbol{S}(\boldsymbol{\beta}) = \frac{\partial l}{\partial \boldsymbol{\beta}^T} = \sum_i \frac{\partial l_i}{\partial \boldsymbol{\beta}^T} = \sum_i \frac{\partial l_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \boldsymbol{\beta}^T} \tag{2.11}$$

where, recalling that $\mu_i = b'(\theta_i)$, $\mu_i = h(\eta_i)$ and $\eta_i = \boldsymbol{\beta}^T \boldsymbol{x}_i$, we have[1]:

$$\frac{\partial l_i}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{\phi_i} = \frac{y_i - \mu_i}{\phi_i} \tag{2.12}$$

$$\frac{\partial \theta_i}{\partial \mu_i} = \frac{\partial (b')^{-1}}{\partial \mu_i} = \frac{1}{b''((b')^{-1}(\mu_i))} = \frac{1}{\mathcal{V}(\mu_i)} = \frac{1}{b''(\theta_i)} \tag{2.13}$$

$$\frac{\partial \mu_i}{\partial \eta_i} = h'(\eta_i) \tag{2.14}$$

$$\frac{\partial \eta_i}{\partial \boldsymbol{\beta}^T} = \boldsymbol{x}_i \tag{2.15}$$

The score function is thus given by

$$\boldsymbol{S}(\boldsymbol{\beta}) = \sum_i \left(\frac{y_i - \mu_i}{\phi_i} \right) \left(\frac{1}{\mathcal{V}(\mu_i)} \right) h'(\eta_i)\, \boldsymbol{x}_i \tag{2.16}$$

$$= \frac{1}{\phi} \sum_i m_i (y_i - \mu_i) \frac{1}{\mathcal{V}(\mu_i)} h'(\eta_i)\, \boldsymbol{x}_i \tag{2.17}$$

The maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ must then satisfy the *score equation*:

$$\boldsymbol{S}(\hat{\boldsymbol{\beta}}) = 0 \tag{2.18}$$

Note that the dispersion parameter ϕ cancels from the score equation, which implies that $\hat{\boldsymbol{\beta}}$ does not depend on ϕ. This is another important property of EDFs.

2.3.1 Natural Link

For the natural link, $\theta_i = \eta_i$, so Equations (2.13) and (2.14) combine to give

$$\frac{h'(\eta_i)}{\mathcal{V}(\mu_i)} = \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \eta_i} = \frac{\partial \theta_i}{\partial \eta_i} = 1 \tag{2.19}$$

The score function thus simplifies to

$$\boldsymbol{S}(\boldsymbol{\beta}) = \frac{1}{\phi} \sum_i m_i (y_i - \mu_i)\, \boldsymbol{x}_i \tag{2.20}$$

[1] for the second line, we use the fact that if $f(f^{-1}(z)) = z$, then $f'(f^{-1}(z))(f^{-1})'(z) = 1$, thus $(f^{-1})'(z) = \frac{1}{f'(f^{-1}(z))}$ (and take $f = b'$).

2.4 Fisher Information

For future developments, we will also need the second derivative of the log likelihood. Up to a *change of sign*, this is called the *Observed Fisher Information*, defined as

$$\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}) = -\frac{\partial^2 l}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} = -\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\beta}} \tag{2.21}$$

Note that, at the MLE, $\boldsymbol{F}_{\text{obs}}(\hat{\boldsymbol{\beta}})$ is positive by definition. Because it is a function of the data $\{y_i\}$, $\boldsymbol{F}_{\text{obs}}$ has a probability distribution. In practice, the Observed Fisher Information is often approximated by the *Expected Fisher Information*[2], otherwise known simply as the *Fisher Information*:

$$\boldsymbol{F}(\boldsymbol{\beta}) = E\left[-\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\beta}}\right] \tag{2.22}$$

where the expectation is taken over the joint probability distribution of the data $P(\{y_{ir_i}\} | \boldsymbol{\beta}, \{\boldsymbol{x}_i\})$.

2.5 Example: Poisson Regression

We look at two example calculations of the score function and Fisher Information for Poisson Regression, that is we have

- $y|\boldsymbol{x}, \boldsymbol{\beta} \sim \text{Poi}(\lambda(\boldsymbol{x}))$
- $\phi = 1$

2.5.1 With Natural Link

We have that

- $\lambda(\boldsymbol{x}) = \mu(\boldsymbol{x}) = h(\eta(\boldsymbol{x})) = e^{\eta(\boldsymbol{x})} = e^{\boldsymbol{\beta}^T \boldsymbol{x}}$

Equation (2.20) then gives:

$$\boldsymbol{S}(\boldsymbol{\beta}) = \sum_i (y_i - e^{\boldsymbol{\beta}^T \boldsymbol{x}_i})\, \boldsymbol{x}_i \tag{2.23}$$

while Equation (2.21) gives

$$\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}) = \sum_i e^{\boldsymbol{\beta}^T \boldsymbol{x}_i}\, \boldsymbol{x}_i \boldsymbol{x}_i^T \tag{2.24}$$

Note that this does not depend on the data, so that Equation (2.22) gives

$$\boldsymbol{F}(\boldsymbol{\beta}) = \mathrm{E}[\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})] = \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}) \tag{2.25}$$

[2] Some texts refer to $\boldsymbol{F}_{\text{obs}}(\hat{\boldsymbol{\beta}})$ as the *Observed* Fisher Information, and to $\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})$ simply as the Fisher Information. Some don't refer to either of these at all. Just to be clear, we will refer to $\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})$ as the Observed Fisher Information and the Expected Fisher Information $\boldsymbol{F}(\boldsymbol{\beta})$ simply as the Fisher Information.

2.5.2 With Identity Link

The identity link is defined such that $h(\eta) = \eta$.

In this case we have that:

- $\lambda(\boldsymbol{x}) = \mu(\boldsymbol{x}) = h(\eta(\boldsymbol{x})) = \eta(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{x}$
- $\mathcal{V}(\mu) = \mu$ (see Poisson example in EDF chapter)
- $h'(\eta) = 1$

Equation (2.17) gives

$$\begin{aligned}
\boldsymbol{S}(\boldsymbol{\beta}) &= \sum_i (y_i - \mu_i)\,\frac{1}{\mu_i}\,1\,\boldsymbol{x}_i && (2.26)\\
&= \sum_i (y_i - \boldsymbol{\beta}^T\boldsymbol{x}_i)\,\frac{1}{\boldsymbol{\beta}^T\boldsymbol{x}_i}\,\boldsymbol{x}_i && (2.27)\\
&= \sum_i \left(\frac{y_i}{\boldsymbol{\beta}^T\boldsymbol{x}_i} - 1\right)\,\boldsymbol{x}_i && (2.28)
\end{aligned}$$

The Observed Fisher Information is given by

$$\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}) = \sum_i \frac{y_i}{(\boldsymbol{\beta}^T\boldsymbol{x}_i)^2}\,\boldsymbol{x}_i\boldsymbol{x}_i^T \qquad (2.29)$$

and the Fisher Information is given by

$$\begin{aligned}
\boldsymbol{F}(\boldsymbol{\beta}) &= \mathrm{E}[\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})] && (2.30)\\
&= \mathrm{E}\left[\sum_i \frac{Y_i}{(\boldsymbol{\beta}^T\boldsymbol{x}_i)^2}\,\boldsymbol{x}_i\boldsymbol{x}_i^T\right] && (2.31)\\
&= \sum_i \frac{\mathrm{E}[Y_i|\boldsymbol{\beta},\boldsymbol{x}_i]}{(\boldsymbol{\beta}^T\boldsymbol{x}_i)^2}\,\boldsymbol{x}_i\boldsymbol{x}_i^T && (2.32)\\
&= \sum_i \frac{\boldsymbol{\beta}^T\boldsymbol{x}_i}{(\boldsymbol{\beta}^T\boldsymbol{x}_i)^2}\,\boldsymbol{x}_i\boldsymbol{x}_i^T && (2.33)\\
&= \sum_i \frac{1}{\boldsymbol{\beta}^T\boldsymbol{x}_i}\,\boldsymbol{x}_i\boldsymbol{x}_i^T && (2.34)
\end{aligned}$$

Note that $\boldsymbol{F}(\boldsymbol{\beta}) \neq \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})$ in this case.

2.6 Properties of $\boldsymbol{S}(\boldsymbol{\beta})$ and $\boldsymbol{F}(\boldsymbol{\beta})$

Defining $S_i(\boldsymbol{\beta}) = \frac{\partial l_i}{\partial \boldsymbol{\beta}}$, we have that $\boldsymbol{S}(\boldsymbol{\beta}) = \sum_i \boldsymbol{S}_i(\boldsymbol{\beta})$.

2.6.1 Expectation of $\boldsymbol{S}(\boldsymbol{\beta})$

The expectation of $\boldsymbol{S}(\boldsymbol{\beta})$ can be computed from Equation (2.11) as follows:

$$\begin{aligned}
\mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})] &= \sum_i \mathrm{E}[\boldsymbol{S}_i(\boldsymbol{\beta})] && (2.35)\\
&= \sum_i \frac{\mathrm{E}[Y_i|\boldsymbol{\beta}, \boldsymbol{x}_i] - \mu_i}{\phi_i} \frac{1}{\mathcal{V}(\mu_i)} h'(\eta_i)\ \boldsymbol{x}_i && (2.36)\\
&= 0 && (2.37)
\end{aligned}$$

because $\mathrm{E}[Y_i|\boldsymbol{\beta}, \boldsymbol{x}_i] = \mu_i$.

2.6.2 Variance of $\boldsymbol{S}(\boldsymbol{\beta})$

To calculate the variance of $\boldsymbol{S}(\boldsymbol{\beta})$, we proceed as follows. First, we note that

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta})] &= \mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})\boldsymbol{S}(\boldsymbol{\beta})^T] - \mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})]\mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})^T] && (2.38)\\
&= \mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})\boldsymbol{S}(\boldsymbol{\beta})^T] && (2.39)
\end{aligned}$$

because $\mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})] = 0$ from Equation (2.37).

We then have (dropping the argument $\boldsymbol{\beta}$)

$$\mathrm{E}[\boldsymbol{S}\boldsymbol{S}^T] = \mathrm{E}\left[\left(\sum_i \boldsymbol{S}_i\right)\left(\sum_j \boldsymbol{S}_j^T\right)\right] = \sum_{i,j} \mathrm{E}[\boldsymbol{S}_i\boldsymbol{S}_j^T] \qquad (2.40)$$

with

$$\begin{aligned}
\mathrm{E}[\boldsymbol{S}_i\boldsymbol{S}_j^T] &= \mathrm{E}\left[\left(\frac{Y_i - \mu_i}{\phi_i}\frac{1}{\mathcal{V}(\mu_i)}h'(\eta_i)\ \boldsymbol{x}_i\right)\left(\frac{Y_j - \mu_j}{\phi_j}\frac{1}{\mathcal{V}(\mu_j)}h'(\eta_j)\ \boldsymbol{x}_j\right)\right] && (2.41)\\
&= \delta_{ij}\ \frac{\phi_i\mathcal{V}(\mu_i)}{\phi_i^2\mathcal{V}(\mu_i)^2}\ h'(\eta_i)^2\ \boldsymbol{x}_i\boldsymbol{x}_i^T && (2.42)\\
&= \delta_{ij}\ \frac{1}{\phi_i\mathcal{V}(\mu_i)}\ h'(\eta_i)^2\ \boldsymbol{x}_i\boldsymbol{x}_i^T && (2.43)
\end{aligned}$$

since

$$\begin{aligned}
\mathrm{E}[(Y_i - \mu_i)^2] &= \mathrm{Var}[Y_i|\boldsymbol{\beta}, \boldsymbol{x}_i] = \phi_i\mathcal{V}(\mu_i) && && (2.44)\\
\mathrm{E}[(Y_i - \mu_i)(Y_j - \mu_j)] &= \mathrm{Cov}[Y_i, Y_j|\boldsymbol{\beta}, \boldsymbol{x}_i] = 0 && i \neq j && (2.45)
\end{aligned}$$

so that

$$\mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta})] = \sum_i \frac{h'(\eta_i)^2}{\phi_i\mathcal{V}(\mu_i)}\ \boldsymbol{x}_i\boldsymbol{x}_i^T \qquad (2.46)$$

2.6.3 $\boldsymbol{F}(\boldsymbol{\beta})$

2.6.3.1 An Important Identity

Let $\rho = e^l$, where l is the log-likelihood, so that $\rho = L(\boldsymbol{\beta}) = P(\{y_{ir_i}\} \,|\, \{\boldsymbol{x}_i\}, \boldsymbol{\beta})$ is the likelihood/probability of the data. Then

$$\frac{\partial l}{\partial \boldsymbol{\beta}^T} = \frac{\partial l}{\partial \rho}\frac{\partial \rho}{\partial \boldsymbol{\beta}^T} = \frac{1}{\rho}\frac{\partial \rho}{\partial \boldsymbol{\beta}^T} \tag{2.47}$$

and[3]

$$\begin{aligned}\frac{\partial^2 l}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} &= -\frac{1}{\rho^2}\frac{\partial \rho}{\partial \boldsymbol{\beta}^T}\frac{\partial \rho}{\partial \boldsymbol{\beta}} + \frac{1}{\rho}\frac{\partial^2 \rho}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} &(2.48)\\ &= -\frac{\partial l}{\partial \boldsymbol{\beta}^T}\frac{\partial l}{\partial \boldsymbol{\beta}} + \frac{1}{\rho}\frac{\partial^2 \rho}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} &(2.49)\end{aligned}$$

The expectation (over the data) of the second term is then

$$\mathrm{E}\left[\frac{1}{\rho}\frac{\partial^2 \rho}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}\right] = \int \rho\,\frac{1}{\rho}\frac{\partial^2 \rho}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} = \int \frac{\partial^2 \rho}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}} = \frac{\partial^2}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}\int \rho = \frac{\partial^2}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}\,1 = 0 \tag{2.50}$$

2.6.3.2 Relating $\boldsymbol{F}(\boldsymbol{\beta})$ and $\mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta})]$

Using Equations (2.49) and (2.50), we have that

$$\boldsymbol{F}(\boldsymbol{\beta}) = -\mathrm{E}\left[\frac{\partial^2 l}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}\right] = \mathrm{E}\left[\frac{\partial l}{\partial \boldsymbol{\beta}^T}\frac{\partial l}{\partial \boldsymbol{\beta}}\right] = \mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta})\boldsymbol{S}(\boldsymbol{\beta})^T] = \mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta})] \tag{2.51}$$

2.6.3.3 Natural Link

For the natural link, recall that $\frac{h'(\eta_i)}{\mathcal{V}(\mu_i)} = 1$, so that:

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{\beta}) &= \sum_i \frac{1}{\phi_i}(y_i - h(\eta_i))\boldsymbol{x}_i &(2.52)\\ \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}) &= -\boldsymbol{S}'(\boldsymbol{\beta}) = \sum_i \frac{h'(\eta_i)}{\phi_i}\boldsymbol{x}_i\boldsymbol{x}_i^T &(2.53)\\ \boldsymbol{F}(\boldsymbol{\beta}) &= \mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta})] = \sum_i \frac{h'(\eta_i)}{\phi_i}\boldsymbol{x}_i\boldsymbol{x}_i^T &(2.54)\end{aligned}$$

Thus, for the natural link, we see that $\boldsymbol{F}(\boldsymbol{\beta}) = \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta})$.

[3] Note that $\frac{\partial}{\partial \boldsymbol{\beta}}\frac{1}{\rho(\boldsymbol{\beta})} = -\frac{1}{\rho^2}\frac{\partial \rho}{\partial \boldsymbol{\beta}}$.

2.7 Matrix Notation

For the next section, it is useful to establish a condensed, matrix notation for some of the previous quantities, analogous to the matrix notation used for linear models.

- Let $\boldsymbol{Y} \in \mathbb{R}^n$ be the random vector with components Y_i, the response values. This is exactly the same quantity as in the linear model case.
- Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ be the *design matrix*, the matrix with components $x_{i,a}$, the value of the a^{th} component of the predictor vector for the i^{th} data point. This is exactly the same quantity as in the linear model case.
- Let $\boldsymbol{\mu} \in \mathbb{R}^n$ be the vector with components $\mu_i = h(\boldsymbol{\beta}^T x_i)$, so that

$$\boldsymbol{\mu} = \mathrm{E}[\boldsymbol{Y}] \tag{2.55}$$

- Let $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ be the diagonal matrix with components $D_{ii} = h'(\eta_i)$. For example, if $h(\eta) = e^{\eta}$, then

$$\boldsymbol{D} = \begin{pmatrix} e^{\boldsymbol{\beta}^T x_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & e^{\boldsymbol{\beta}^T x_n} \end{pmatrix} \tag{2.56}$$

- Let $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ be the covariance matrix for $\boldsymbol{Y}$, with components:

$$\Sigma_{ij} = \mathrm{Cov}[Y_i, Y_j] = \mathrm{Var}[Y_i]\ \delta_{ij} = \phi_i \mathcal{V}(\mu_i)\ \delta_{ij} \tag{2.57}$$

that is,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{Var}[Y_1] & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathrm{Var}[Y_n] \end{pmatrix} = \begin{pmatrix} \phi_1\mathcal{V}(\mu_1) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \phi_n\mathcal{V}(\mu_n) \end{pmatrix} \tag{2.58}$$

2.7.1 Score Function and Fisher Information

Recall that

$$\boldsymbol{S}(\boldsymbol{\beta}) = \sum_i \left(\frac{y_i - \mu_i}{\phi_i \mathcal{V}(\mu_i)}\right)\ h'(\eta_i)\ x_i \tag{2.59}$$

$$\boldsymbol{F}(\boldsymbol{\beta}) = \sum_i \frac{h'(\eta_i)^2}{\phi_i \mathcal{V}(\mu_i)}\ x_i x_i^T \tag{2.60}$$

In terms of the matrix notation, these become

$$\boldsymbol{S} = \boldsymbol{X}^T \boldsymbol{D} \boldsymbol{\Sigma}^{-1} (\boldsymbol{Y} - \boldsymbol{\mu}) \tag{2.61}$$

$$\boldsymbol{F} = \boldsymbol{X}^T \boldsymbol{D}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{D} \boldsymbol{X} \tag{2.62}$$

2.7.2 Natural Link

Note that for the natural link,

$$\frac{\partial\theta_i}{\partial\eta_i} = \frac{h'(\eta_i)}{\mathcal{V}(\mu_i)} = 1 \tag{2.63}$$

Thus

$$h'(\eta_i) = \mathcal{V}(\mu_i) = \frac{\mathrm{Var}[Y_i]}{\phi_i} = m_i\frac{\mathrm{Var}[Y_i]}{\phi} \tag{2.64}$$

if $\phi_i = \phi/m_i$.

Now

- Let $\boldsymbol{G} \in \mathbb{R}^{n\times n}$ be the diagonal matrix with components $m_i\delta_{ij}$, known as the *grouping* matrix. Then

$$\boldsymbol{D} = \frac{1}{\phi}\boldsymbol{G\Sigma} = \frac{1}{\phi}\boldsymbol{\Sigma G} \tag{2.65}$$

and thus

$$\boldsymbol{S}(\boldsymbol{\beta}) = \frac{1}{\phi}\boldsymbol{X}^T\boldsymbol{G}(\boldsymbol{Y}-\boldsymbol{\mu}) \tag{2.66}$$

$$\boldsymbol{F}(\boldsymbol{\beta}) = \frac{1}{\phi^2}\boldsymbol{X}^T\boldsymbol{G}^T\boldsymbol{\Sigma G X} \tag{2.67}$$

2.8 Iterative Solution of $\boldsymbol{S}(\hat{\boldsymbol{\beta}}) = 0$

So far we have seen how to set up the score equation for the maximum likelihood estimate, and some of its properties, as well as those of the Fisher Information. We now turn to the question of how to solve the score equation. As we have seen, except in rare cases, this cannot be done in closed form, and so we turn to numerical methods, implemented on a computer. We have the same two options here as in the binary regression case. We can try to optimise l directly, or we can attempt to solve the score equation. There are many algorithms that can be used to perform these tasks. Here we focus on one: *iteratively reweighted least squares (IRLS)*, also known as *iteratively weighted least squares (IWLS)*.[4]

We start by recalling the Newton-Raphson method for finding the zero of a function. We wish to solve an equation

$$\boldsymbol{S}(\hat{\boldsymbol{\beta}}) = 0 \tag{2.68}$$

We then approximate $\boldsymbol{S}$ linearly about some point:

$$\boldsymbol{S}(\boldsymbol{\beta}_0 + \delta\boldsymbol{\beta}_0) = \boldsymbol{S}(\boldsymbol{\beta}_0) + \frac{\partial\boldsymbol{S}(\boldsymbol{\beta}_0)}{\partial\boldsymbol{\beta}}\delta\boldsymbol{\beta}_0 + \mathcal{O}(\delta\boldsymbol{\beta}_0^2) \tag{2.69}$$

[4] You have cause to be particularly interested in this algorithm as a Durham student. It is on the undergraduate syllabus of nearly every maths degree in the world which includes a large statistical component and some of the important early development was researched by Dr Peter Green when he was a lecturer at Durham: Green [1984].

where the reason for the subscript 0 will become apparent soon. In the case when $\boldsymbol{S}(\boldsymbol{\beta}_0 + \delta\boldsymbol{\beta}_0) = 0$ (such as we are interested in), we have approximately that

$$\frac{\partial \boldsymbol{S}(\boldsymbol{\beta}_0)}{\partial \boldsymbol{\beta}} \delta\boldsymbol{\beta}_0 = -\boldsymbol{S}(\boldsymbol{\beta}_0) \tag{2.70}$$

Now in our case

$$-\frac{\partial \boldsymbol{S}(\boldsymbol{\beta}_0)}{\partial \boldsymbol{\beta}} = \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0) \tag{2.71}$$

so Equation (2.70) becomes

$$\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0)\delta\boldsymbol{\beta}_0 = \boldsymbol{S}(\boldsymbol{\beta}_0) \tag{2.72}$$

or equivalently

$$\delta\boldsymbol{\beta}_0 = \left(\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0)\right)^{-1} \boldsymbol{S}(\boldsymbol{\beta}_0) \tag{2.73}$$

This then gives a new value

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + \delta\boldsymbol{\beta}_0 \tag{2.74}$$

, and we then iterate:

$$\boldsymbol{\beta}_{m+1} = \boldsymbol{\beta}_m + \delta\boldsymbol{\beta}_m \tag{2.75}$$

where

$$\delta\boldsymbol{\beta}_m = \left(\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_m)\right)^{-1} \boldsymbol{S}(\boldsymbol{\beta}_m) \tag{2.76}$$

Because $\boldsymbol{F}_{\text{obs}}$ is hard to find and hard to invert in general, we approximate it with the expected Fisher Information. This is known as *Fisher scoring*:

$$\delta\boldsymbol{\beta}_m = \left(\boldsymbol{F}(\boldsymbol{\beta}_m)\right)^{-1} \boldsymbol{S}(\boldsymbol{\beta}_m) \tag{2.77}$$

More usefully, we have that

$$\boldsymbol{F}(\boldsymbol{\beta}_m)\delta\boldsymbol{\beta}_m = \boldsymbol{S}(\boldsymbol{\beta}_m) \tag{2.78}$$

or equivalently that

$$\boldsymbol{F}(\boldsymbol{\beta}_m)\boldsymbol{\beta}_{m+1} = \boldsymbol{F}(\boldsymbol{\beta}_m)\boldsymbol{\beta}_m + \boldsymbol{S}(\boldsymbol{\beta}_m) \tag{2.79}$$

By defining $\boldsymbol{W} = \boldsymbol{D}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{D}$, we can write

$$\boldsymbol{F} = \boldsymbol{X}^T\boldsymbol{D}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{D}\boldsymbol{X} = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} \tag{2.80}$$

and

$$\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{D}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}) = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{D}^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}) \tag{2.81}$$

Thus we can calculate the right-hand side of Equation (2.79) from[5]:

$$\boldsymbol{F}_m\boldsymbol{\beta}_m + \boldsymbol{S}_m = \boldsymbol{X}^T\boldsymbol{W}_m\boldsymbol{X}\boldsymbol{\beta}_m + \boldsymbol{X}^T\boldsymbol{W}_m\boldsymbol{D}_m^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}_m) = \boldsymbol{X}^T\boldsymbol{W}_m\tilde{\boldsymbol{Y}}_m \tag{2.82}$$

where

$$\tilde{\boldsymbol{Y}}_m = \boldsymbol{X}\boldsymbol{\beta}_m + \boldsymbol{D}_m^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}_m) \tag{2.83}$$

[5] subscript m means evaluated using $\boldsymbol{\beta}_m$ or derived quantities

are the so-called *working observations*.

By replacing the left hand side of Equation (2.79) with $\boldsymbol{X}^T\boldsymbol{W}_m\boldsymbol{X}$, we have that

$$
\begin{aligned}
(\boldsymbol{X}^T\boldsymbol{W}_m\boldsymbol{X})\boldsymbol{\beta}_{m+1} &= \boldsymbol{X}^T\boldsymbol{W}_m\tilde{\boldsymbol{Y}}_m \qquad (2.84)\\
\boldsymbol{\beta}_{m+1} &= (\boldsymbol{X}^T\boldsymbol{W}_m\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}_m\tilde{\boldsymbol{Y}}_m \qquad (2.85)
\end{aligned}
$$

This sequence of iterated operations is called *iteratively re-weighted least squares (IRLS)* or *iterative weighted least squares (IWLS)* since each iteration is the solution to the following least squares problem: minimize the quantity $l_m(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$, where

$$
l_m(\boldsymbol{\beta}) = (\tilde{\boldsymbol{Y}}_m - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{W}_m(\tilde{\boldsymbol{Y}}_m - \boldsymbol{X}\boldsymbol{\beta}) \tag{2.86}
$$

As a result, $\boldsymbol{W}$ is known as the *weight matrix*.

2.8.1 IRLS Pseudo-Code

Note that the following is pseudo-code for running IRLS, as without computing $\boldsymbol{\mu}$, $\boldsymbol{D}$ and $\boldsymbol{W}$ using a specific example this will not run.

```
IRLS <- function(Y, X, phi, epsilon) {
    # Pick an initial value for hatBeta.
    hatbeta = initializeBeta()

    # Set up convergence.
    converged = false

    # Loop as long as convergence condition is not satisfied.
    while not converged loop
    {

        # Compute mu, D, and Sigma (use h, h', V as subroutines)
        mu = computeMu(hatBeta, X)
        D = computeD(hatBeta, X)
        Sigma = computeSigma(hatBeta, phi)

        # Compute the weight matrix, W.
        W = t(D) %*% solve(Sigma) %*% D

        # Compute the working observations, tildeY.
        tildeY = X %*% hatBeta + solve(D) %*% (Y - mu)

        # Compute the new value of hatBeta.
        newHatBeta = solve(t(X) %*% W %*% X) %*% (t(X) %*% W %*% tildeY)

        # Check whether we have converged.
        converged = ((norm(newHatBeta - hatBeta) / norm(hatBeta)) <= epsilon)
        # Store new value of hatBeta ready for next iteration or return.
        hatBeta = newHatBeta

    }

    return hatBeta

}
```

2.9 Practical Example: US Polio Data

We start by loading the amended polio data from library `gamlss.data` as discussed in the last term.

```
library( "gamlss.data" )
```

```
##
## Attaching package: 'gamlss.data'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
data( "polio" )
uspolio <- as.data.frame( matrix( c( 1:168, t( polio ) ), ncol = 2 ) )
colnames( uspolio ) <- c("time", "cases")
```

We begin by fitting a Poisson model with a linear time trend.

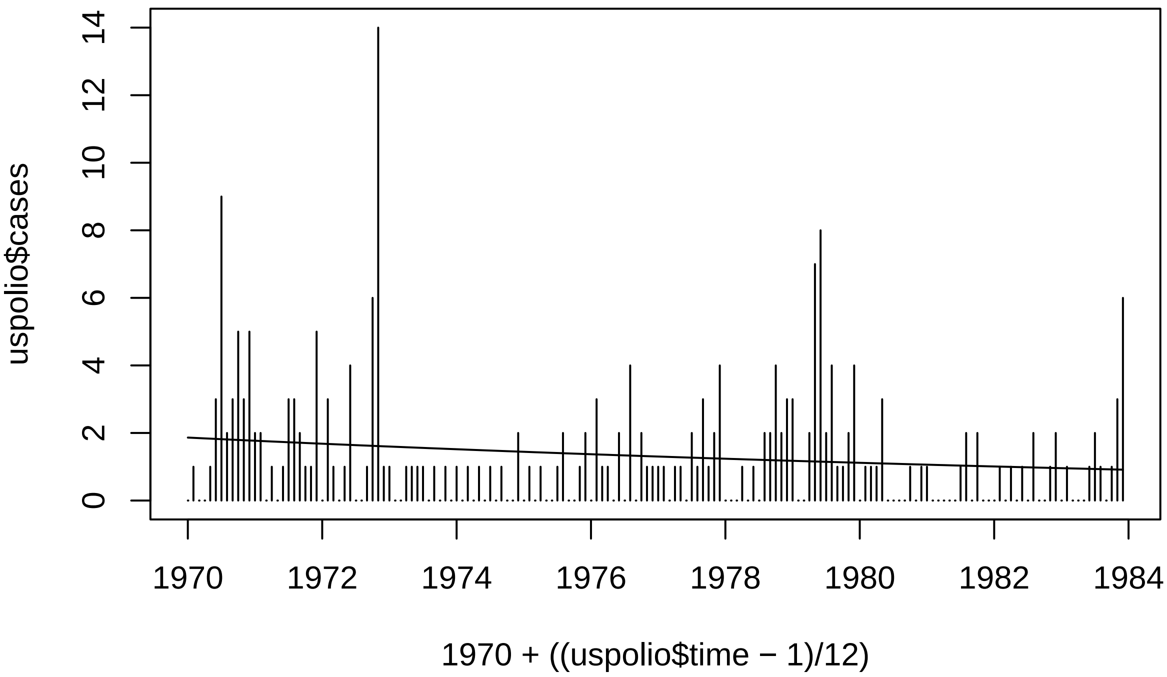
```
# Poisson model with linear time trend
polio.glm <- glm( cases ~ time, family = poisson( link = log ), data = uspolio )

# Look at the summary.
summary( polio.glm )
```

```
##
## Call:
## glm(formula = cases ~ time, family = poisson(link = log), data = uspolio)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.626639   0.123641   5.068 4.02e-07 ***
## time        -0.004263   0.001395  -3.055  0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 333.55  on 166  degrees of freedom
## AIC: 594.59
##
## Number of Fisher Scoring iterations: 5
```

We can then plot the model as follows.

```
plot(1970 + ((uspolio$time - 1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time - 1)/12), polio.glm$fitted)
```

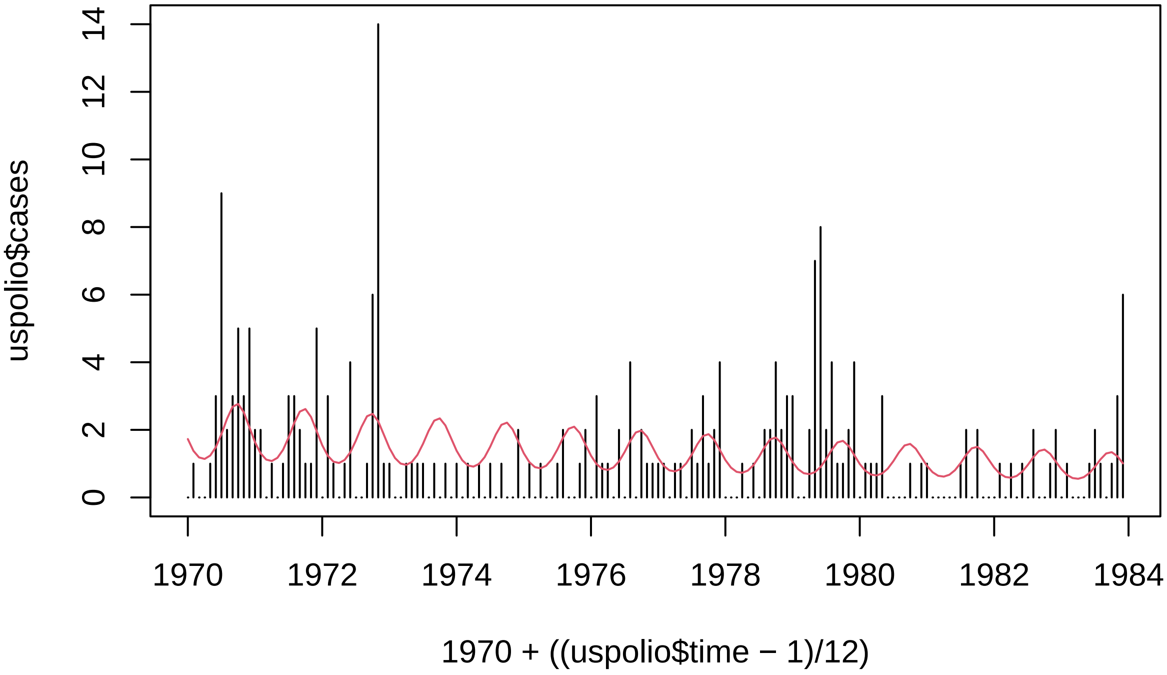


We can see that this is perhaps unsatisfactory. We explore a linear trend with seasonal (annual) component.

```
# Poisson model with linear trend and seasonal (annual) component
polio1.glm<- glm(cases~time + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12)),
family=poisson(link=log), data=uspolio)

summary(polio1.glm)
```

```
##
## Call:
## glm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)), family = poisson(link = log), data = uspolio)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.606612   0.124800   4.861 1.17e-06 ***
## time                     -0.004644   0.001401  -3.315 0.000916 ***
## I(cos(2 * pi * time/12))  0.181254   0.096160   1.885 0.059442 .
## I(sin(2 * pi * time/12)) -0.423187   0.097590  -4.336 1.45e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 310.72  on 164  degrees of freedom
## AIC: 575.77
##
## Number of Fisher Scoring iterations: 5
```

```
plot(1970 + ((uspolio$time - 1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time - 1)/12), polio1.glm$fitted,col=2)
```



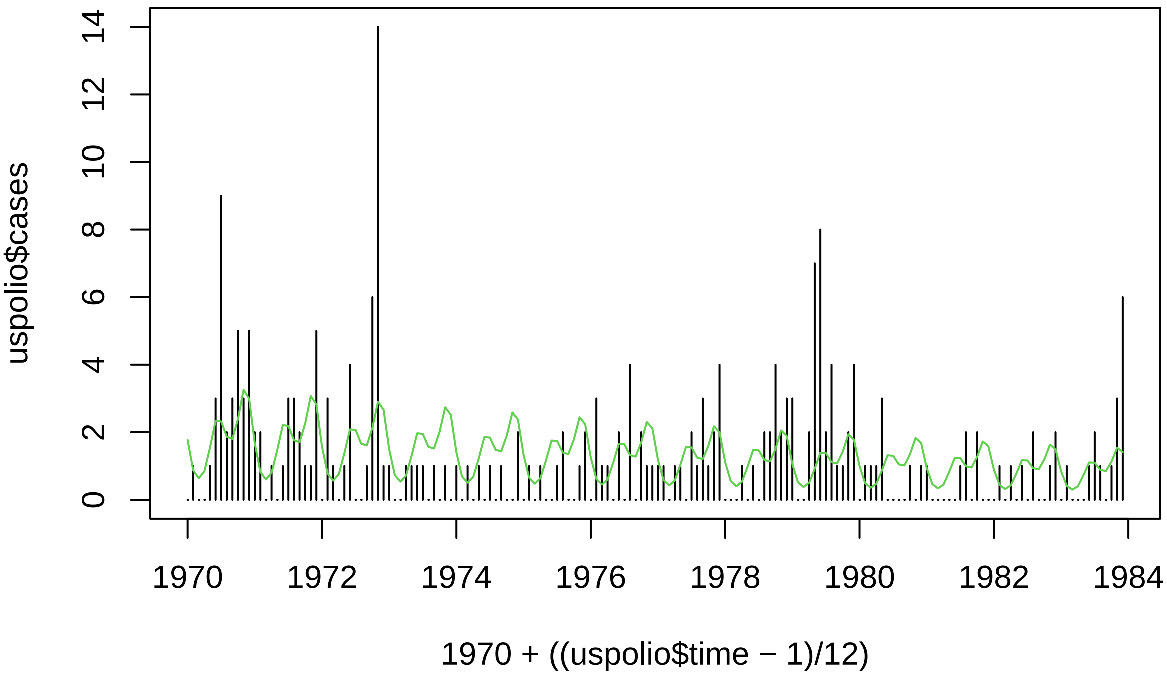
How about now with a six-monthly component.

```
# Poisson model with linear trend and seasonal (annual + sixmonthly) component
polio2.glm<- glm(cases~time + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
+ I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)), family=poisson(link=log),
data=uspolio)

summary(polio2.glm)
```

```
##
## Call:
## glm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi *
##     time/6)), family = poisson(link = log), data = uspolio)
##
## Coefficients:
```

```
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.557241   0.127303   4.377 1.20e-05 ***
## time                       -0.004799   0.001403  -3.421 0.000625 ***
## I(cos(2 * pi * time/12))    0.137132   0.089479   1.533 0.125384
## I(sin(2 * pi * time/12))   -0.534985   0.115476  -4.633 3.61e-06 ***
## I(cos(2 * pi * time/6))     0.458797   0.101467   4.522 6.14e-06 ***
## I(sin(2 * pi * time/6))    -0.069627   0.098123  -0.710 0.477957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 288.85  on 162  degrees of freedom
## AIC: 557.9
##
## Number of Fisher Scoring iterations: 5
```

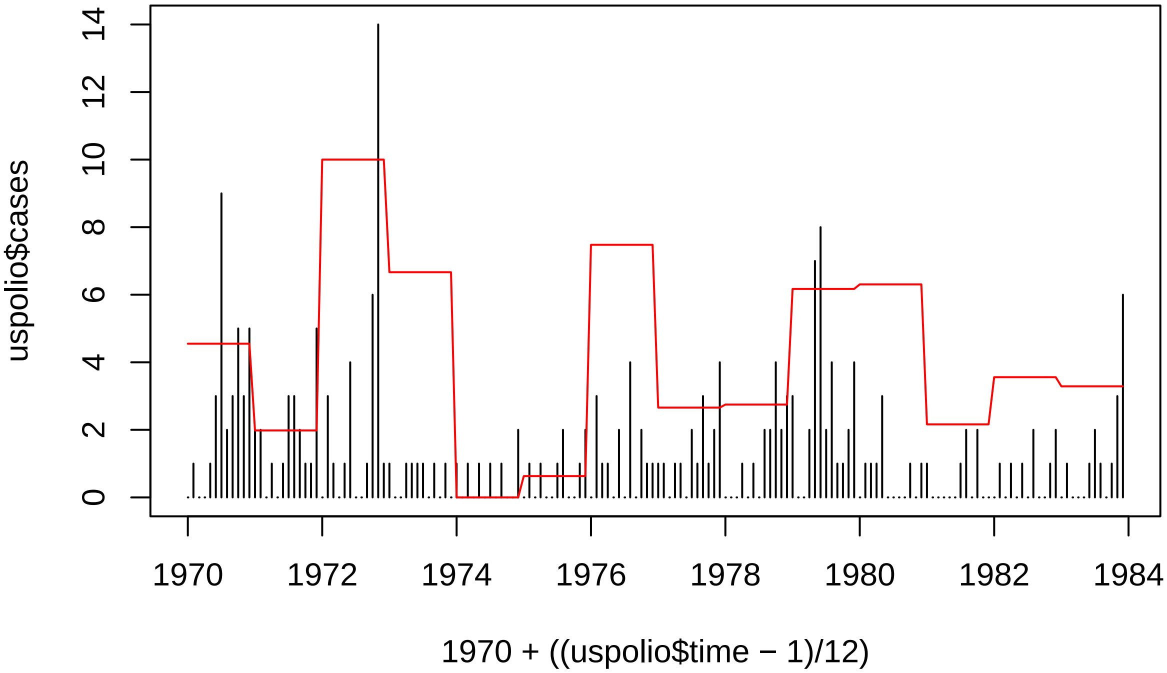
```r
plot(1970 + ((uspolio$time - 1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time - 1)/12), polio2.glm$fitted,col=3)
```



Add in temperature data.

```r
# average annual temperature data over the 14 years
temp_data <- rep(c(5.195, 5.138, 5.316, 5.242, 5.094, 5.108, 5.260, 5.153,
                   5.155, 5.231, 5.234, 5.142, 5.173, 5.167), each = 12 )
# scale the data so that it plots nicely.
scaled_temp = 10 * (temp_data - min(temp_data))/(max(temp_data) - min(temp_data))
uspolio$temp = scaled_temp
# Plot temperature data against cases data to see interest.
```

```
plot(1970 + ((uspolio$time - 1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time - 1)/12), uspolio$temp, col="red")
```



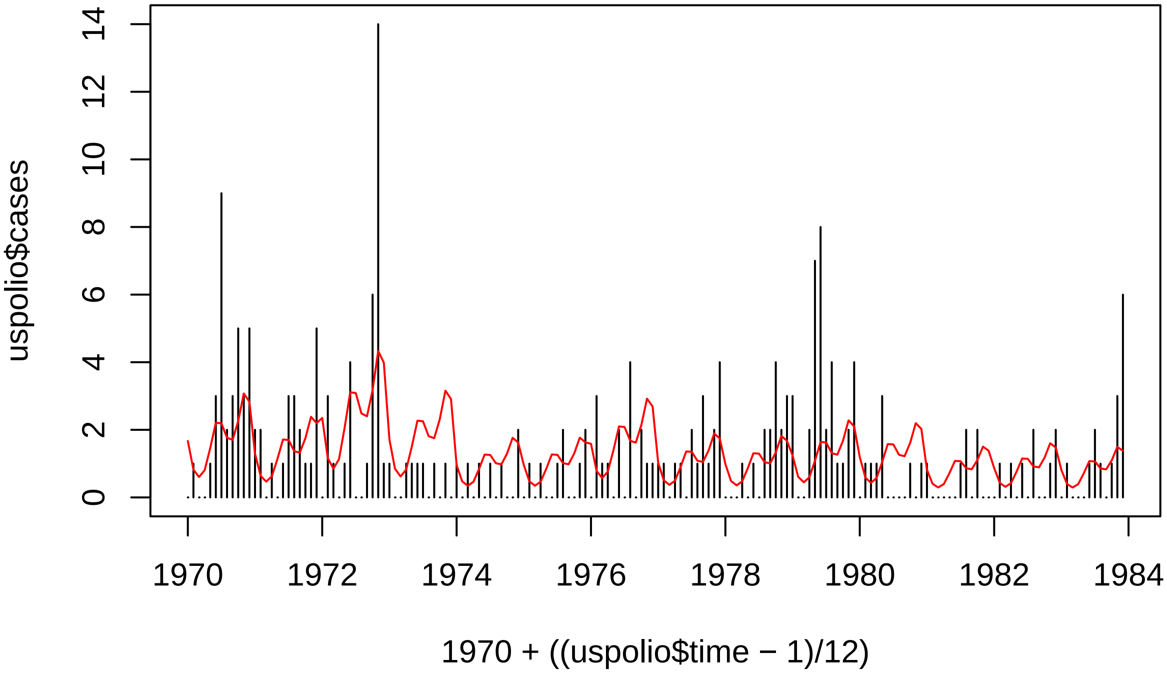
Poisson GLM with temp data.

```
# Construct GLM.
polio3.glm<- glm(cases~time + temp + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
+ I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)) , family=poisson(link=log),
data=uspolio)

summary(polio3.glm)
```

```
##
## Call:
## glm(formula = cases ~ time + temp + I(cos(2 * pi * time/12)) +
##     I(sin(2 * pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 *
##     pi * time/6)), family = poisson(link = log), data = uspolio)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.129643   0.186352   0.696 0.486623
## time                     -0.003972   0.001439  -2.761 0.005770 **
## temp                      0.080308   0.023139   3.471 0.000519 ***
## I(cos(2 * pi * time/12))  0.136094   0.089489   1.521 0.128314
## I(sin(2 * pi * time/12)) -0.531668   0.115466  -4.605 4.13e-06 ***
## I(cos(2 * pi * time/6))   0.457487   0.101435   4.510 6.48e-06 ***
## I(sin(2 * pi * time/6))  -0.068345   0.098149  -0.696 0.486218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

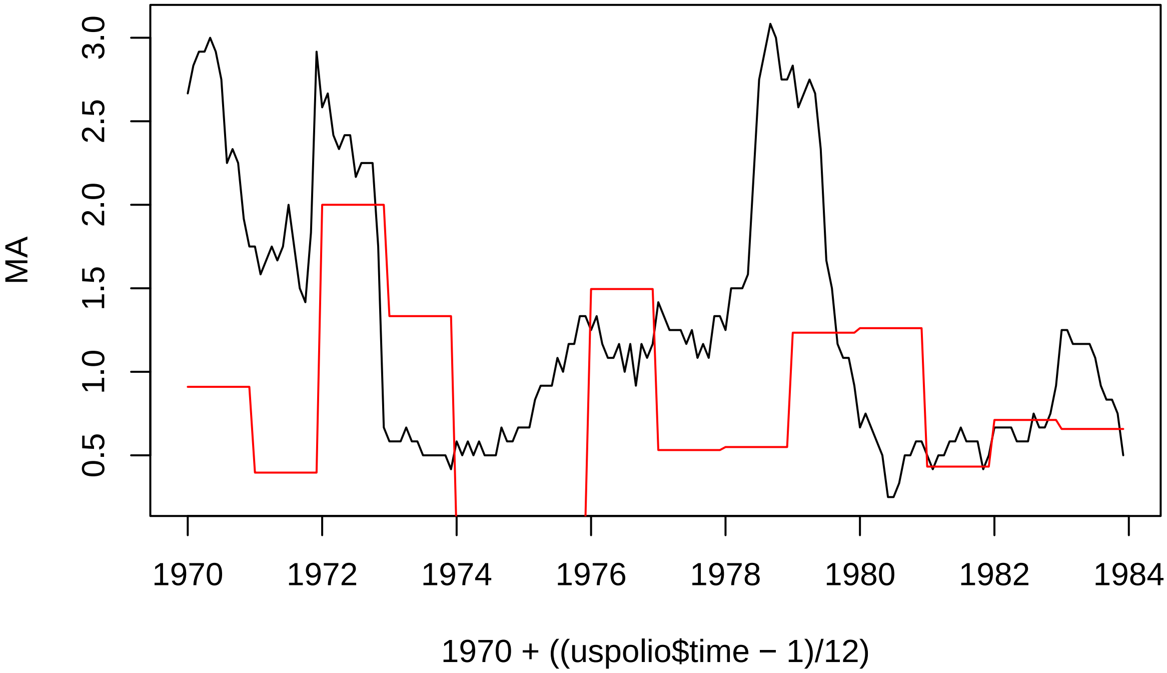
```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 276.84  on 161  degrees of freedom
## AIC: 547.88
##
## Number of Fisher Scoring iterations: 5
```

```
plot(1970 + ((uspolio$time - 1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time - 1)/12), polio3.glm$fitted, col="red")
```



Compare to moving average window. . .

```
# Compare to simple moving average

# Size of averaging window.

# Try m = 3, 6, 12, 60, 120
m = 12

MA = rep(0, length(uspolio$time))

for (time in uspolio$time)
{
times = time:min(time + m - 1, length(uspolio$time))
n = length(times)
sum = 0

for (newtime in times)
{
sum = sum + uspolio$cases[newtime]
}

MA[time] = sum / m
}


plot(1970 + ((uspolio$time - 1)/12), MA, type = "l")
lines(1970 + ((uspolio$time - 1)/12), 0.2*uspolio$temp, col="red")
```



2.10 Estimation of ϕ

There is no need to estimate the dispersion ϕ in order to estimate $\boldsymbol{\beta}$, because ϕ cancels from the equation $\boldsymbol{S}(\hat{\boldsymbol{\beta}}) = 0$. However, $\text{Var}[\hat{\boldsymbol{\beta}}]$ does depend on ϕ, as one might expect. If necessary, or of interest, ϕ can be estimated via:

$$\hat{\phi} = \frac{1}{n-p} \sum_i m_i \frac{(y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)} \tag{2.87}$$

The motivation is that

$$\text{Var}[y_i] = \text{E}[((y_i - \mu_i)^2)] = \phi_i \mathcal{V}(\mu_i) = \frac{\phi}{m_i} \mathcal{V}(\mu_i) \tag{2.88}$$

2.10.1 Special Cases

Two special cases are as follows:

2.10.1.1 Gaussian

When $Y|\boldsymbol{\beta}, x \sim \mathcal{N}(\mu, \sigma^2)$, with $m = 1$, we have

$$\hat{\phi} = \frac{1}{n-p}\sum_i (y_i - \hat{\mu}_i)^2 = \hat{\sigma}^2 \tag{2.89}$$

2.10.1.2 Gamma

When $Y|\boldsymbol{\beta}, x \sim \text{Gamma}(\mu, \sigma^2)$[6], we have

$$\frac{1}{\hat{\nu}} = \hat{\phi} = \frac{1}{n-p}\sum_i m_i \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i^2} \tag{2.90}$$

2.10.2 Practical Example: Hospital Stay Data

```r
library(npmlreg)
data(hosp)
hosp.glm <- glm(duration~age+temp1, data=hosp, family=Gamma(link=log))

summary(hosp.glm)
```

```
##
## Call:
## glm(formula = duration ~ age + temp1, family = Gamma(link = log),
##     data = hosp)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.654096  16.621018  -1.724   0.0987 .
## age           0.014900   0.005698   2.615   0.0158 *
## temp1         0.306624   0.168141   1.824   0.0818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2690233)
##
##     Null deviance: 8.1722  on 24  degrees of freedom
## Residual deviance: 5.7849  on 22  degrees of freedom
## AIC: 142.73
##
## Number of Fisher Scoring iterations: 6
```

```r
#(Dispersion parameter for Gamma family taken to be 0.2690233)
# by hand:
1/(hosp.glm$df.res)*sum( (hosp$duration-hosp.glm$fitted)^2/(hosp.glm$fitted^2))
```

[6] Recall: that it was shown in Exercise 6.1 how to parameterise the Gamma function in terms of its mean and variance, and we found that $\mathcal{V}(\mu) = \mu^2$.

```
## [1] 0.2690233
```

```
#[1]   0.2690233
```

2.11 Asymptotic Properties of $\hat{\beta}$

In our context, *asymptotic* means that $M = \sum_{i=1}^{n} m_i \to \infty$. This could be because $n \to \infty$, or because the $m_i \to \infty$, or a combination of both.

Let us denote the true value of $\boldsymbol{\beta}$ by $\boldsymbol{\beta}_0$. Note that this is *not* the initial value of the IRLS algorithm (although we used the same symbol for both). In the following, we assume consistency of $\hat{\boldsymbol{\beta}}$, i.e. that $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}_0$, meaning convergence in probability, the probability that $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0|| \geq \varepsilon$ tends to 0 as n tends to infinity. We will also denote this by $\hat{\boldsymbol{\beta}} \overset{a}{=} \boldsymbol{\beta}$, and we will abuse notation by also using this to mean *tends to asymptotically* for expectations.

Asymptotically, $\hat{\boldsymbol{\beta}}$ will thus be close to $\boldsymbol{\beta}_0$, and we can expand $\boldsymbol{S}$ around it:

$$\boldsymbol{S}(\hat{\boldsymbol{\beta}}) = 0 \overset{a}{=} \boldsymbol{S}(\boldsymbol{\beta}_0) + \frac{\partial \boldsymbol{S}(\boldsymbol{\beta}_0)}{\partial \boldsymbol{\beta}^T}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \tag{2.91}$$

$$= \boldsymbol{S}(\boldsymbol{\beta}_0) - \boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \tag{2.92}$$

or equivalently

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 = (\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0))^{-1}\boldsymbol{S}(\boldsymbol{\beta}_0) \tag{2.93}$$

2.11.1 Fisher Scoring

In Section 2.8, we stated that we often use the (expected) Fisher Information in place of the Observed Fisher Information (known as Fisher Scoring). Doing so in the context of asymptotic arguments is acceptable. We can roughly see this as follows:

$$\frac{1}{n}\boldsymbol{F}_{\text{obs}}(\boldsymbol{\beta}_0) = -\frac{1}{n}\frac{\partial l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}(\boldsymbol{\beta}_0) = -\frac{1}{n}\sum_{i-1}^{n}\frac{\partial l_i}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}(\boldsymbol{\beta}_0) \to -\text{E}\left[\frac{\partial l_i}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}(\boldsymbol{\beta}_0)\right] = F_1(\boldsymbol{\beta}) \tag{2.94}$$

where $F_1(\boldsymbol{\beta})$ is the expected Fisher Information for a sample of size 1, by the law of large numbers as $n \to \infty$. It can be shown that $\boldsymbol{F}(\boldsymbol{\beta}) = nF_1(\boldsymbol{\beta})$[7], thus justifying use of $\boldsymbol{F}(\boldsymbol{\beta})$ in the forthcoming asymptotic arguments.

2.11.2 Expectation

We have

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \overset{a}{=} \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \boldsymbol{S}(\boldsymbol{\beta}_0) \tag{2.95}$$

Because convergence in probability implies convergence in distribution, this in turn implies that

$$E[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0] \overset{a}{=} \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ E[\boldsymbol{S}(\boldsymbol{\beta}_0)] = 0 \tag{2.96}$$

In other words, $\hat{\boldsymbol{\beta}}$ is asymptotically unbiased.

[7] See Statistical Inference II lecture notes or prove as exercise. Incidentally, there is much background on Fisher Information in the Statistical Inference II lecture notes, so some of this stuff should just be a review!

2.11.3 Variance

Similarly, we have that

$$\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0] &= \mathrm{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T] && (2.97)\\
&\overset{a}{=} \mathrm{E}[\boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \boldsymbol{S}(\boldsymbol{\beta}_0)\ \boldsymbol{S}(\boldsymbol{\beta}_0)^T\ \boldsymbol{F}^{-T}(\boldsymbol{\beta}_0)] && (2.98)\\
&= \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \mathrm{E}[\boldsymbol{S}(\boldsymbol{\beta}_0)\ \boldsymbol{S}(\boldsymbol{\beta}_0)^T]\ \boldsymbol{F}^{-T}(\boldsymbol{\beta}_0) && (2.99)\\
&= \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta}_0)]\ \boldsymbol{F}^{-T}(\boldsymbol{\beta}_0) && (2.100)\\
&= \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0) && (2.101)
\end{aligned}$$

where we have used symmetry of $\boldsymbol{F}$ and the fact that $\boldsymbol{F}(\boldsymbol{\beta}_0) = \mathrm{Var}[\boldsymbol{S}(\boldsymbol{\beta}_0)]$. Thus

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \mathrm{Var}[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0] \overset{a}{=} \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0) \tag{2.102}$$

Note that

$$\boldsymbol{F}(\boldsymbol{\beta}) = \mathrm{E}\left[-\frac{\partial^2 l}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}\right] \tag{2.103}$$

so that the variance of $\hat{\boldsymbol{\beta}}$ can be seen as the inverse precision, or the ‘curvature’, of the log-likelihood function. Note that the greater the curvature, the more precise the inference about $\boldsymbol{\beta}$.

2.11.4 Asymptotic Normality

The following is a sketch of the argument of asymptotic normality. We start from

$$\boldsymbol{S}(\boldsymbol{\beta}) = \sum_i \boldsymbol{S}_i(\boldsymbol{\beta}) \tag{2.104}$$

which defines the $\boldsymbol{S}_i$. This is a sum of independent random variables, with zero mean and finite variance. As the number of terms in the sum tends to infinity, then under a certain condition, the distribution of the sum converges in distribution to a normal distribution:

$$\boldsymbol{S}(\boldsymbol{\beta}) \overset{a}{\sim} \mathcal{N}(0, \boldsymbol{F}(\boldsymbol{\beta})) \tag{2.105}$$

Hence

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \overset{a}{=} \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \boldsymbol{S}(\boldsymbol{\beta}_0) \overset{a}{\sim} \mathcal{N}(0, \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)\ \boldsymbol{F}(\boldsymbol{\beta}_0)\ \boldsymbol{F}^{-T}(\boldsymbol{\beta}_0)) \tag{2.106}$$

Convergence in probability implies convergence in distribution, so

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \overset{a}{\sim} \mathcal{N}(0, \boldsymbol{F}^{-1}(\boldsymbol{\beta}_0)) \tag{2.107}$$

This also implies that the Mahalanobis distance between $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}_0$ is asymptotically chi-square distributed:

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T\ \boldsymbol{F}(\boldsymbol{\beta}_0)\ (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \overset{a}{\sim} \chi^2(p) \tag{2.108}$$

2.11.5 Closing The Circle

Under some regularity conditions,

$$\boldsymbol{F}^{-1}(\boldsymbol{\beta}) = \left(\sum_i m_i \ldots\right)^{-1} \to 0 \tag{2.109}$$

as $M \to \infty$. Thus $\hat{\boldsymbol{\beta}}$ converges in distribution to a constant random variable, which means that it converges in probability too, which is what we were assuming.

Equations (2.102), (2.107), and (2.108) remain valid when $\boldsymbol{F}(\boldsymbol{\beta}_0)$ is replaced by $\boldsymbol{F}(\hat{\boldsymbol{\beta}})$.

2.11.6 Next Step

Now that we have seen how to estimate the parameters, and some of their sampling properties (asymptotically), we can move on to how to use these estimates to make inferences: predictions about new values and confidence intervals.

Chapter 3

Prediction and Inference

3.1 Prediction

Assume a GLM has been fitted, yielding $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$. If we are given a new predictor vector $\boldsymbol{x}_0$, we can compute

$$\hat{\eta}_0 = \hat{\boldsymbol{\beta}}^T \boldsymbol{x}_0 \tag{3.1}$$

Now,

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}] \overset{a}{=} F^{-1}(\hat{\boldsymbol{\beta}}) \tag{3.2}$$

so

$$\mathrm{Var}[\hat{\eta}_0] \overset{a}{=} \boldsymbol{x}_0^T \ \mathrm{Var}[\hat{\boldsymbol{\beta}}] \ \boldsymbol{x}_0 \overset{a}{=} \boldsymbol{x}_0^T \ F^{-1}(\hat{\boldsymbol{\beta}}) \ \boldsymbol{x}_0 \tag{3.3}$$

or, alternatively,

$$\mathrm{SE}[\hat{\eta}_0] \overset{a}{=} \sqrt{x_0^T \ F^{-1}(\hat{\boldsymbol{\beta}}) \ x_0} \tag{3.4}$$

A prediction for a new, unobserved y_0 is then

$$\hat{y}_0 = \mathrm{E}[Y|\hat{\boldsymbol{\beta}}, x_0] = h(\hat{\boldsymbol{\beta}}^T x_0) \tag{3.5}$$

An approximate $(1-\alpha)$ confidence interval for $\mathrm{E}[Y|\boldsymbol{\beta}, x_0]$ is then

$$CI = \left[h\left(\hat{\boldsymbol{\beta}}^T x_0 - z_{\frac{\alpha}{2}} \sqrt{x_0^T \ F^{-1}(\hat{\boldsymbol{\beta}}) \ x_0}\right) \ , \ h\left(\hat{\boldsymbol{\beta}}^T x_0 + z_{\frac{\alpha}{2}} \sqrt{x_0^T \ F^{-1}(\hat{\boldsymbol{\beta}}) \ x_0}\right)\right] \tag{3.6}$$

Note that this is not, in general, symmetric about $h(\hat{\boldsymbol{\beta}}^T x_0)$.

What about predictive intervals for y_0, by analogy with the linear model case? These are more complicated, as they depend on the response distribution, and we do not consider them here.

3.1.1 Example: Hospital Stay Data

Predict the duration of stay for a new individual with age 60, and temperature 99.

We could try

```
predict(hosp.glm, newdata=data.frame(age=60, temp1=99))
```

```
##        1
## 2.595732
```

But this is not what we want - for GLMs, the predict function gives by default the value of the linear predictor.

To predict on the scale of the response, one needs

```
exp(predict(hosp.glm, newdata=data.frame(age=60, temp1=99)))
```

```
##       1
## 13.4064
```

or

```
predict(hosp.glm, newdata=data.frame(age=60, temp1=99), type="response")
```

```
##       1
## 13.4064
```

Similarly we aim to achieve a 95% confidence interval for the expected mean function at age 60 and temperature 99 as for the linear model as follows:

```
# Attempt, as for lm:
predict(hosp.glm, newdata=data.frame(age=60, temp1=99), type="response",
interval="confidence")
```

```
##       1
## 13.4064
```

This does not work, so we need to do it manually!

```
# Compute the predicted linear predictor as above.
lphat  <- predict(hosp.glm, newdata=data.frame(age=60, temp1=99))

# Extract the covariance.
varhat <- summary(hosp.glm)$cov.scaled     # = F^(-1)(betahat)

# Define new data point
x0 = c(1, 60, 99)

# Compute the width of the interval for the linear predictor.
span   <- qnorm(0.975) * sqrt( x0 %*%  varhat %*%  x0)

# Compute the interval for the mean.
c(exp(lphat-span), exp(lphat+span))
```

```
## [1]  8.836973 20.338601
```

```
# Note that this is quite large, as the dataset is small!
```

3.2 Hypothesis Tests

We wish to test the values of $\hat{\boldsymbol{\beta}}$, just as for linear models.

3.2.1 Simple Tests

We take as hypotheses $\mathcal{H}_0 : \boldsymbol{\beta} = \boldsymbol{b}$ and $\mathcal{H}_1 : \boldsymbol{\beta} \neq \boldsymbol{b}$.

3.2.1.1 Wald Test

An obvious candidate for a test statistic is the *Mahalanobis* distance of $\hat{\boldsymbol{\beta}}$ from $\boldsymbol{\beta}$, otherwise know as the *Wald statistic*. Under $\mathcal{H}_0$,

$$W = (\hat{\boldsymbol{\beta}} - \boldsymbol{b})^T \; F(\hat{\boldsymbol{\beta}}) \; (\hat{\boldsymbol{\beta}} - \boldsymbol{b}) \overset{a}{\sim} \chi^2(p) \tag{3.7}$$

The test is then:

- *Reject $\mathcal{H}_0$ at significance level α if $W > \chi^2_{p,\alpha}$.*

3.2.1.2 Likelihood Ratio Test

An alternative is a likelihood ratio test. Define

$$\Lambda = 2 \log \left(\frac{L(\hat{\boldsymbol{\beta}})}{L(\boldsymbol{\beta})} \right) = 2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\boldsymbol{\beta})) \tag{3.8}$$

What is the distribution of Λ? Taylor-expanding ℓ, we find

$$\ell(\boldsymbol{\beta}) \overset{a}{=} \ell(\hat{\boldsymbol{\beta}}) + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T S(\hat{\boldsymbol{\beta}}) - \frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \; F(\hat{\boldsymbol{\beta}}) \; (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \tag{3.9}$$

But $S(\hat{\boldsymbol{\beta}}) = 0$, so we have

$$2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\boldsymbol{\beta})) \overset{a}{=} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \; F(\hat{\boldsymbol{\beta}}) \; (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \overset{a}{\sim} \chi^2(p) \tag{3.10}$$

Under $\mathcal{H}_0$, $\boldsymbol{\beta} = \boldsymbol{b}$, so we have

$$\Lambda = 2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\boldsymbol{b})) \overset{a}{\sim} \chi^2(p) \tag{3.11}$$

We then

- *Reject $\mathcal{H}_0$ at significance level α if $\Lambda > \chi^2_{p,\alpha}$.*

3.2.2 Generalisation to Nested Models

Our hypotheses are now

$$\mathcal{H}_0 : C\boldsymbol{\beta} = \gamma \tag{3.12}$$
$$\mathcal{H}_1 : C\boldsymbol{\beta} \neq \gamma \tag{3.13}$$

where: $C \in \mathbb{R}^{s \times p}$; $\dim(\text{image}(C)) = s$; $\gamma \in \mathbb{R}^s$.

The equation $C\boldsymbol{\beta} = \gamma$ constrains the possible values of $\boldsymbol{\beta}$, reducing the dimensionality of the space of possible solutions by s. The set of $\boldsymbol{\beta} \in \mathbb{R}^p$ satisfying $C\boldsymbol{\beta} = \gamma$ forms a $(p-s)$-dimensional affine subspace of $\mathbb{R}^p$. This therefore corresponds to a *restricted* or *reduced* model, as against $\mathcal{H}_1$, which corresponds to the *full* model. We may sometimes say that $\mathcal{H}_0$ is a *submodel* of $\mathcal{H}_1$ because the parameter space of $\mathcal{H}_0$ is a subset of the parameter space of $\mathcal{H}_1$.

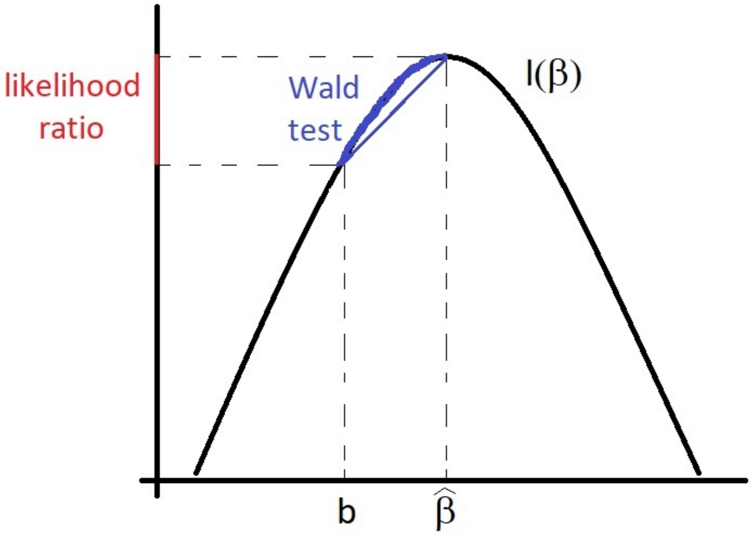


Figure 3.1: Illustration of the relationship between the Wald test and the likelihood ratio test.

3.2.2.1 Example

Let

$$C = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{2 \times p} \tag{3.14}$$
$$\gamma = 0 \in \mathbb{R}^2 \tag{3.15}$$

with $\boldsymbol{\beta} \in \mathbb{R}^p$. Then

$$\mathcal{H}_0 : \begin{cases} \beta_1 = 0 \\ \beta_2 = 0 \end{cases} \tag{3.16}$$

whereas $\mathcal{H}_1$ has β_1 and β_2 unrestricted.

3.2.2.2 Wald Test

We have

$$W = (C\hat{\boldsymbol{\beta}} - \gamma)^T \left(C\, F^{-1}(\hat{\boldsymbol{\beta}})\, C^T\right)^{-1} (C\hat{\boldsymbol{\beta}} - \gamma) \overset{a}{\sim} \chi^2(s) \tag{3.17}$$

where, recall, s is the number of constraints; or, equivalently, the difference in the number of parameters; or, more abstractly, the difference in the dimensions of the parameter spaces.

3.2.2.3 Likelihood Ratio Test

We have

$$\Lambda = 2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\tilde{\boldsymbol{\beta}})) \overset{a}{\sim} \chi^2(s) \tag{3.18}$$

where $\hat{\boldsymbol{\beta}}$ is the MLE under $\mathcal{H}_1$, while $\tilde{\boldsymbol{\beta}}$ is the MLE under $\mathcal{H}_0$, that is, the MLE for the restricted model.[1]

3.2.3 Example: Hospital Stay Data

Consider the hospital data. Construct a Gamma GLM with log link for the `duration` of hospital stay as a function of `age` and `temp1`, the temperature at admission, as follows

$$\eta = \beta_1 + \beta_2 \texttt{age} + \beta_3 \texttt{temp1} \tag{3.19}$$

where

$$\texttt{duration}|\texttt{age}, \texttt{temp1} \sim \text{Gamma}(\nu, \nu e^{-\eta}) \tag{3.20}$$

which can be coded in R as follows:

```r
data(hosp, package="npmlreg")
hosp.glm <- glm(duration~age + temp1, data=hosp, family=Gamma(link=log))
```

Note that the parameterisation chosen means that, from properties of the Gamma distribution:

$$\text{E}[\texttt{duration}|\texttt{age}, \texttt{temp1}] = \frac{\nu}{\nu e^{-\eta}} = e^{\eta} \tag{3.21}$$

$$\text{Var}[\texttt{duration}|\texttt{age}, \texttt{temp1}] = \frac{\nu}{(\nu e^{-\eta})^2} = \frac{e^{2\eta}}{\nu} \tag{3.22}$$

The first equation says that we are using a log link, i.e. an exponential response; the second equation identifies $\phi = \frac{1}{\nu}$ and $\mathcal{V}(\mu) = \mu^2$.

We now wish to test

$$\mathcal{H}_0 : \beta_3 = 0 \tag{3.23}$$

against

$$\mathcal{H}_1 : \beta_3 \neq 0 \tag{3.24}$$

[1] Note that the Wald test does not require the MLE under $\mathcal{H}_0$. Whether this is a good thing is open to question.

We note that

$$C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{1\times 3} \tag{3.25}$$

while $\gamma = 0$, so that the constraint equation can be written

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = 0 \tag{3.26}$$

The variance we are looking for is

$$C\ F^{-1}(\hat{\boldsymbol{\beta}})\ C^T = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} F^{-1}(\hat{\boldsymbol{\beta}}) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \mathrm{Var}[\hat{\boldsymbol{\beta}}_3] = 0.028 \tag{3.27}$$

which can be obtained from R as follows:

```
( varhat <- summary(hosp.glm)$cov.scaled )
```

```
##               (Intercept)           age         temp1
## (Intercept) 276.25822713 -3.728838e-02 -2.7943778049
## age          -0.03728838  3.246846e-05  0.0003656812
## temp1        -2.79437780  3.656812e-04  0.0282713219
```

The Wald statistic is then given by

$$W = \frac{\hat{\boldsymbol{\beta}}_3^2}{\mathrm{Var}[\hat{\boldsymbol{\beta}}_3]} = \frac{(0.31)^2}{0.028} = 3.32 \tag{3.28}$$

Since $\chi^2_{1,0.05} = 3.84$ and $\chi^2_{1,0.1} = 2.71$, we see that we do not reject $\mathcal{H}_0$ at the 5% level, but do reject at the 10% level.

3.2.3.1 Does R give what one expects?

Notice that when ϕ has to be estimated as well as $\boldsymbol{\beta}$, we have a situation similar to an unknown variance in the testing of means, going under the title 'small sample t-tests'.

For the example we are looking at, we have

$$\sqrt{W} = \frac{\hat{\boldsymbol{\beta}}_3}{\mathrm{SE}(\hat{\boldsymbol{\beta}}_3)} = \frac{0.31}{0.17} = 1.82 \tag{3.29}$$

This is the same as the number in the '*t*-value' column in the R summary:

```
summary(hosp.glm)
```

```
##
## Call:
## glm(formula = duration ~ age + temp1, family = Gamma(link = log),
##     data = hosp)
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.654096  16.621018  -1.724   0.0987 .
## age           0.014900   0.005698   2.615   0.0158 *
## temp1         0.306624   0.168141   1.824   0.0818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2690233)
##
##     Null deviance: 8.1722  on 24  degrees of freedom
## Residual deviance: 5.7849  on 22  degrees of freedom
## AIC: 142.73
##
## Number of Fisher Scoring iterations: 6
```

However, if $W \sim \chi^2(1)$, then $\sqrt{W} \sim \mathcal{N}(0,1)$, leading to

$$p = 2(1 - \Phi(1.82)) = 0.068 \tag{3.30}$$

This number does not appear in the R summary. The explanation is that if ϕ is estimated, R uses t_{n-p} rather than $\mathcal{N}(0,1)$, leading to

$$p = 2(1 - \Phi_t(1.82)) = 0.082 \tag{3.31}$$

and this number does appear in the R summary.

The use of the t distribution rather than the Gaussian distribution accounts for the extra variability introduced by estimating ϕ. It still uses asymptotic normality as a foundation. In fact, R also allows one to assume the dispersion is known:

```r
summary(hosp.glm, dispersion=  0.2690233)
```

```
##
## Call:
## glm(formula = duration ~ age + temp1, family = Gamma(link = log),
##     data = hosp)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -28.654096  16.621017  -1.724  0.08471 .
## age           0.014900   0.005698   2.615  0.00892 **
## temp1         0.306624   0.168141   1.824  0.06821 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2690233)
##
```

```
##     Null deviance: 8.1722  on 24  degrees of freedom
## Residual deviance: 5.7849  on 22  degrees of freedom
## AIC: 142.73
##
## Number of Fisher Scoring iterations: 6
```

The resulting summary talks of a z-value rather than a t-value, and computes the corresponding p using a Gaussian distribution.

We will use χ^2 tests exclusively, thereby ignoring the variability introduced by the estimation of ϕ.

3.3 Confidence Regions for $\hat{\boldsymbol{\beta}}$

These follow from standard maximum likelihood theory. There are two popular types, which in general are not equivalent.

3.3.1 $(1-\alpha)$ Hessian CR

This is:

$$R^H_{1-\alpha} = \left\{\boldsymbol{\beta} : (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T F(\hat{\boldsymbol{\beta}})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \leq \chi^2_{p,\alpha}\right\}$$

3.3.2 $(1-\alpha)$ *method of support* CR

This is:

$$R_{1-\alpha} = \left\{\boldsymbol{\beta} : \ell(\boldsymbol{\beta}) \geq \ell(\hat{\boldsymbol{\beta}}) - \frac{1}{2}\chi^2_{p,\alpha}\right\}$$

3.4 Issues with GLMs and the Wald Test

3.4.1 Separation

Consider a logistic regression problem with linear predictor: $\eta = \beta_1 + \beta_2 x$. Suppose that the data has the following property (not so unreasonable): the x values of all the points with $y = 0$ are less than the x values of all the points with $y = 1$. This is illustrated in Figure 3.2.

```r
x <- runif( n = 21, min = 0, max = 4 )
y <- as.numeric( x > 1.8 )
plot( x, y, pch = 16 )
```

Problem

What will be the estimated value $\hat{\beta}_2$ of β_2? This question seems unanswerable, but in fact it has a very simple answer. First, consider reparameterising the linear predictor. Define

$$\omega = \beta_2 \tag{3.32}$$

$$x_0 = -\frac{\beta_1}{\beta_2} \tag{3.33}$$

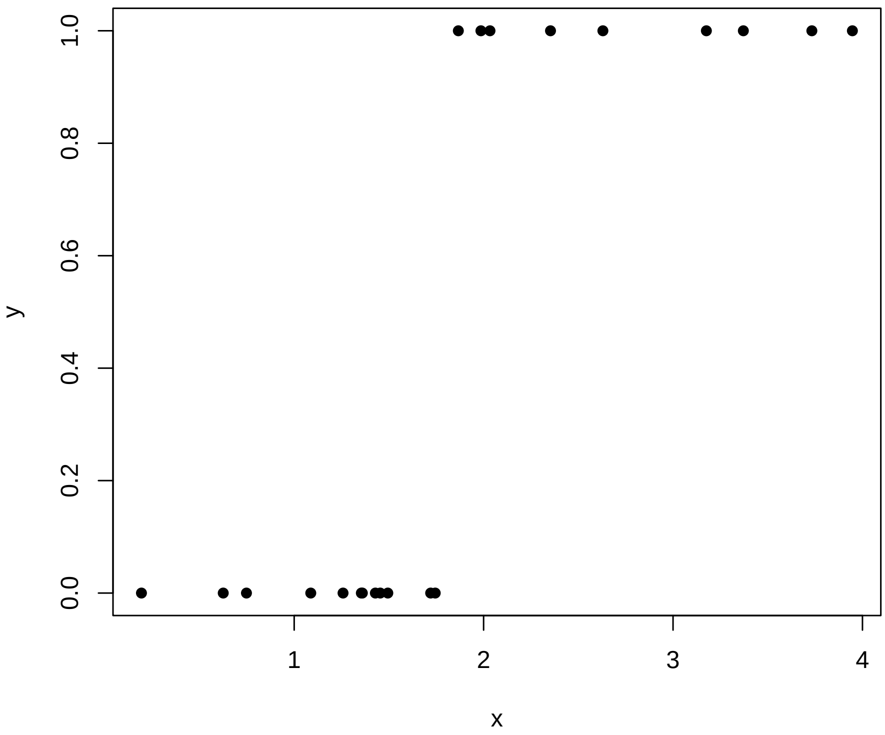


Figure 3.2: Illustration of 'separated' binary data.

The linear predictor is thus:

$$\eta = \omega(x - x_0) \tag{3.34}$$

The expression for the mean, that is, the probability that $y = 1$ given x, is then

$$\pi(x) = \frac{e^{\omega(x-x_0)}}{1 + e^{\omega(x-x_0)}} \tag{3.35}$$

The estimation task is to pick values of ω and x_0 that maximize the probability of the data. Clearly, if we can choose the parameters so that $\pi(x) = 1$ for those points with $y = 1$ and $\pi(x) = 0$ for those points with $y = 0$, we cannot do better: this is the maximum achievable with the model, equivalent to the saturated model in fact. Call this a *perfect fit*.

Now consider the following. Pick x_0 so that it lies between the x with $y = 0$ and the x with $y = 1$. This must be possible because of the initial assumption about the data. Now note that for all the x with $y = 0$, $(x - x_0) < 0$. If we let $\omega \to \infty$, then $\pi(x) \to 0$. On the other hand, for all the x with $y = 1$, $(x - x_0) > 0$, so that as $\omega \to \infty$, $\pi(x) \to 1$. The limiting solution is thus a step function with the step at x_0.

We can therefore achieve a *perfect fit* by allowing $\omega \to \infty$. Unfortunately, this means that the linear predictor is not defined, and, practically speaking, the estimation algorithm will not converge.[2]

[2] A secondary problem is also the fact that all values of x_0 that lie between the two groups of x values are equivalent, and there is thus no way to pick one.

Of course, in this simple case, we can see what is happening, and can anticipate that a step function might be a solution. In general, however, this situation might be hard to detect, and hard to correct, at least within the framework of GLMs. So what is to be done?

Solution

Remember that we set out to model functional relationships. GLMs are one way to do this, by constraining the form of the function in a useful way. In this case, however, they seem to be too limiting. There are two reasons why this might be the case.

One is that the step function solution is appropriate for the data and context with which we are dealing. In this case, the main problem is that our set of functions is poorly parameterised, and includes the step function only as a singular limiting case. There is no real solution for this in the context of GLMs, although more general models could be used.

The other is that the step function solution is not appropriate, and that we really would expect a smoother solution. This is much harder to deal with in the context of classical statistics. We are saying that we expect the value of ω to be finite, larger values becoming less and less probable, until in the limit, an infinite value is impossible. The only real way to deal with this situation is via a prior probability distribution on ω or by imposing some regularising constraint, but those are another story and another course. Within the GLM world, one has simply to be aware of the possibility of separation, and that it may be caused by overly subdividing the data via categorical variables, that is, essentially by overfitting.

3.4.2 Hauck-Donner Effect

A related but independent effect was noted by Hauck and Donner [1976].

Consider

$$W = \frac{\hat{\boldsymbol{\beta}}^2}{\mathrm{Var}[\hat{\boldsymbol{\beta}}]} \tag{3.36}$$

If $\hat{\boldsymbol{\beta}} \to \infty$ (e.g. in cases of separation), then it is quite likely that $\mathrm{Var}[\hat{\boldsymbol{\beta}}] \to \infty$ also. The result can be that the test statistic becomes very small, and in fact tends to zero! Hauck and Donner showed that:

- *Wald's statistic decreased to zero as the distance between the parameter estimate and the null value increased.*

So, as one's null hypothesis gets more and more wrong, the Wald statistic gets smaller and smaller, and one is decreasingly able to reject the increasingly wrong null.

3.4.3 Next Step

We have seen how to set up and describe GLMs, and how to estimate their parameters. We have also seen how to use these parameters to make predictions and generate confidence intervals. We now move on to study how we can evaluate the effectiveness of our models.

Chapter 4

Deviance

4.1 Goodness-of-Fit

We would like to find a measure for *goodness-of-fit*, or, to put it another way, a measure for the *discrepancy* between the data $\boldsymbol{y} \in \mathbb{R}^n$ and the *fit* $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_n) \in \mathbb{R}^n$, where $\hat{\mu}_i = h(\boldsymbol{\beta}^T \boldsymbol{x}_i)$.

First we need to understand how well any GLM could be expected to fit.

4.1.1 The Saturated Model

The log likelihood at the MLE $\hat{\boldsymbol{\beta}}$ is

$$\ell(\hat{\boldsymbol{\beta}}) = \sum_i \left(\frac{y_i \hat{\theta}_i - b(\hat{\theta}_i)}{\phi_i} + c(y_i, \phi_i) \right).$$

The larger $\ell(\hat{\boldsymbol{\beta}})$, the better the fit, but what is *large*? Consider the following. In the GLM, $\boldsymbol{\mu}$, or equivalently $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$, takes values in $\mathbb{R}^n$. However, $\mu_i = h(\boldsymbol{\beta}^T \boldsymbol{x}_i)$. Thus, as $\boldsymbol{\beta} \in \mathbb{R}^n$ varies, $\boldsymbol{\mu} = \{\mu_i\}$ can only trace out a p-dimensional submanifold of $\mathbb{R}^n$: the possible values are constrained by the model structure. (Indeed, as we saw at the beginning, this is the whole point of the model in the first place.)

An upper bound for $\ell(\hat{\boldsymbol{\beta}})$ would therefore be attained by a model that placed less constraints on $\boldsymbol{\mu}$ (since maximisation over a superset necessarily produces a larger value). This can be achieved by simply allowing $\boldsymbol{\mu}$ to range over all of $\mathbb{R}^n$, or in other words by allowing as many parameters as there are data points. This means intuitively that we end up 'joining the dots'.

The maximum likelihood problem then breaks down into n simpler problems, as each term in equation (4.1.1) can be maximised separately. Differentiation with respect to θ_i then gives

$$\ell_i'(\theta_i) = \frac{y_i - b'(\theta_i)}{\phi_i} \tag{4.1}$$

leading to the MLE $\hat{\boldsymbol{\theta}}$, or equivalently, $\hat{\boldsymbol{\mu}}$, given by

$$y_i = b'(\hat{\theta}_i) = \hat{\mu}_i \tag{4.2}$$

This model, in which $\boldsymbol{\mu}$ may vary over the whole of $\mathbb{R}^n$, and there is thus one parameter for each data point, is known as the *saturated model.* Its log likelihood at the MLE value $\hat{\boldsymbol{\mu}}_{\text{sat}}$ is denoted ℓ_{sat}.

This then leads us to the notion of *deviance.*

4.1.2 Deviance

The *deviance* of a GLM is defined as follows:

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\,\phi\,\left(\ell_{\text{sat}} - \ell(\hat{\boldsymbol{\beta}})\right) \tag{4.3}$$

while the *scaled deviance* is defined as

$$D_{\text{sc}}(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\,\left(\ell_{\text{sat}} - \ell(\hat{\boldsymbol{\beta}})\right) \tag{4.4}$$

Now,

$$\ell(\hat{\boldsymbol{\beta}}) = \frac{1}{\phi}\sum_i m_i\,\left(y_i\hat{\theta}_i - b(\hat{\theta}_i)\right) + \sum_i c(y_i, \phi_i) \tag{4.5}$$

with $\hat{\theta}_i = (b')^{-1}(\hat{\mu}_i)$, and $\phi_i = \frac{\phi}{m_i}$; and

$$\ell_{\text{sat}} = \frac{1}{\phi}\sum_i m_i\,\left(y_i\hat{\theta}_{\text{sat},i} - b(\hat{\theta}_{\text{sat},i})\right) + \sum_i c(y_i, \phi_i) \tag{4.6}$$

with $\hat{\theta}_{\text{sat},i} = (b')^{-1}(y_i)$, that is, $\hat{\mu}_{\text{sat},i} = y_i$. We thus have that

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\sum_i m_i \left\{y_i\left(\hat{\theta}_{\text{sat},i} - \hat{\theta}_i\right) - \left(b(\hat{\theta}_{\text{sat},i}) - b(\hat{\theta}_i)\right)\right\} \tag{4.7}$$

The deviance is thus independent of ϕ.

4.1.3 Example Special Cases

We now consider some examples. In these examples, we assume non-grouped data; that is, $m_i = 1$ and $n = M$, where M is the number of groups. The results can easily be generalised to grouped data.

4.1.3.1 Gaussian

We have

- $b(\theta_i) = \frac{1}{2}\theta_i^2$
- $\theta_i = (b')^{-1}(\mu_i) = \mu_i$

We thus find that

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\sum_i \left(y_i(y_i - \hat{\mu}_i) - \left(\frac{1}{2}y_i^2 - \frac{1}{2}\hat{\mu}_i^2 \right) \right) \tag{4.8}$$

$$= 2\sum_i \left(\frac{1}{2}y_i^2 - y_i\hat{\mu}_i + \frac{1}{2}\hat{\mu}_i^2 \right) \tag{4.9}$$

$$= \sum_i (y_i - \hat{\mu}_i)^2 \tag{4.10}$$

But this is just the residual sum of squares (RSS)!

4.1.3.2 Poisson

We have

- $b(\theta_i) = e^{\theta_i}$
- $\theta_i = (b')^{-1}(\mu_i) = \log \mu_i$.

We thus find that

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\sum_i (y_i(\log y_i - \log \hat{\mu}_i) - (y_i - \hat{\mu}_i)) \tag{4.11}$$

$$= 2\sum_i \left(y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i) \right) \tag{4.12}$$

4.1.3.3 Bernoulli

We have

- $b(\theta_i) = \log(1 + e^{\theta_i})$
- $\mu_i = \frac{e^{\theta_i}}{1+e^{\theta_i}}$.
- $\theta_i = \log \frac{\mu_i}{1-\mu_i}$

  However, there is a problem. We have

$$\hat{\theta}_{\text{sat},i} = \log\left(\frac{y_i}{1 - y_i}\right) \tag{4.13}$$

for $y_i \in \{0, 1\}$: the MLE $\hat{\boldsymbol{\theta}}_{\text{sat}}$ is apparently not defined. However, this is easily solved. It is easiest to see if we write the maximum likelihood in terms of $\hat{\boldsymbol{\mu}}$:

$$\ell(\hat{\boldsymbol{\mu}}) = \sum_i \left(y_i \log \hat{\mu}_i + (1 - y_i) \log(1 - \hat{\mu}_i) \right).$$

The saturated log likelihood is therefore

$$\ell_{\text{sat}} = \sum_i \left(y_i \log y_i + (1 - y_i) \log(1 - y_i) \right) = 0$$

for $y_i \in \{0,1\}$ by continuity.

We thus have

$$\begin{aligned} D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) &= -2\,\ell(\hat{\boldsymbol{\beta}}) \\ &= -2 \sum_i \left(y_i \log \hat{\mu}_i + (1-y_i) \log(1-\hat{\mu}_i) \right) \\ &= -2 \left(\sum_{i:y_i=0} \log(1-\hat{\mu}_i) + \sum_{i:y_i=1} \log \hat{\mu}_i \right). \end{aligned}$$

4.2 Asymptotic Properties

In order to use deviance effectively as a measure of goodness-of-fit, we need to be able to analyse its probabilistic behaviour, in order to perform tests, etc. Does deviance have, at least asymptotically, a nice distribution that we can use?

Looking at the form of the deviance,

$$\frac{D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}})}{\phi} = 2\,(\ell_{\text{sat}} - \ell(\hat{\boldsymbol{\beta}})),$$

one might suppose that, to be analogous with the quantities used in likelihood ratio tests, it would be $\chi^2(n-p)$-distributed asymptotically, since the saturated model has n parameters, and the model in which we are interested has p. *If* this were the case, then we could say that if

$$\frac{D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}})}{\phi} > \chi^2_{p,\alpha}$$

then the model *does not fit well*.

Unfortunately, it is not true that $\frac{D(\boldsymbol{Y},\hat{\boldsymbol{\mu}})}{\phi}$ is asymptotically χ^2-distributed in general. This is because the limit theorems that give the χ^2 distribution do not apply when the number of parameters varies as the amount of data increases. Here that is the case, as the dimensionality of the saturated model is not fixed, but n.

In special cases, most notably for the Poisson distribution, or when the $m_i \gg 1$, the asymptotics do hold, and we can use Equation (4.2) as a test of goodness-of-fit. In general, however, this is *not* the case.

Thus, as promising as deviance appears, it cannot serve as a complete replacement for the RSS, even though this is a special case. We will see, however, that deviance is still extremely useful.[1]

[1] One could argue that it is a good thing that deviance cannot be used as a general measure of *goodness-of-fit*, as it forces one to consider comparing one model against another. The idea that there is a measure of goodness-of-fit that applies in the absence of an alternative model is quite a dubious one.

4.3 Pearson Statistic

We now take a slight detour to discuss an alternative measure of goodness-of-fit. This bears the same relationship to deviance that the Wald test bears to the likelihood ratio test: one works in the domain of the probability distribution; and one in its codomain, or in other words, in terms of probability itself.

The *Pearson statistic* is defined as

$$\chi_P^2 = \sum_i m_i \frac{(y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)}.$$

We then see that

$$\begin{aligned}\frac{\chi_P^2}{\phi} &= \sum_i \frac{(y_i - \hat{\mu}_i)^2}{\frac{\phi}{m_i}\mathcal{V}(\hat{\mu}_i)} \\ &= \sum_i \frac{(y_i - \hat{\mu}_i)^2}{\widehat{\mathrm{Var}}[y_i]} \\ &\overset{a}{\sim} \chi^2(n-p).\end{aligned}$$

Hence,

$$\chi_P^2 \overset{a}{\sim} \phi\, \chi^2(n-p).$$

Thus χ_P^2 can be used to measure goodness-of-fit.

4.3.1 Relation to Deviance

Consider $D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}})$ for the Poisson model:

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\sum_i \left(y_i \log \frac{y_i}{\hat{\mu}_i} - (y_i - \hat{\mu}_i) \right).$$

Expanding this as a function of $\boldsymbol{y}$ around $\hat{\boldsymbol{\mu}}$, we find

$$D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) \simeq \sum_i \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}.$$

This is just the normal Pearson statistic for the Poisson distribution.

4.3.2 Pearson Residuals

We will see soon that we can define several types of residual for GLMs. One is defined based on the Pearson statistic, the ‘Pearson residual’:

$$r_i^P = \sqrt{m_i}\frac{y_i - \hat{\mu}_i}{\sqrt{\mathcal{V}(\hat{\mu}_i)}} = \sqrt{\hat{\phi}}\frac{y_i - \hat{\mu}_i}{\sqrt{\widehat{\mathrm{Var}}[y_i]}}.$$

If $y_i \sim \mathcal{N}(\mu_i, \sigma^2)$, with $m_i = 1$, then $\mathcal{V}(\mu_i) = 1$, so that

$$r_i^P = y_i - \hat{\mu}_i = \epsilon_i.$$

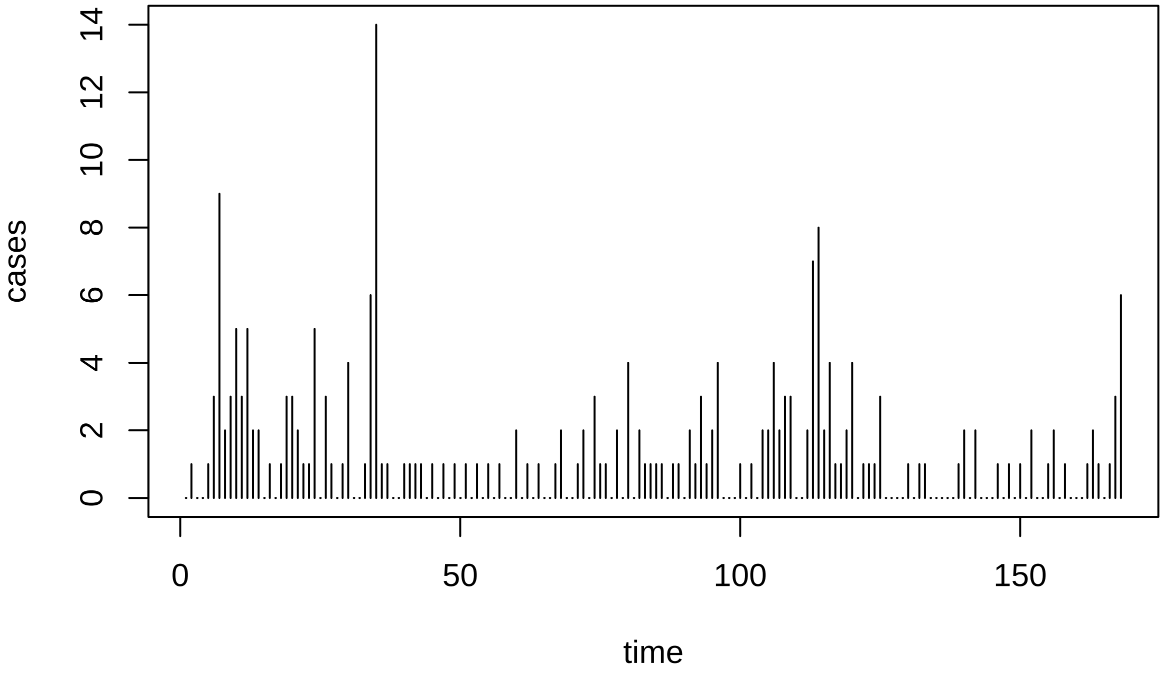
Thus in a linear model, the Pearson residuals are just the ‘usual’ residuals.

4.3.3 Example: US Polio Data

This example concerns a Poisson model, so we can use deviance as a goodness-of-fit measure. We will use deviance and Pearson statistic to test if the model is a good fit.

The R code for this example is presented here:

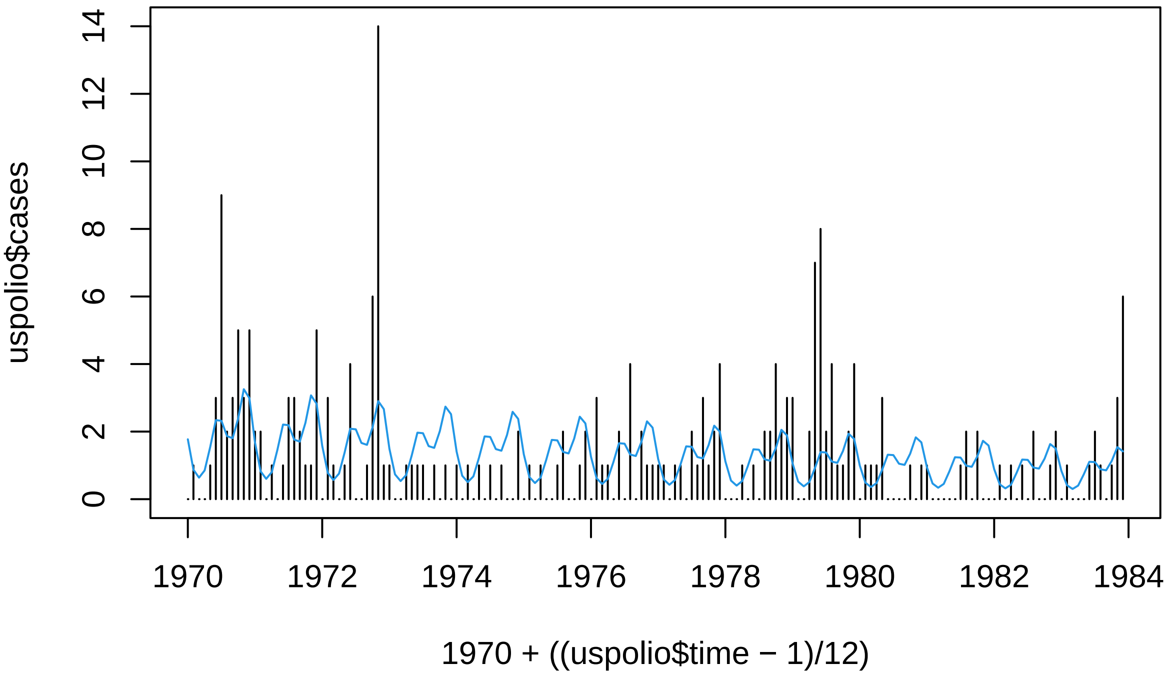
```
# Load and plot the data
library("gamlss.data")
data("polio")
uspolio <- as.data.frame(matrix(c(1:168, t(polio)), ncol=2))
colnames(uspolio) <- c("time", "cases")

plot(uspolio, type="h")
```



```
# Create Poisson GLM which includes time, six-month cycles
# and twelve-month cycles (see Estimation chapter)
polio2.glm<- glm(cases~time + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                 + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                 family=poisson(link=log), data=uspolio)
summary(polio2.glm)
```

```
##
## Call:
## glm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi *
##     time/6)), family = poisson(link = log), data = uspolio)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.557241   0.127303   4.377 1.20e-05 ***
## time                     -0.004799   0.001403  -3.421 0.000625 ***
## I(cos(2 * pi * time/12))  0.137132   0.089479   1.533 0.125384
## I(sin(2 * pi * time/12)) -0.534985   0.115476  -4.633 3.61e-06 ***
## I(cos(2 * pi * time/6))   0.458797   0.101467   4.522 6.14e-06 ***
## I(sin(2 * pi * time/6))  -0.069627   0.098123  -0.710 0.477957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 288.85  on 162  degrees of freedom
## AIC: 557.9
##
## Number of Fisher Scoring iterations: 5
```

```
plot(1970 + ((uspolio$time-1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time-1)/12), polio2.glm$fitted,col=4)
```



```
# Deviance
polio2.glm$dev
```

```
## [1] 288.8549
```

```
# Pearson statistic
sum((uspolio$cases-polio2.glm$fitted)^2 / polio2.glm$fitted)
```

```
## [1] 318.7216
```

```
# Either way, critical value at 5% level
qchisq(0.95, 162)
```

```
## [1] 192.7001
```

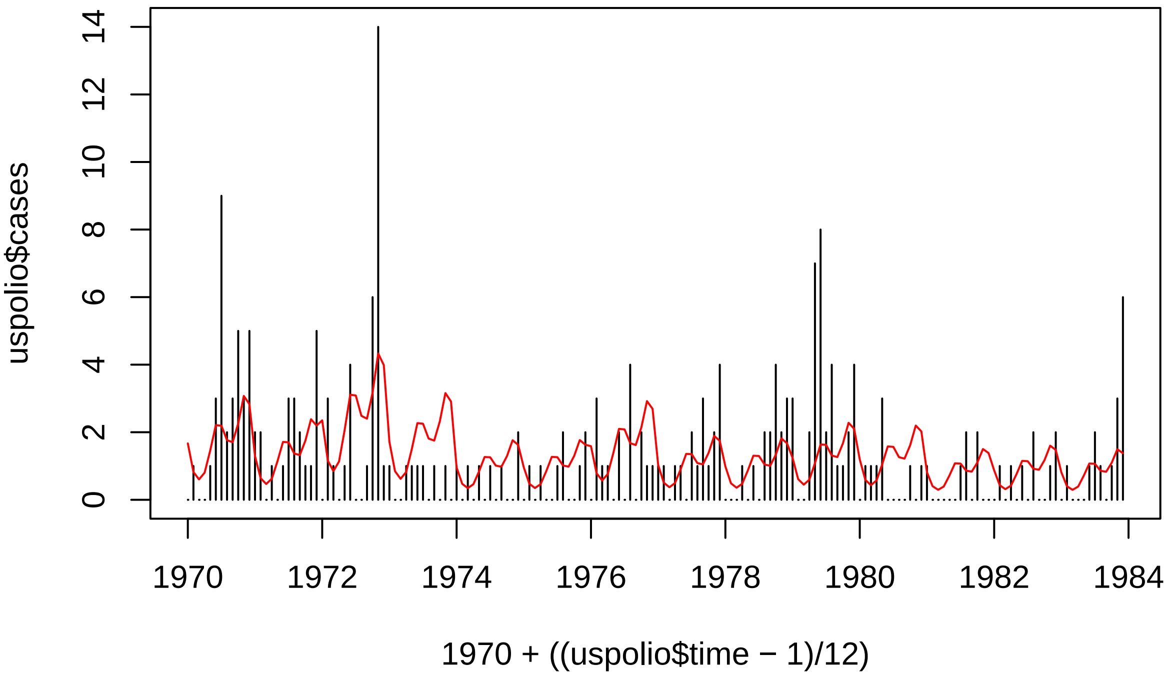
The deviance is 288.86, while the Pearson statistic is 318.72. We have that $\chi^2_{162,0.05} = 192.7$, so either way we reject $\mathcal{H}_0$, which is that the model is adequate, at 5%. (The test is of a model with 6 parameters against a model with 168 parameters, hence the χ^2 distribution has 162 degrees of freedom.)

We can also test the model with temperature data.

```
# Read in temperature data and scale it
temp_data <- rep(c(5.195, 5.138, 5.316, 5.242, 5.094, 5.108, 5.260, 5.153,
                   5.155, 5.231, 5.234, 5.142, 5.173, 5.167), each = 12 )
scaled_temp = 10 * (temp_data - min(temp_data))/(max(temp_data) - min(temp_data))
uspolio$temp = scaled_temp

# Construct GLM
polio3.glm <- glm(cases~time + temp
                  + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                  + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                  family=poisson(link=log), data=uspolio)
summary(polio3.glm)
```

```
##
## Call:
## glm(formula = cases ~ time + temp + I(cos(2 * pi * time/12)) +
##     I(sin(2 * pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 *
##     pi * time/6)), family = poisson(link = log), data = uspolio)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               0.129643   0.186352   0.696 0.486623
## time                     -0.003972   0.001439  -2.761 0.005770 **
## temp                      0.080308   0.023139   3.471 0.000519 ***
## I(cos(2 * pi * time/12))  0.136094   0.089489   1.521 0.128314
## I(sin(2 * pi * time/12)) -0.531668   0.115466  -4.605 4.13e-06 ***
## I(cos(2 * pi * time/6))   0.457487   0.101435   4.510 6.48e-06 ***
## I(sin(2 * pi * time/6))  -0.068345   0.098149  -0.696 0.486218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 276.84  on 161  degrees of freedom
## AIC: 547.88
##
## Number of Fisher Scoring iterations: 5
```

```
plot(1970 + ((uspolio$time-1)/12), uspolio$cases, type="h")
lines(1970 + ((uspolio$time-1)/12), polio3.glm$fitted, col="red")
```



```
# Deviance
polio3.glm$dev
```

```
## [1] 276.8357
```

```
# Pearson statistic
sum((uspolio$cases-polio3.glm$fitted)^2 / polio3.glm$fitted)
```

```
## [1] 279.2618
```

```
# Now, critical value at 5% level
qchisq(0.95, 161)
```

```
## [1] 191.6084
```

Note here that the χ^2 distribution has 161 degrees of freedom. From the results, we still reject $\mathcal{H}_0$, although the deviance and Pearson statistic both reduced.

4.4 Residuals and Diagnostics

Just as there were two types of hypothesis test, and two measures of goodness-of-fit, there are two types of residual typically used for GLMs. These are as follows:

$$\begin{array}{rr}
\text{Deviance residuals} & \text{Pearson residuals} \\
D = \sum_i d_i & \chi^2_P = \sum_i m_i \dfrac{(y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)} \\
r_i^D = \operatorname{sign}(y_i - \hat{\mu}_i)\sqrt{d_i} & r_i^P = \sqrt{m_i}\dfrac{y_i - \hat{\mu}_i}{\sqrt{\mathcal{V}(\hat{\mu}_i)}}
\end{array}$$

Here d_i is the contribution of point (or data group) i to the overall deviance. That is,

$$d_i = 2\, m_i \left\{ y_i \left(\hat{\theta}_{\text{sat},i} - \hat{\theta}_i \right) - \left(b(\hat{\theta}_{\text{sat},i}) - b(\hat{\theta}_i) \right) \right\}$$

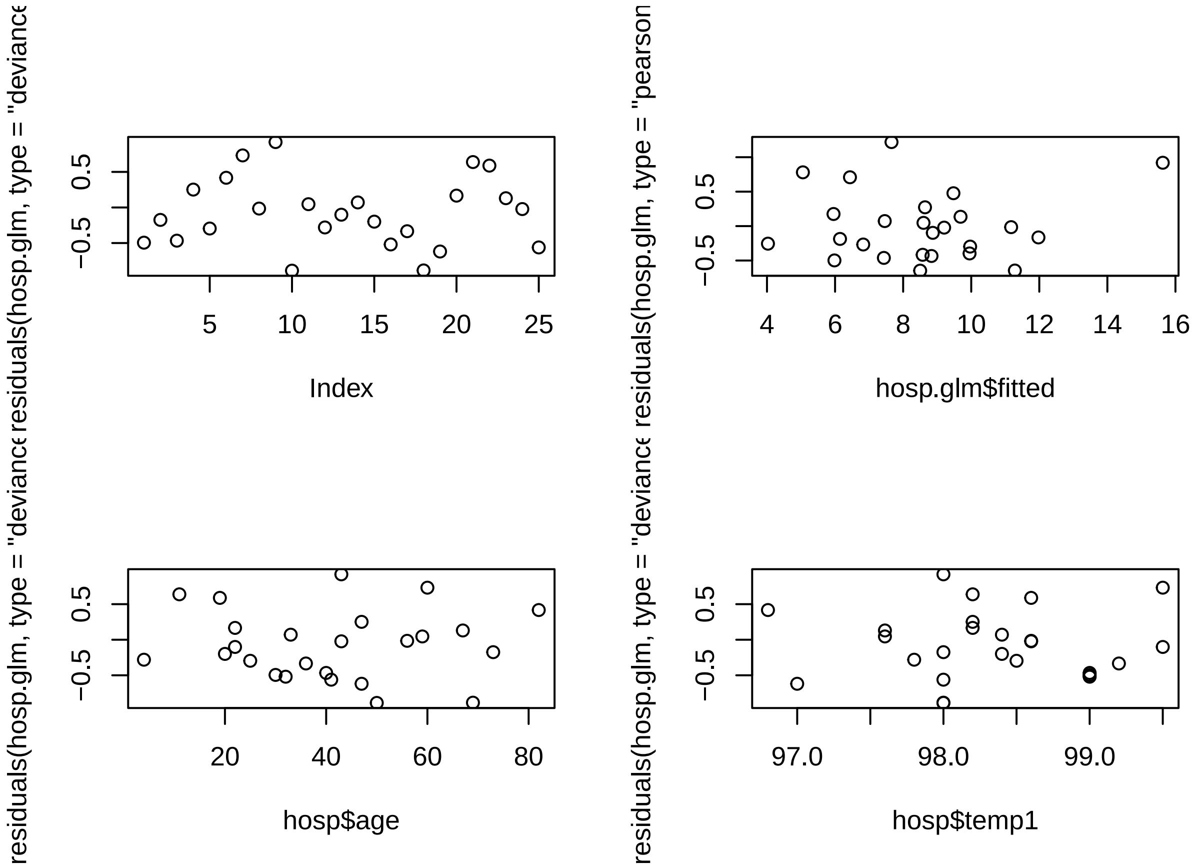
in Equation (4.7).

Just as in a linear model, the r_i^D or r_i^P can be plotted against i or against individual predictors, to detect violations of model assumptions. There is a problem though: neither r_i^D nor r_i^P is Gaussian. This makes 'knowing what to look for' in such plots somewhat tricky.

As a result, many modifications and transformations have been suggested: 'adjusted deviance residuals', 'Anscombe residuals', etc. We will not study these, but content ourselves with checking plots for suspicious looking patterns.

4.4.1 Example: Hospital Stay Data

```r
data(hosp, package="npmlreg")
hosp.glm <- glm(duration~age+temp1, data=hosp, family=Gamma(link=log))

par(mfrow=c(2,2))
plot(residuals(hosp.glm, type="deviance"))
plot(hosp.glm$fitted, residuals(hosp.glm, type="pearson"))
plot(hosp$age, residuals(hosp.glm, type="deviance"))
plot(hosp$temp1, residuals(hosp.glm, type="deviance"))
```



There are no obvious patterns here, but the sample size is quite small, which makes it more difficult. We can also compute and check autocorrelations.

```
cor(residuals(hosp.glm, type="deviance")[1:24], residuals(hosp.glm, type="deviance")[2:2
```

```
## [1] 0.1430499
```

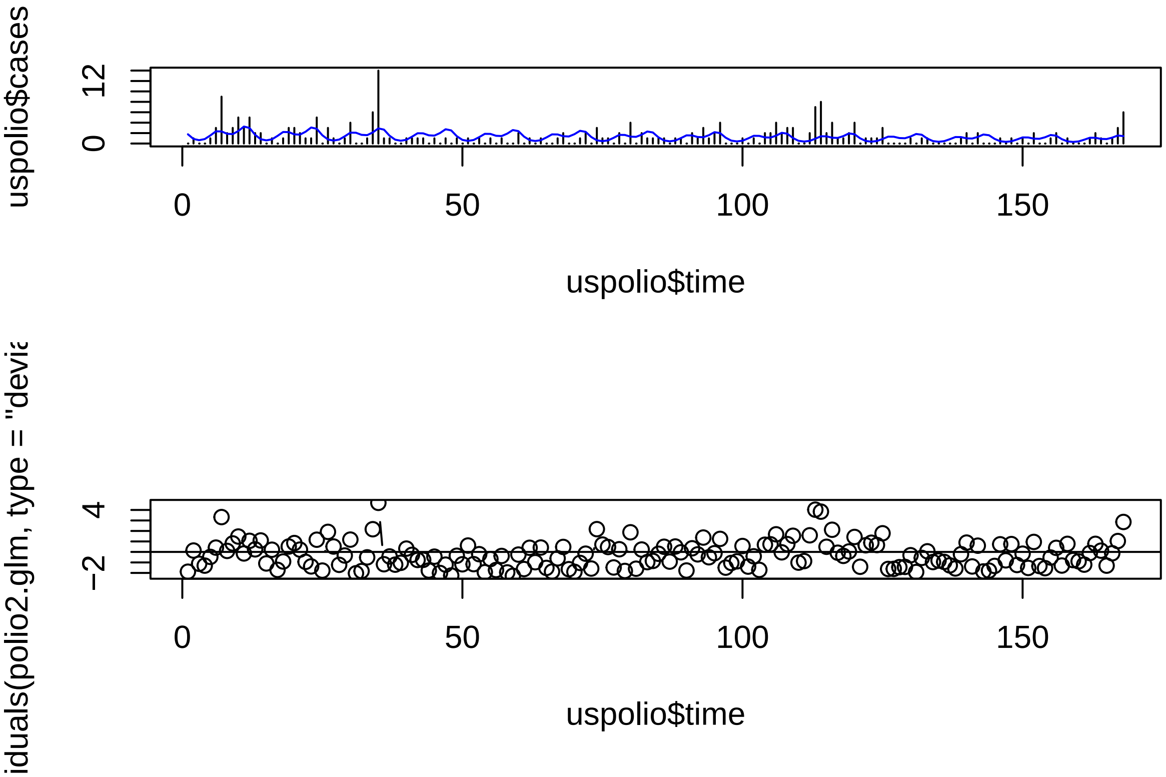
```
cor(residuals(hosp.glm, type="pearson")[1:24], residuals(hosp.glm, type="pearson")[2:25]
```

```
## [1] 0.1444896
```

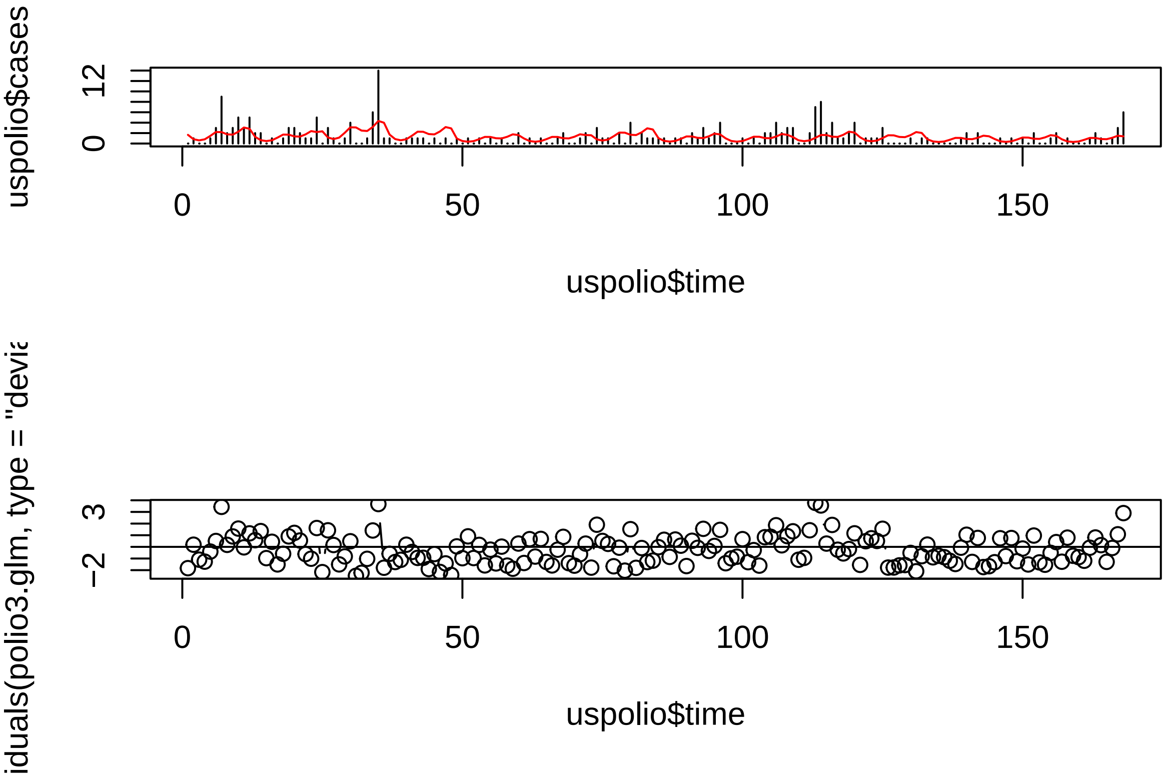
Note that there are still some positive autocorrelations here, but again, the sample size is quite small for an accurate interpretation.

4.4.2 Example: US Polio Data

```
# For polio2 model
par(mfrow=c(2,1))
plot(uspolio$time, uspolio$cases, type="h")
lines(uspolio$time, polio2.glm$fitted, col="blue")
plot(uspolio$time, residuals(polio2.glm, type="deviance"), type="b")
abline(a=0,b=0)
```



```r
# For polio3 model
par(mfrow=c(2,1))
plot(uspolio$time, uspolio$cases, type="h")
lines(uspolio$time, polio3.glm$fitted, col="red")
plot(uspolio$time, residuals(polio3.glm, type="deviance"), type="b")
abline(a=0,b=0)
```



Here there is clearly residual autocorrelation present, so that the independence of different y_i is violated. We also compute the autocorrelations.

```
cor(residuals(polio2.glm, type="deviance")[2:168], residuals(polio2.glm, type="deviance'
```

```
## [1] 0.1677067
```

```
cor(residuals(polio3.glm, type="deviance")[2:168], residuals(polio3.glm, type="deviance'
```

```
## [1] 0.1235241
```

Note that the autocorrelation reduced in the second model but is still high.

4.5 Analysis of Deviance

The analysis of deviance is based on comparing the deviance of a model, not with the 'perfect' saturated model (which in practice is not perfect, since it overfits), but with the deviance of other competing models. These differences of deviances are much more useful in practice than the deviance itself.

Consider two nested GLMs $\tilde{\mathcal{M}} \subset \mathcal{M}$.[2] For example, $\tilde{\mathcal{M}}$ could be the null model with $g(\mu) = \beta_1$ and $\mathcal{M}$ could be our model with $g(\mu) = \boldsymbol{\beta}^T \boldsymbol{x}$, where $\boldsymbol{\beta} \in \mathbb{R}^p$. More generally:

- Let $\mathcal{M}$ be a GLM, the 'full' model;

[2] Model $\tilde{\mathcal{M}}$ is 'nested' in model $\mathcal{M}$ when the parameter space of $\tilde{\mathcal{M}}$ is a subset of the parameter space of $\mathcal{M}$.

- Let $\tilde{\mathcal{M}}$ be a GLM nested in $\mathcal{M}$, the 'reduced' model, with

$$C\boldsymbol{\beta} = \gamma \tag{4.14}$$

with $C \in \mathbb{R}^{s \times p}$.

- Let $\hat{\boldsymbol{\beta}}$ be the MLE under $\mathcal{M}$;
- Let $\tilde{\boldsymbol{\beta}}$ be the MLE under $\tilde{\mathcal{M}}$.

Then we define

$$\begin{aligned} D(\tilde{\mathcal{M}}, \mathcal{M}) &= D(\tilde{\mathcal{M}}) - D(\mathcal{M}) && (4.15) \\ &= 2\,\phi\,\Big(\ell_{\text{sat}} - \ell(\tilde{\boldsymbol{\beta}})\Big) - 2\,\phi\,\Big(\ell_{\text{sat}} - \ell(\hat{\boldsymbol{\beta}})\Big) && (4.16) \\ &= 2\,\phi\,\Big(\ell(\hat{\boldsymbol{\beta}}) - \ell(\tilde{\boldsymbol{\beta}})\Big) && (4.17) \end{aligned}$$

Note that

$$\frac{1}{\phi} D(\tilde{\mathcal{M}}, \mathcal{M}) = 2\Big(\ell(\hat{\boldsymbol{\beta}}) - \ell(\tilde{\boldsymbol{\beta}})\Big) \tag{4.18}$$

This is just the likelihood ratio statistics, and thus

$$\frac{1}{\phi} D(\tilde{\mathcal{M}}, \mathcal{M}) \overset{a}{\sim} \chi^2(s) \tag{4.19}$$

where s is the number of constraints, that is, the difference in the dimensions of the parameter spaces, or the difference in the number of parameters.

From the definition of $D(\tilde{\mathcal{M}}, \mathcal{M})$, we have that

$$D(\tilde{\mathcal{M}}) = D(\tilde{\mathcal{M}}, \mathcal{M}) + D(\mathcal{M}) \tag{4.20}$$

In words, "the discrepancy between the data and $\tilde{\mathcal{M}}$ is equal to the discrepancy between the data and $\mathcal{M}$ plus the discrepancy between $\mathcal{M}$ and $\tilde{\mathcal{M}}$."

4.5.1 Interpretation and Testing

Consider applying this idea in the linear model case. There we have the following:

$$\sum_i (y_i - \tilde{\boldsymbol{\beta}}^T \boldsymbol{x}_i)^2 = D(\tilde{\mathcal{M}}, \mathcal{M}) + \sum_i (y_i - \hat{\boldsymbol{\beta}}^T \boldsymbol{x}_i)^2 \tag{4.21}$$

The left-hand side is the RSS of the reduced model, while the second term on the right-hand side is the RSS of the full model.

In this context, we know we have the partial F-test, based on the statistic:

$$F = \frac{\Big(\text{RSS reduced} - \text{RSS full}\Big)/s}{\Big(\text{RSS full}/(n-p)\Big)} \sim F(s, n-p) \tag{4.22}$$

We then apply this as follows: if $F > F_{s,n-p,\alpha}$, we reject $\mathcal{H}_0 : \tilde{\mathcal{M}}$ in favour of $\mathcal{H}_1 : \mathcal{M}$ at level α.

How should we adapt this to the GLM case? By strict analogy, we have

$$F = \frac{D(\tilde{\mathcal{M}}, \mathcal{M})/s}{\hat{\phi}} = \frac{1}{s}\frac{D(\tilde{\mathcal{M}}, \mathcal{M})}{\hat{\phi}} \sim \frac{1}{s}\chi^2(s) \tag{4.23}$$

where the latter, distributional result is true if we know ϕ, or if we simply ignore the extra variability introduced by estimating it.

In practice, we compute $sF = \frac{D(\tilde{\mathcal{M}},\mathcal{M})}{\hat{\phi}}$ and reject $\mathcal{H}_0 : \tilde{\mathcal{M}}$ if $\frac{D(\tilde{\mathcal{M}},\mathcal{M})}{\hat{\phi}} > \chi^2_{s,\alpha}$.

4.5.2 General Case

More generally we may have a series of nest models: $\mathcal{M}_1 \subset \mathcal{M}_2 \subset \cdots \subset \mathcal{M}_N$, that is, $\mathcal{M}_i \subset \mathcal{M}_{i+1}$ for $i \in [1..(N-1)]$. We can then write down the following telescoping sum:

$$\begin{aligned}
D(\mathcal{M}_1) &= \sum_{i=1}^{N-1} D(\mathcal{M}_i, \mathcal{M}_{i+1}) + D(\mathcal{M}_N) \qquad (4.24)\\
&= \sum_{i=1}^{N-1} \Big(D(\mathcal{M}_i) - D(\mathcal{M}_{i+1})\Big) + D(\mathcal{M}_N) \qquad (4.25)\\
&= \sum_{i=1}^{N-1} D(\mathcal{M}_i) - \sum_{i=2}^{N} D(\mathcal{M}_i) + D(\mathcal{M}_N) \qquad (4.26)\\
&= D(\mathcal{M}_1) - D(\mathcal{M}_N) + D(\mathcal{M}_N) \qquad (4.27)\\
&= D(\mathcal{M}_1) \qquad (4.28)
\end{aligned}$$

A tabular representation of this sum is produced in R when the `anova` command is applies to a fitted GLM, as the next example demonstrates.

4.5.3 Example: Hospital Stay Data

Here analysis of deviance is applied to the full model for the hospital data, with the linear predictor:

$$\eta = \beta_1 + \beta_2\texttt{age} + \beta_3\texttt{temp1} + \beta_4\texttt{wbc1} + \beta_5\texttt{antib} + \beta_6\texttt{bact} + \beta_7\texttt{serv}, \tag{4.29}$$

Gamma family, and log link, as shown below:

```r
data(hosp, package="npmlreg")

# Full model
fit1<- glm(duration~age+temp1+wbc1+antib+bact+serv, data=hosp,
           family=Gamma(link=log))

summary(fit1)
```

```
##
## Call:
## glm(formula = duration ~ age + temp1 + wbc1 + antib + bact +
##     serv, family = Gamma(link = log), data = hosp)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.925401  17.540130  -1.079    0.295
## age           0.010026   0.006636   1.511    0.148
## temp1         0.219006   0.178154   1.229    0.235
## wbc1          0.001654   0.044930   0.037    0.971
## antib        -0.346060   0.242145  -1.429    0.170
## bact          0.075859   0.280639   0.270    0.790
## serv         -0.291875   0.255843  -1.141    0.269
##
## (Dispersion parameter for Gamma family taken to be 0.2661922)
##
##     Null deviance: 8.1722  on 24  degrees of freedom
## Residual deviance: 5.1200  on 18  degrees of freedom
## AIC: 147.57
##
## Number of Fisher Scoring iterations: 10
```

```
# Get the dispersion
summary(fit1)$dispersion
```

```
## [1] 0.2661922
```

```
# Get the deviance table
anova(fit1)
```

```
## Analysis of Deviance Table
##
## Model: Gamma, link: log
##
## Response: duration
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev
## NULL                     24     8.1722
## age    1  1.38428        23     6.7879
## temp1  1  1.00299        22     5.7849
## wbc1   1  0.03236        21     5.7526
## antib  1  0.31246        20     5.4401
## bact   1  0.00017        19     5.4400
## serv   1  0.31995        18     5.1200
```

The resulting anova table has the significance shown in the table in Figure 4.1.

Model	Deviance	Resid. df	Resid. deviance
M_1, 'NULL'			$D(M_1)$
M_2, age	$D(M_1,M_2)$		$D(M_2)$
$\vdots$	$\vdots$		$\vdots$
M_7, serv	$D(M_6,M_7)$		$D(M_7)$

Figure 4.1: R output from an 'anova' command on a GLM.

- Each row represents the model containing the predictors in that row and all the previous rows.

- From the definition of $D(\tilde{\mathcal{M}}, \mathcal{M})$, in each row, the sum of the `Resid. deviance` and the `Deviance` gives the `Resid. deviance` in the row above.

- If the `anova` command is run with the argument `test = "Chisq"`, then there will be an extra column in the table (see the code below). This represents the p value of a χ^2 test applied to the `deviance` in that row. It therefore tests the model in row above against the model in the row in which it appears.

```
anova(fit1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: Gamma, link: log
##
## Response: duration
##
## Terms added sequentially (first to last)
##
##
##       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                     24     8.1722
## age    1  1.38428        23     6.7879  0.02258 *
## temp1  1  1.00299        22     5.7849  0.05224 .
## wbc1   1  0.03236        21     5.7526  0.72735
## antib  1  0.31246        20     5.4401  0.27862
## bact   1  0.00017        19     5.4400  0.97990
## serv   1  0.31995        18     5.1200  0.27293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can use the results in the deviance table to perform different tests. Below are some examples.

4.5.3.1 Test Problem 1

Test $\mathcal{H}_0 : \mathcal{M}_1$, the null model where $g(\eta) = \beta_1$, against $\mathcal{H}_1 : \mathcal{M}_7$, the full model. This is the analogue of a full F test.

From the table we read that $D(\mathcal{M}_1) = 8.17$ while $D(\mathcal{M}_7) = 5.12$. We also see from the R output that $\hat{\phi} = 0.27$. We therefore have

$$\frac{D(\mathcal{M}_1, \mathcal{M}_7)}{\hat{\phi}} = \frac{8.17 - 5.12}{0.27} = \frac{3.05}{0.27} = 11.47.$$

This quantity is approximately $\chi^2(6)$ distributed as $\mathcal{M}_7$ has 6 more parameters than $\mathcal{M}_1$. We find the p value using `R`:

$$p = 1 - \texttt{pchisq}(11.47, 6) = 0.075.$$

We would thus just reject $\mathcal{H}_0$ at the 7.5% level, quite weak evidence that the model explains anything at all.

Note the intuition here. If $D(\tilde{\mathcal{M}}, \mathcal{M})$ is large, it means that the more complex model $\mathcal{M}$ is doing a much better job at explaining the data than the simpler model $\tilde{\mathcal{M}}$. If it is enough better, then we will reject $\mathcal{H}_0 : \tilde{\mathcal{M}}$. At the same time, when $D(\tilde{\mathcal{M}}, \mathcal{M})$ is large, it means that the χ^2 value will be large and thus that the p value will be small, meaning that there is a small probability of finding our value of $D(\tilde{\mathcal{M}}, \mathcal{M})$ or greater, if $\mathcal{H}_0$ is true and $\tilde{\mathcal{M}}$ is the correct model. Thus the smaller the p value, the less favourably we look on the null hypothesis and the more significant the level at which we can reject (more significance but smaller level number: we need $p < 0.05$ to reject at 5%, but $p < 0.01$ to reject at 1%).

4.5.3.2 Test Problem 2

Now we take $\mathcal{H}_0 : \mathcal{M}_2$, with $\eta = \beta_1 + \beta_2\texttt{age}$, and $\mathcal{H}_1 : \mathcal{M}_3$, with $\eta = \beta_1 + \beta_2\texttt{age} + \beta_3\texttt{temp1}$. These correspond to successive levels in the table, and so we can read the deviance directly from the table. We then find

$$\frac{D(\mathcal{M}_2, \mathcal{M}_3)}{\hat{\phi}} = \frac{1.003}{0.27} = 3.77.$$

This quantity is (approximately) $\chi^2(1)$ distributed, as there is one parameter difference between the two models. The p value is

$$p = 1 - \texttt{pchisq}(3.77, 1) = 0.052.$$

Thus, *given* `age`, there is some weak evidence to support adding `temp1` to the model: we can reject $\mathcal{H}_0$ at the 5.2% level.

4.5.3.3 Test Problem 3

Now we take $\mathcal{H}_0 : \mathcal{M}_3$, with $\eta = \beta_1 + \beta_2\texttt{age} + \beta_3\texttt{temp1}$, and $\mathcal{H}_1 : \mathcal{M}_7$. Reading from the table, we find $D(\mathcal{M}_7) = 5.12$, while $D(\mathcal{M}_3) = 5.79$. This gives

$$\frac{D(\mathcal{M}_3, \mathcal{M}_7)}{\hat{\phi}} = \frac{5.79 - 5.12}{0.27} = 2.50.$$

This quantity is (approximately) $\chi^2(4)$ distributed. The p value is

$$p = 1 - \texttt{pchisq}(2.50, 4) = 0.65.$$

There is thus no evidence at all for including any variable beyond `age` and `temp1`.

Chapter 5

Quasi-Likelihood Methods

5.1 Dispersion

Recall the exponential dispersion family

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

Assume grouping has been taken care of, that is $\phi_i = \phi/m_i$, and so the grouped data distribution is

$$P(y_i|\theta_i,\phi_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi_i} + c(y_i,\phi_i,m_i)\right\},$$

where y_i is the average response in group i.

We know a basic property (directly from GLM theory):

$$\mathrm{Var}(y_i|\theta_i,\phi_i) = \phi_i\mathcal{V}(\mu_i) = \phi\,\mathcal{V}(\mu_i)/m_i.$$

So, the dispersion ϕ scales the variance but does not affect $E(y_i|\theta_i,\phi_i) = \mu_i = h(\boldsymbol{\beta}^T\boldsymbol{x}_i)$.

We know that the function $\mathcal{V}(\mu)$ is characteristic for the response distribution at play:

Y	$\mathcal{V}(\mu)$	ϕ
$\mathcal{N}(\mu,\sigma^2)$	1	σ^2
Bernoulli(p)	$p(1-p)$	1
Poisson(μ)	μ	1
Gamma(μ,ν)	μ^2	$1/\nu$
$IG(\mu,\sigma^2)$	μ^3	σ^2

What is then the relevance of dispersion?

For estimation of $\boldsymbol{\beta}$, we note from (2.17):

$$S(\boldsymbol{\beta}) = \frac{1}{\phi}\sum_{i=1}^{n} m_i \, \frac{y_i - \mu_i}{\mathcal{V}(\mu_i)} \, h'(\eta_i) \, \boldsymbol{x}_i \overset{!}{=} 0,$$

where $\overset{!}{=} 0$ means we set the left-hand side to zero and solve for $\boldsymbol{\beta}$. In this case, ϕ cancels out when setting the score-function to 0. Hence, dispersion is irrelevant for the estimation of $\boldsymbol{\beta}$.

But for the variance of $\hat{\boldsymbol{\beta}}$, we have

$$\text{Var}(\hat{\boldsymbol{\beta}}) = F^{-1}_{(\phi)}(\hat{\boldsymbol{\beta}}) = \left[\frac{1}{\phi}\sum_{i=1}^{n} m_i \{\ldots\}\right]^{-1} = \phi F^{-1}_{(\phi=1)}(\hat{\boldsymbol{\beta}}),$$

where $F_{(\phi)}(\cdot)$ is the (expected) Fisher information when using dispersion value ϕ.

This result implies that a dispersion of ϕ will inflate all standard errors, $SE(\hat{\beta}_j)$, by a factor $\sqrt{\phi}$ (when compared to using $\phi = 1$).

Estimation of dispersion can also be motivated through goodness-of-fit statistics:

- via Pearson goodness-of-fit statistic $\chi^2_P = \sum_{i=1}^{n} m_i \frac{(y_i - \hat{\mu}_i)^2}{\mathcal{V}(\hat{\mu}_i)}$:

$$\begin{aligned} \chi^2_P &\overset{a}{\sim} \phi \chi^2(n-p) \\ E(\chi^2_P) &= \phi \, (n-p), \end{aligned}$$

suggesting

$$\hat{\phi}_{\text{Pearson}} = \frac{\chi^2_P}{n-p}.$$

* via Deviance, $D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) = 2\,\phi\,(\ell_{sat} - \ell(\hat{\boldsymbol{\beta}}))$:

$$\begin{aligned} D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}}) &\overset{a}{\sim} \phi \, \chi^2(n-p) \\ E(D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}})) &= \phi \, (n-p), \end{aligned}$$

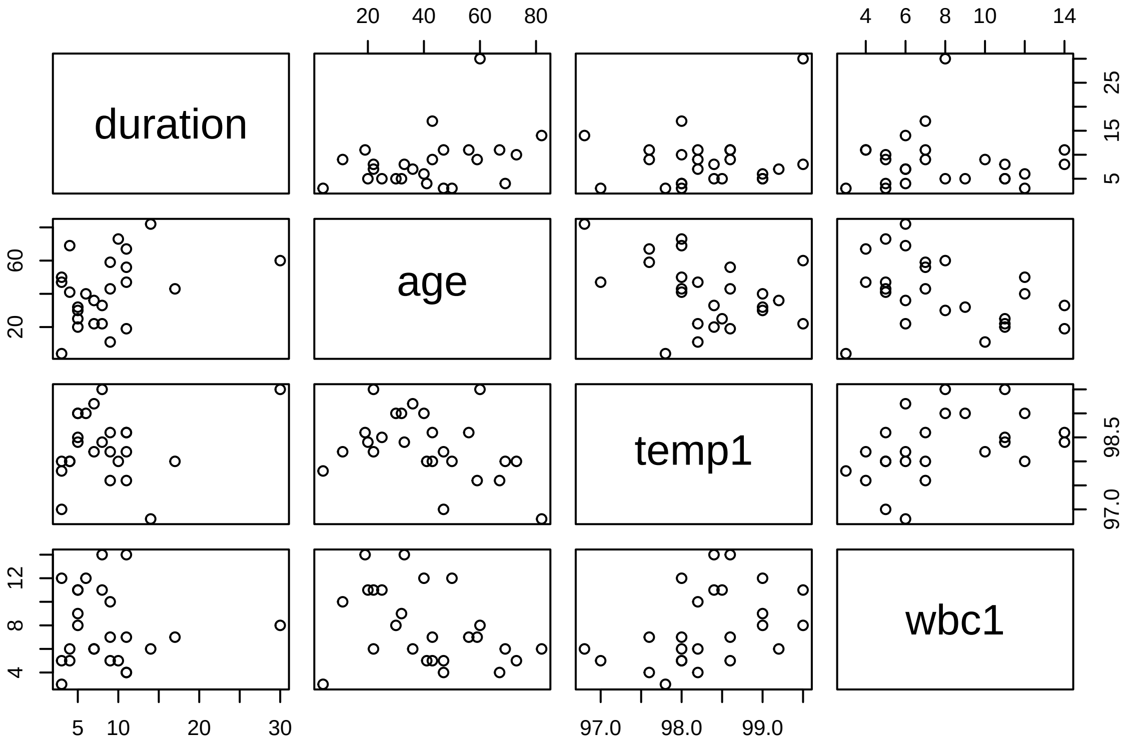
suggesting

$$\hat{\phi}_{\text{dev}} = \frac{D(\boldsymbol{Y}, \hat{\boldsymbol{\mu}})}{n-p}.$$

The notation a in the distributional expressions highlights that these are just approximations. For the deviance, we know that the approximation can be very poor! Therefore the deviance-based estimate is sometimes called the “quick-and-dirty” dispersion estimate.

5.1.1 Example: Hospital Stay Data

```r
require(npmlreg)
data(hosp)
plot(hosp[,c("duration","age","temp1","wbc1")])
```



```r
hosp.glm <- glm(duration~age+temp1, data=hosp, family=Gamma(link=log))
summary(hosp.glm)
```

```
##
## Call:
## glm(formula = duration ~ age + temp1, family = Gamma(link = log),
##     data = hosp)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.654096  16.621018  -1.724   0.0987 .
## age           0.014900   0.005698   2.615   0.0158 *
## temp1         0.306624   0.168141   1.824   0.0818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2690233)
##
##     Null deviance: 8.1722  on 24  degrees of freedom
## Residual deviance: 5.7849  on 22  degrees of freedom
## AIC: 142.73
##
## Number of Fisher Scoring iterations: 6
```

Dispersion estimate (Pearson):

```r
hosp.disp <- 1/(hosp.glm$df.res) * sum((hosp$duration-hosp.glm$fitted)^2/(hosp.glm$fitte
hosp.disp
```

```
## [1] 0.2690233
```

Dispersion estimate (Deviance):

```
hosp.glm$deviance/hosp.glm$df.residual
```

```
## [1] 0.2629518
```

```
summary(hosp.glm)$cov.unscaled  # F^(-1) under phi=1
```

```
##              (Intercept)           age         temp1
## (Intercept) 1026.8932335 -0.1386065114 -10.387121099
## age           -0.1386065  0.0001206901   0.001359292
## temp1        -10.3871211  0.0013592917   0.105088741
```

```
sqrt(hosp.disp) * sqrt(diag(summary(hosp.glm)$cov.unscaled))
```

```
## (Intercept)         age       temp1
## 16.62101763  0.00569811  0.16814078
```

```
summary(hosp.glm)$coef[,"Std. Error"]
```

```
## (Intercept)         age       temp1
## 16.62101763  0.00569811  0.16814078
```

5.2 Overdispersion

Recall, for the Poisson model, one has $\phi = 1$, i.e.

$$\text{Var}(y_i|\theta_i) = 1 \times \mathcal{V}(\mu_i) = \mathcal{V}(\mu_i) = \mu_i = E(y_i|\theta_i).$$

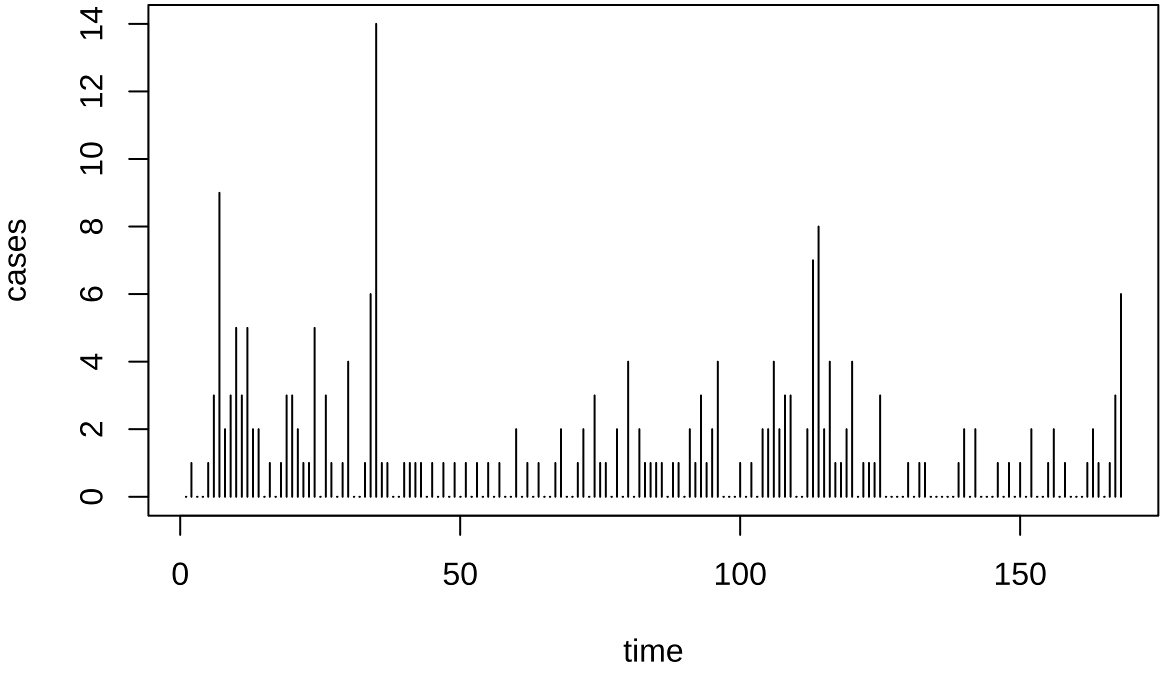
Thus,

$$\frac{\text{Var}(y_i|\theta_i)}{E(y_i|\theta_i)} = 1,$$

a property referred to as "equidispersion".

5.2.1 Example: US Polio Data

```
require(gamlss.data)
data(polio)
uspolio <- as.data.frame(matrix(c(1:168, t(polio)), ncol=2))
colnames(uspolio) <- c("time", "cases")
plot(uspolio, type="h")
```



Simple linear model:

```r
polio.glm <- glm(cases ~ time, family=poisson(link=log), data=uspolio)
summary(polio.glm)
```

```
##
## Call:
## glm(formula = cases ~ time, family = poisson(link = log), data = uspolio)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.626639   0.123641   5.068 4.02e-07 ***
## time        -0.004263   0.001395  -3.055  0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 333.55  on 166  degrees of freedom
## AIC: 594.59
##
## Number of Fisher Scoring iterations: 5
```

Looking at the summary, we see $\phi = 1$. But let's get a quick dispersion estimate:

```r
polio.disp <- 333.55/166
polio.disp
```

```
## [1] 2.009337
```

Now with seasonal model (annual and semi-annual cycles):

```
polio2.glm <- glm(cases ~ time + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                  + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                  family=poisson(link=log), data=uspolio)
summary(polio2.glm)
```

```
## 
## Call:
## glm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 * 
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi * 
##     time/6)), family = poisson(link = log), data = uspolio)
## 
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)    
## (Intercept)               0.557241   0.127303   4.377 1.20e-05 ***
## time                     -0.004799   0.001403  -3.421 0.000625 ***
## I(cos(2 * pi * time/12))  0.137132   0.089479   1.533 0.125384    
## I(sin(2 * pi * time/12)) -0.534985   0.115476  -4.633 3.61e-06 ***
## I(cos(2 * pi * time/6))   0.458797   0.101467   4.522 6.14e-06 ***
## I(sin(2 * pi * time/6))  -0.069627   0.098123  -0.710 0.477957    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 288.85  on 162  degrees of freedom
## AIC: 557.9
## 
## Number of Fisher Scoring iterations: 5
```

```
polio2.disp <- 288.65/162
polio2.disp
```

```
## [1] 1.78179
```

We see that the dispersion reduces when improving the fit of the model, but it is unlikely to 'disappear' fully.

In the example above, there is **overdispersion**, i.e. more dispersion in the data than supported by the model assumption (e.g. $\phi = 1$).

Overdispersion can arise for all one-parameter exponential family distributions (Poisson, Bernoulli/Binomial,...).

Possible reasons for overdispersion can include:

- Model misspecification (such as omitted covariates)

- Latent clusters/subpopulations in the data ("unobserved heterogeneity")

What is the impact of overdispersion?

- There is no impact on $\hat{\beta}$.
- The standard error, $SE(\hat{\beta}_j)$, when estimated using $\phi = 1$, is underestimated by the factor $\sqrt{\phi}$ (with ϕ denoting the "true" ϕ).

As a consequence,

- Confidence intervals will be too small;
- p-values will be too small;
- Hence, significances overstated and potentially wrong decisions.

Fortunately, there is a simple remedy to this problem:

- Fit a Poisson or Bernoulli/Binomial model as usual.
- Estimate ϕ as discussed above, yielding $\hat{\phi}$.
- Multiply all standard errors by $\sqrt{\hat{\phi}}$.

De facto, this amounts to the model:

$$\mu_i = h(\boldsymbol{\beta}^T \boldsymbol{x}_i)$$
$$\text{Var}(y_i) = \phi\,\mathcal{V}(\mu_i),$$

where ϕ is now *allowed to vary* to capture the actual dispersion in the data.

The corresponding "quasi-score function" (in which clearly ϕ cancels out) of this model is:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} m_i \frac{y_i - \mu_i}{\phi\,\mathcal{V}(\mu_i)} h'(\eta_i)\ \boldsymbol{x}_i \overset{!}{=} 0$$

and the variance estimate is:

$$\widehat{\text{Var}}(\hat{\boldsymbol{\beta}}) = \hat{\phi}\,F^{-1}(\hat{\boldsymbol{\beta}}).$$

Note that doing all this requires no actual distributional assumption (and there could also not be any: The 'Poisson'-type EDF with $\phi > 1$ is *not* a valid pdf!)

Estimation methods using quasi-score functions are also referred to as **quasi-likelihood methods**.

5.2.2 Example: US Polio Data

Let us firstly produce the Pearson-type dispersion estimates.

```
polio.phi <- 1/(polio.glm$df.res) * sum((uspolio$cases-polio.glm$fitted)^2/(polio.glm$fi
polio.phi
```

```
## [1] 2.481818
```

```
polio2.phi <- 1/(polio2.glm$df.res) * sum((uspolio$cases-polio2.glm$fitted)^2/(polio2.gl
polio2.phi
```

```
## [1] 1.967417
```

Then adjust the standard errors:

```
polio.se <- sqrt(polio.phi)*summary(polio.glm)$coef[,2]
polio.se
```

```
## (Intercept)        time
## 0.194781773 0.002198181
```

```
polio2.se <- sqrt(polio2.phi)*summary(polio2.glm)$coef[,2]
polio2.se
```

```
##             (Intercept)                     time I(cos(2 * pi * time/12))
##             0.178560577              0.001967754              0.125507133
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##             0.161971821              0.142321901              0.137631208
```

These results can be obtained in R directly through the use of the `quasipoisson` (or, similarly, `quasibinomial`) family:

```
polio.qglm <- glm(cases ~ time, family=quasipoisson(link=log), data=uspolio)

summary(polio.qglm)
```

```
##
## Call:
## glm(formula = cases ~ time, family = quasipoisson(link = log),
##     data = uspolio)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.626639   0.194788   3.217  0.00156 **
## time        -0.004263   0.002198  -1.939  0.05415 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.481976)
##
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 333.55  on 166  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
polio2.qglm <- glm(cases ~ time + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                     + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                   family=quasipoisson(link=log), data=uspolio)

summary(polio2.qglm)
```

```
## 
## Call:
## glm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 * 
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi * 
##     time/6)), family = quasipoisson(link = log), data = uspolio)
## 
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)   
## (Intercept)               0.557241   0.178566   3.121  0.00214 **
## time                     -0.004799   0.001968  -2.439  0.01583 * 
## I(cos(2 * pi * time/12))  0.137132   0.125511   1.093  0.27620   
## I(sin(2 * pi * time/12)) -0.534985   0.161977  -3.303  0.00118 **
## I(cos(2 * pi * time/6))   0.458797   0.142326   3.224  0.00153 **
## I(sin(2 * pi * time/6))  -0.069627   0.137635  -0.506  0.61363   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for quasipoisson family taken to be 1.96754)
## 
##     Null deviance: 343.00  on 167  degrees of freedom
## Residual deviance: 288.85  on 162  degrees of freedom
## AIC: NA
## 
## Number of Fisher Scoring iterations: 5
```

One can test for the presence of overdispersion by comparing $\hat{\phi}$ to $\chi^2_{n-p,\alpha}/(n-p)$. For instance, for the two models under consideration, the critical values for H_0: $\phi = 1$ at the 5% level of significance would be

```
qchisq(0.95, polio.glm$df.res)/polio.glm$df.res
```

```
## [1] 1.187132
```

```
qchisq(0.95, polio2.glm$df.res)/polio2.glm$df.res
```

```
## [1] 1.189507
```

so that H_0 would be rejected in both cases.

5.3 Generalized Estimating Equations

The quasi-likelihood techniques discussed in the previous subsection motivate a more general concept. First, we rewrite the score function in matrix form:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\phi_i \, \mathcal{V}(\mu_i)} h'(\eta_i) \, \boldsymbol{x}_i = \boldsymbol{X}^T \boldsymbol{D} \boldsymbol{\Sigma}^{-1} (\boldsymbol{Y} - \boldsymbol{\mu}),$$

which is known from Section 2.7.1 and where as usual:

$$\begin{aligned}
\phi_i &= \phi / m_i, \\
\eta_i &= \boldsymbol{\beta}^T \boldsymbol{x}_i, \\
\boldsymbol{D} &= \operatorname{diag}(h'(\eta_i)), \text{ and} \\
\boldsymbol{\Sigma} &= \operatorname{diag}(\operatorname{Var}(y_i)) = \operatorname{diag}(\phi_i \, \mathcal{V}(\mu_i)).
\end{aligned}$$

The idea is now to replace the matrix $\boldsymbol{\Sigma}$ (the shape of which was originally motivated directly by GLM properties) by *any* appropriate "working covariance" matrix $\boldsymbol{\Sigma} = \operatorname{Var}(\boldsymbol{Y})$, in this manner trying to capture any correlation structures in the data as correctly as possible. From here, we would compute (just as for GLMs):

$$F(\boldsymbol{\beta}) = \boldsymbol{X}^T \boldsymbol{D}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{D} \boldsymbol{X},$$

allowing us to estimate the variance of $\hat{\boldsymbol{\beta}}$ as

$$\operatorname{Var}(\hat{\boldsymbol{\beta}}) \approx F^{-1}(\hat{\boldsymbol{\beta}}).$$

Important theoretical result: Under some regularity conditions, the estimator $\hat{\boldsymbol{\beta}}$ is consistent and asymptotically normal even if $\boldsymbol{\Sigma}$ is wrong!

5.3.1 Example: US Polio Data

We first use GEEs to provide an equivalent estimate of the quasipoisson approach.

```
require(gee)
```

```
## Loading required package: gee
```

```
uspolio.gee <- gee(cases ~ time
                   + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                   + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                   family=poisson(link=log), id=rep(1,168),
                   corstr = "independence", data=uspolio)
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

## running glm to get initial regression estimate

##            (Intercept)                    time I(cos(2 * pi * time/12))
##            0.557240558            -0.004798661              0.137131634
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##           -0.534985461             0.458797164             -0.069627044
```

```
uspolio.gee
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Logarithm
##  Variance to Mean Relation: Poisson
##  Correlation Structure:     Independent
##
## Call:
## gee(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi *
##     time/6)), id = rep(1, 168), data = uspolio, family = poisson(link = log),
##     corstr = "independence")
##
## Number of observations :  168
##
## Maximum cluster size   :  168
##
##
## Coefficients:
##              (Intercept)                     time I(cos(2 * pi * time/12))
##              0.557240556             -0.004798661              0.137131637
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##             -0.534985464              0.458797164             -0.069627041
##
## Estimated Scale Parameter:  1.967417
## Number of Iterations:  1
##
## Working Correlation[1:4,1:4]
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
##
## Returned Error Value:
## [1] 0
```

```
summary(uspolio.gee)$coef
```

```
##                              Estimate Naive S.E.    Naive z  Robust S.E.
## (Intercept)               0.557240556  0.1785640  3.1206774 1.166422e-16
## time                     -0.004798661  0.0019678 -2.4385923 4.904902e-18
## I(cos(2 * pi * time/12))  0.137131637  0.1255114  1.0925831 1.219796e-16
## I(sin(2 * pi * time/12)) -0.534985464  0.1619769 -3.3028513 4.739094e-16
## I(cos(2 * pi * time/6))   0.458797164  0.1423255  3.2235766 2.333805e-16
## I(sin(2 * pi * time/6))  -0.069627041  0.1376347 -0.5058827 4.818703e-17
##                               Robust z
## (Intercept)               4.777351e+15
## time                     -9.783399e+14
## I(cos(2 * pi * time/12))  1.124218e+15
## I(sin(2 * pi * time/12)) -1.128877e+15
## I(cos(2 * pi * time/6))   1.965876e+15
## I(sin(2 * pi * time/6))  -1.444933e+15
```

But then, there is very likely serial correlation! So

```
# uspolio.gee2 <- gee(cases ~ time
#                     + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
#                     + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
#                     family=poisson(link=log), id=rep(1,168),
#                     data=uspolio, corstr="AR-M", Mv=1)
```

Does not fit! We can try another package. . . .

```
require(geepack)
```

```
## Loading required package: geepack
```

```
uspolio.gee3 <- geeglm(cases ~ time
                       + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                       + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                       family=poisson(link=log), id=rep(1,168),
                       corstr="ar1", data=uspolio)

uspolio.gee3
```

```
##
## Call:
## geeglm(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi *
##     time/6)), family = poisson(link = log), data = uspolio, id = rep(1,
##     168), corstr = "ar1")
##
## Coefficients:
##              (Intercept)                     time I(cos(2 * pi * time/12))
##              0.533915758             -0.004450645              0.143705437
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##             -0.529102025              0.455450494             -0.065291103
##
## Degrees of Freedom: 168 Total (i.e. Null);  162 Residual
##
```

```
## Scale Link:                   identity
## Estimated Scale Parameters:  [1] 1.892594
##
## Correlation:  Structure = ar1    Link = identity
## Estimated Correlation Parameters:
##     alpha
## 0.2790038
##
## Number of clusters:   1   Maximum cluster size: 168
```

… or better simplify the model. For instance, let us assume we have an AR(1) correlation structure within each year, but different years are independent:

```
require(gee)
id = rep(1:14,each=12)
id
```

```
##   [1]  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  3
##  [26]  3  3  3  3  3  3  3  3  3  3  3  4  4  4  4  4  4  4  4  4  4  4  4  5  5
##  [51]  5  5  5  5  5  5  5  5  5  5  6  6  6  6  6  6  6  6  6  6  6  6  7  7  7
##  [76]  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  9
## [101]  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11
## [126] 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13
## [151] 13 13 13 13 13 13 14 14 14 14 14 14 14 14 14 14 14 14
```

```
uspolio.gee4 <- gee(cases ~ time
                    + I(cos(2*pi*time/12)) + I(sin(2*pi*time/12))
                    + I(cos(2*pi*time/6)) + I(sin(2*pi*time/6)),
                    family=poisson(link=log), id=id,
                    data=uspolio, corstr="AR-M", Mv=1)
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
```

```
##              (Intercept)                     time I(cos(2 * pi * time/12))
##              0.557240558             -0.004798661              0.137131634
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##             -0.534985461              0.458797164             -0.069627044
```

```
uspolio.gee4
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Logarithm
##  Variance to Mean Relation: Poisson
##  Correlation Structure:     AR-M , M = 1
##
```

```
## Call:
## gee(formula = cases ~ time + I(cos(2 * pi * time/12)) + I(sin(2 *
##     pi * time/12)) + I(cos(2 * pi * time/6)) + I(sin(2 * pi *
##     time/6)), id = id, data = uspolio, family = poisson(link = log),
##     corstr = "AR-M", Mv = 1)
##
## Number of observations :  168
##
## Maximum cluster size   :  12
##
##
## Coefficients:
##              (Intercept)                     time I(cos(2 * pi * time/12))
##              0.534670137             -0.004504214              0.127605025
## I(sin(2 * pi * time/12))  I(cos(2 * pi * time/6))  I(sin(2 * pi * time/6))
##             -0.518732586              0.434976179             -0.059598999
##
## Estimated Scale Parameter:  1.983319
## Number of Iterations:  3
##
## Working Correlation[1:4,1:4]
##            [,1]       [,2]       [,3]       [,4]
## [1,] 1.00000000 0.26087713 0.06805688 0.01775448
## [2,] 0.26087713 1.00000000 0.26087713 0.06805688
## [3,] 0.06805688 0.26087713 1.00000000 0.26087713
## [4,] 0.01775448 0.06805688 0.26087713 1.00000000
##
##
## Returned Error Value:
## [1] 0
```

```
round(summary(uspolio.gee4)$working.correlation, digits=3)
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
##  [1,] 1.000 0.261 0.068 0.018 0.005 0.001 0.000 0.000 0.000 0.000 0.000 0.000
##  [2,] 0.261 1.000 0.261 0.068 0.018 0.005 0.001 0.000 0.000 0.000 0.000 0.000
##  [3,] 0.068 0.261 1.000 0.261 0.068 0.018 0.005 0.001 0.000 0.000 0.000 0.000
##  [4,] 0.018 0.068 0.261 1.000 0.261 0.068 0.018 0.005 0.001 0.000 0.000 0.000
##  [5,] 0.005 0.018 0.068 0.261 1.000 0.261 0.068 0.018 0.005 0.001 0.000 0.000
##  [6,] 0.001 0.005 0.018 0.068 0.261 1.000 0.261 0.068 0.018 0.005 0.001 0.000
##  [7,] 0.000 0.001 0.005 0.018 0.068 0.261 1.000 0.261 0.068 0.018 0.005 0.001
##  [8,] 0.000 0.000 0.001 0.005 0.018 0.068 0.261 1.000 0.261 0.068 0.018 0.005
##  [9,] 0.000 0.000 0.000 0.001 0.005 0.018 0.068 0.261 1.000 0.261 0.068 0.018
## [10,] 0.000 0.000 0.000 0.000 0.001 0.005 0.018 0.068 0.261 1.000 0.261 0.068
## [11,] 0.000 0.000 0.000 0.000 0.000 0.001 0.005 0.018 0.068 0.261 1.000 0.261
## [12,] 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.005 0.018 0.068 0.261 1.000
```

This last assumption on the correlation structure of the data leads us to **repeated measures**

data, which is the actual strength of GEEs.

Chapter 6

Marginal models

6.1 Repeated measures data

6.1.1 Example: Oxford boys data

We consider a data set giving repeated measures of height (cm) of 26 boys over eight time points (i.e. total sample size $N = 234$).

```r
require(nlme)
```

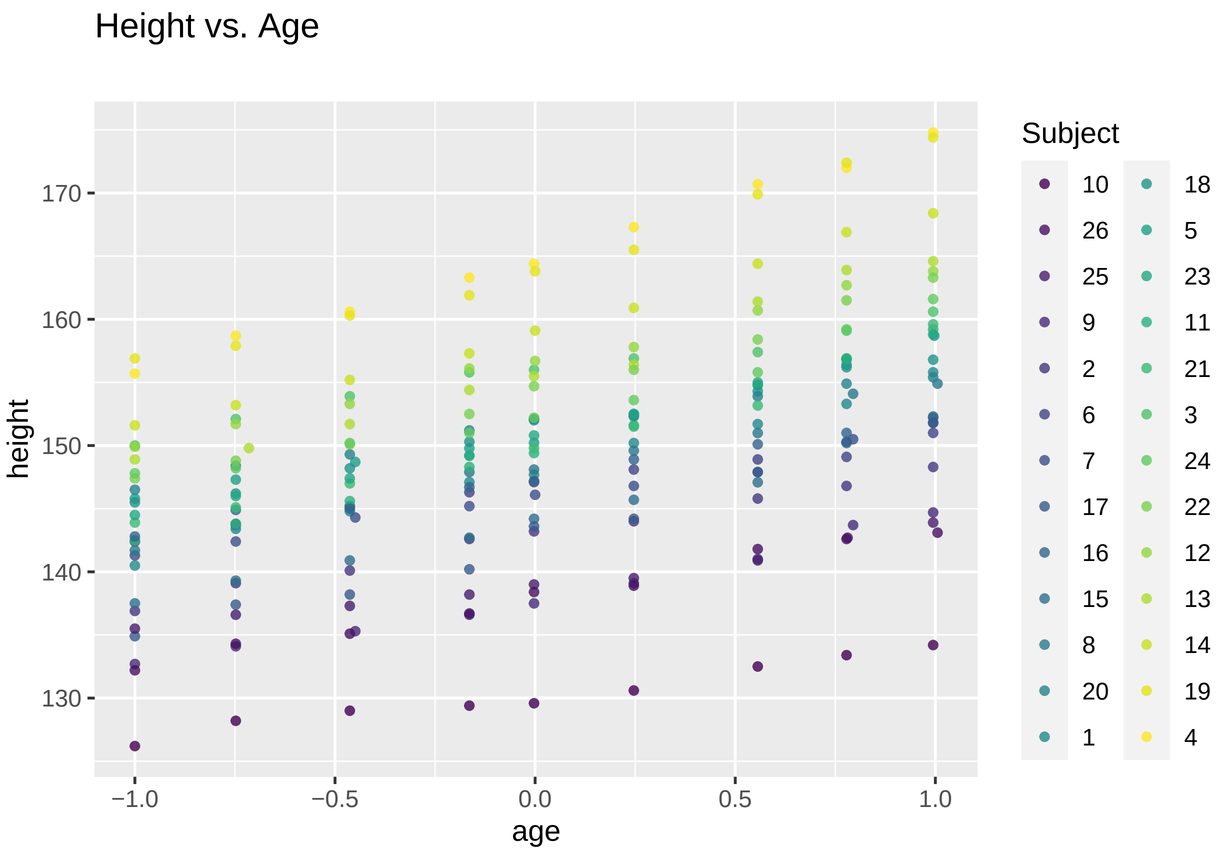
```
## Loading required package: nlme
```

```r
data(Oxboys)
# ?Oxboys
```

```r
require(ggplot2)
```

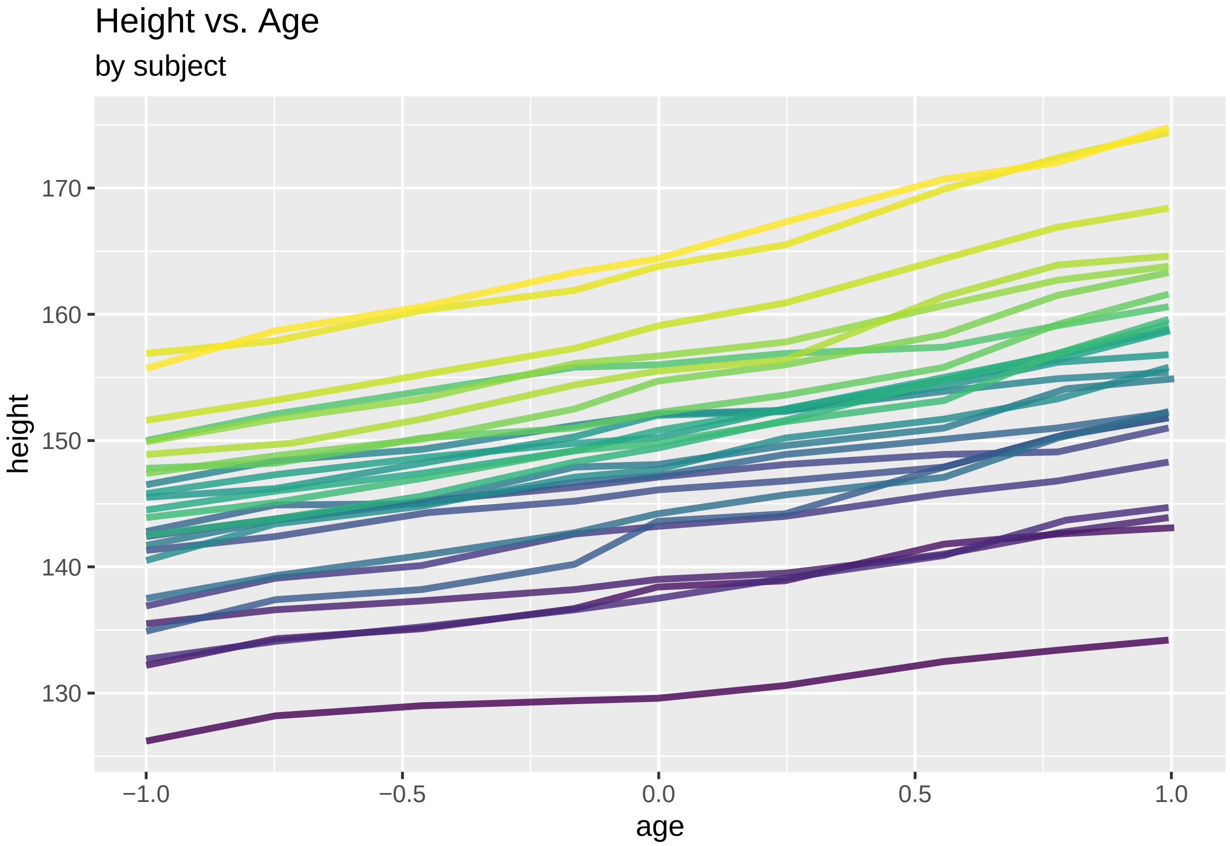
```
## Loading required package: ggplot2
```

```r
ggplot(data = Oxboys, aes(x = age, y = height, col = Subject)) +
    geom_point(size = 1.2, alpha = .8) +
    labs(title = "Height vs. Age", subtitle = "")
```



Different look at the data:

```
ggplot(data = Oxboys, aes(x = age, y = height, col = Subject)) +
       geom_line(linewidth = 1.2, alpha = .8) +
       labs(title = "Height vs. Age", subtitle = "by subject") +
       theme(legend.position = "none")
```



Clearly, there is within-subject correlation: Measures from one subject are more similar to each other than those from different subjects.

This is a special type of repeated measures data which is commonly referred to as **longitudinal data**: Repeated measures on certain individuals over time.

6.1.2 Example: Mathematics achievement data

The data used here represent Mathematics achievement scores of a subsample of subjects from the 1982 High School and Beyond Survey. The full dataset can be found within the package merTools.

For each of 30 schools, we have pupil-level data on Mathematics achievement and a few covariates including socioeconomic status.

```
load("Datasets/sub_hsb.RData") # Insert directory
head(sub_hsb)
```

```
##   schid minority female    ses mathach size schtype meanses
## 1  1224        0      1 -1.528   5.876  842       0  -0.428
## 2  1224        0      1 -0.588  19.708  842       0  -0.428
## 3  1224        0      0 -0.528  20.349  842       0  -0.428
## 4  1224        0      0 -0.668   8.781  842       0  -0.428
## 5  1224        0      0 -0.158  17.898  842       0  -0.428
## 6  1224        0      0  0.022   4.583  842       0  -0.428
```

```r
dim(sub_hsb)
```

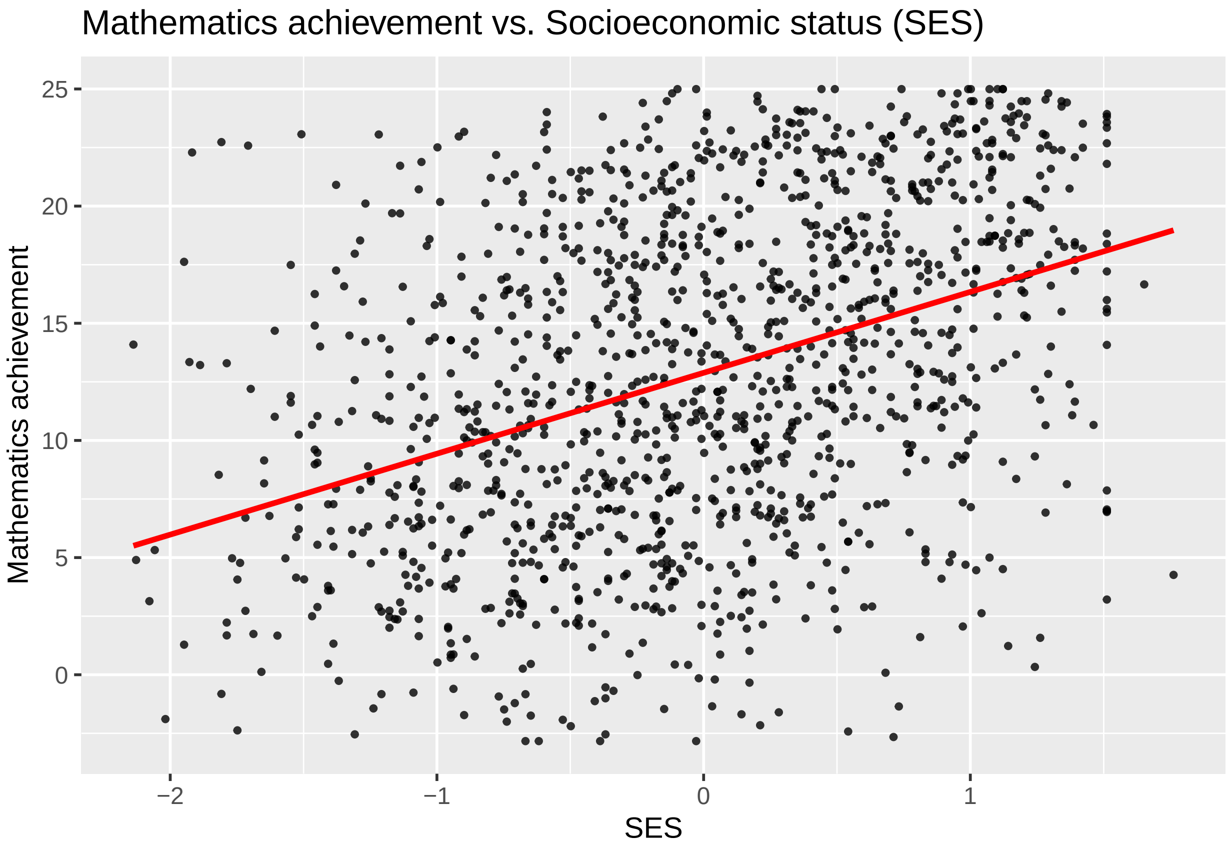
```
## [1] 1329    8
```

```r
school.id <- as.factor(sub_hsb$schid)
length(levels(school.id))
```

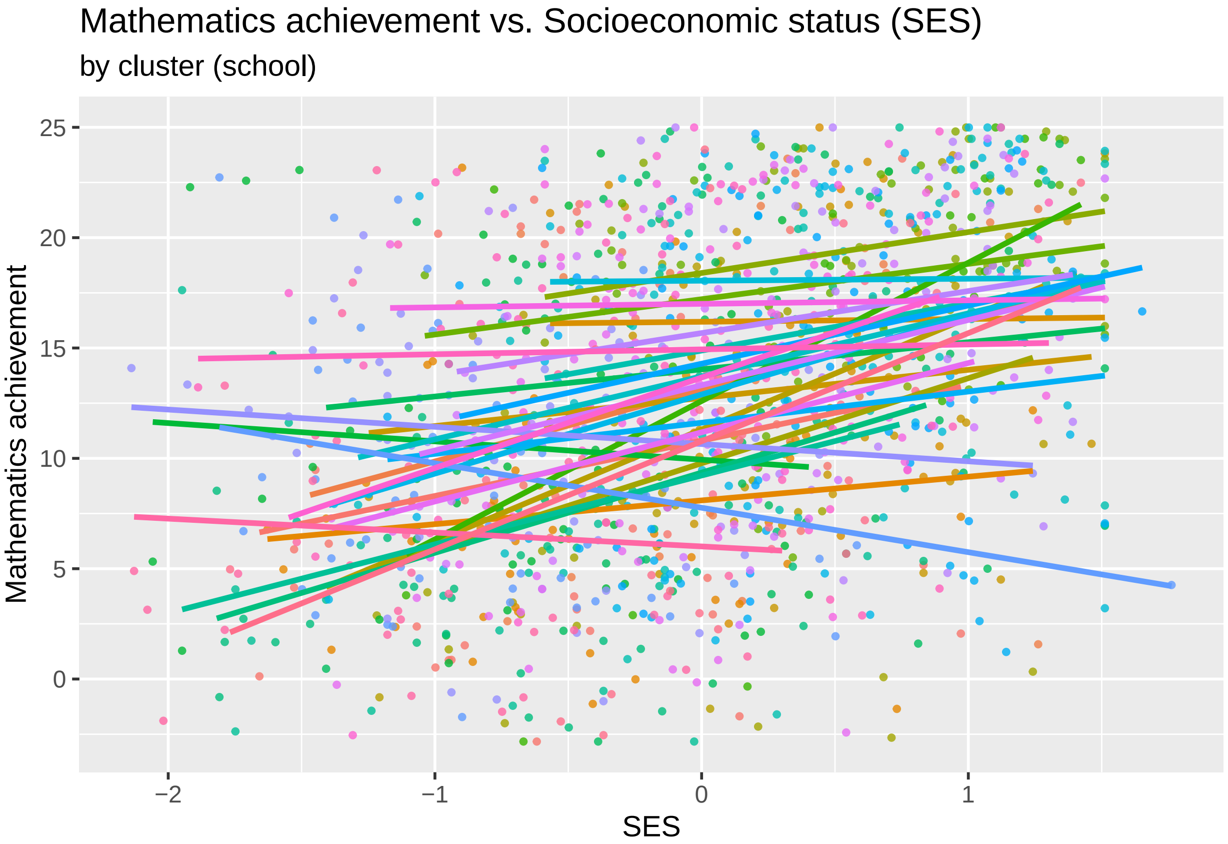
```
## [1] 30
```

```r
ggplot(data = sub_hsb, aes(x = ses, y = mathach)) +
  geom_point(size = 0.8, alpha = .8) +
  geom_smooth(method = "lm", se = FALSE, col = "Red")+
  ggtitle("Mathematics achievement vs. Socioeconomic status (SES)") +
  xlab("SES") +
  ylab("Mathematics achievement")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
ggplot(data = sub_hsb, aes(x = ses, y = mathach, colour = school.id)) +
  geom_point(size = 0.8, alpha = .8) +
  geom_smooth(method="lm", se=FALSE) +
  ggtitle("Mathematics achievement vs. Socioeconomic status (SES)",
          subtitle ="by cluster (school)") +
  xlab("SES") +
  ylab("Mathematics achievement") +
  theme(legend.position = "none")
```



One will suspect from these plots that there is within-school-correlation: Pupils from one school tend tend to be more similar to each other than to pupils from different schools.

This type of repeated measures data is commonly referred to as "clustered" data, with clusters here referring to schools. Note that the term clustered is here just to be interpreted in the sense of "items of increased similarity due to a structural reason", but it has nothing to do with "clustering" such as for instance k-means.

Fitting naïve LMs or GLMs to repeated measures data may or may not result in correct inferences for $\boldsymbol{\beta}$, but in any case the standard errors $SE(\hat{\beta}_j)$ will be incorrect. Fortunately, longitudinal and clustered data can be dealt with in the same framework.

6.2 The marginal model for repeated measures

Denote y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, n_i$, the jth repeated measurement for cluster/individual i (we will just speak of "cluster" henceforth), so that the total sample size is $N = \sum_{i=1}^{n} n_i$. Denote further the vector of all responses (repeated messurements) belonging to the ith cluster by

$$\boldsymbol{Y}_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{in_i} \end{pmatrix},$$

with associated covariates

$$\boldsymbol{X}_i = \begin{pmatrix} \boldsymbol{x}_{i1}^T \\ \vdots \\ \boldsymbol{x}_{in_i}^T \end{pmatrix} = \begin{pmatrix} x_{i11} & \dots & x_{i1p} \\ \vdots & \ddots & \vdots \\ x_{in_i1} & \dots & x_{in_ip} \end{pmatrix}.$$

A **marginal model** for y_{ij} has three components:

- the mean function ("correctly specified")

$$\mu_{ij} = E(y_{ij}) = h(\boldsymbol{\beta}^T \boldsymbol{x}_{ij});$$

- the mean-variance relationship

$$\text{Var}(y_{ij}) = \phi \mathcal{V}(\mu_{ij});$$

- the association between the responses

$$\begin{aligned} \text{corr}(y_{ij}, y_{i'k}) &= 0 \quad \text{for all} \quad i \neq i' \\ \text{corr}(y_{ij}, y_{ik}) &= r_{jk}(\boldsymbol{\alpha}), \end{aligned}$$

where $r_{jk}(\cdot)$ is a known function indexed by $j, k = 1, \dots n_i$, and $\boldsymbol{\alpha}$ is a set of parameters.

The specified variances and correlations of the y_{ij} define uniquely the variance matrix $\boldsymbol{\Sigma}_i$ of the elements of the ith cluster.

The most common settings of the function $r_{jk}(\cdot)$ are as follows:

- "Independence". Here $r_{jk}(\boldsymbol{\alpha}) \equiv r_{jk}$ does not depend on parameters, where

$$r_{jk} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

- "Exchangeable" or "Equicorrelation": For $\boldsymbol{\alpha} = \alpha \in \mathbb{R}$,

$$r_{jk}(\alpha) = \begin{cases} 1 & j = k \\ \alpha & j \neq k \end{cases}$$

- "Autoregressive model" (AR-1): For $\boldsymbol{\alpha} = \alpha \in \mathbb{R}$,

$$r_{jk}(\alpha) = \alpha^{|j-k|}$$

- "Unstructured": For $n_i \equiv n^*$ for all i, then with $\boldsymbol{\alpha} \in \mathbb{R}^{n^* \times n^*}$, and

$$r_{jk}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}_{jk}$$

Some notes:

- There is no distributional assumption and there is no likelihood (Beyond the repeated measures context, this approach is also useful for complex modelling scenarios where building a likelihood may be very hard)
- $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are overall model parameters which do not depend on the cluster, i.

- Marginal models provide "population-averaged" effects (unlike "conditional effects", which provide effects conditional on each cluster i, as we will see later for the mixed models).

What can we say about the matrices $\mathbf{\Sigma}_i$?

- Firstly, element-wise, combining the specifications of variances and correlations, we have

$$\text{Cov}(y_{ij}, y_{ik}) = \phi r_{jk}(\boldsymbol{\alpha})\sqrt{\mathcal{V}(\mu_{ij})}\sqrt{\mathcal{V}(\mu_{ik})}.$$

- Globally, defining the **working correlation matrix**

  $$R_i(\boldsymbol{\alpha}) = (r_{jk}(\boldsymbol{\alpha}))_{1\le j\le n_i, 1\le k\le n_i}$$

  and

  $$C_i(\boldsymbol{\beta}, \phi) = \text{diag}(\phi\mathcal{V}(\mu_{ij})),$$

  one has

  $$\mathbf{\Sigma}_i = C_i^{1/2}(\boldsymbol{\beta}, \phi)R_i(\boldsymbol{\alpha})C_i^{1/2}(\boldsymbol{\beta}, \phi)$$

  (clearly depends on $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, ϕ).

6.2.1 Some examples

We exemplify the model structure through some of our recent data examples.

- US Polio data (Section 5.3.1)

$$\begin{aligned} \mu_{ij} &= \exp(\beta_0 + \beta_1 t_{ij}) \\ \mathcal{V}(\mu_{ij}) &= \mu_{ij} \\ r_{jk}(\alpha) &= \alpha^{|j-k|} \\ \text{Cov}(y_{ij}, y_{ik}) &= \phi\alpha^{|j-k|}\sqrt{\mu_{ij}\mu_{ik}} \end{aligned}$$

- Oxford boys data (Section 6.1.1)

$$\begin{aligned} \mu_{ij} &= \beta_0 + \beta_1 t_{ij} \\ \mathcal{V}(\mu_{ij}) &= 1, \text{ where } \phi \equiv \sigma^2 \\ r_{jk}(\alpha) &= \alpha^{|j-k|} \\ \text{Cov}(y_{ij}, y_{ik}) &= \sigma^2\alpha^{|j-k|} \end{aligned}$$

- Mathematics achievement data (Section 6.1.2)

$$\begin{aligned} \mu_{ij} &= \beta_0 + \beta_1 \text{ses}_{ij} \\ \mathcal{V}(\mu_{ij}) &= 1, \text{ where } \phi \equiv \sigma^2 \\ r_{jk}(\alpha) &= \begin{cases} \alpha & \text{if } j \ne k \\ 1 & \text{if } j = k \end{cases} \\ \text{Cov}(y_{ij}, y_{ik}) &= \begin{cases} \alpha\sigma^2 & \text{if } j \ne k \\ \sigma^2 & \text{if } j = k \end{cases} \end{aligned}$$

6.3 Estimation

All what is needed to estimate the model elaborated above is the (generalized version of the) Quasi-Score function

$$S(\boldsymbol{\beta}) = \boldsymbol{X}^T \boldsymbol{D}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{\mu})$$

where

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1 \\ \vdots \\ \boldsymbol{Y}_n \end{pmatrix} \in \mathbb{R}^N, \;\; \boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n \end{pmatrix} \in \mathbb{R}^{N\times p}, \;\; \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \vdots \\ \boldsymbol{\mu}_n \end{pmatrix} \in \mathbb{R}^N,$$

where $\boldsymbol{Y}_i$ and $\boldsymbol{X}_i$ are as defined in Section 6.2, and $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{in_i})^T$, with $\mu_{ij} = h(\boldsymbol{\beta}^T \boldsymbol{x}_{ij})$ and $\boldsymbol{\beta} \in \mathbb{R}^p$. Furthermore,

$$\boldsymbol{D} = \begin{pmatrix} h'(\boldsymbol{\beta}^T\boldsymbol{x}_{11}) & & & \\ & h'(\boldsymbol{\beta}^T\boldsymbol{x}_{12}) & & \\ & & \ddots & \\ & & & h'(\boldsymbol{\beta}^T\boldsymbol{x}_{nn_n}) \end{pmatrix} \in \mathbb{R}^{N\times N}, \, \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & & & \\ & \boldsymbol{\Sigma}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Sigma}_n \end{pmatrix} \in \mathbb{R}^{N\times N},$$

noting that $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n_i \times n_i}$. Setting these to 0 yields the **generalized estimating equation** (GEE),

$$\boldsymbol{X}^T \boldsymbol{D}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}) = 0$$

- If $\boldsymbol{\Sigma}$ is known (and correctly specified) up to a multiplicative function of ϕ (but *not* depending on further parameters, $\boldsymbol{\alpha}$), then solve the GEE via Iteratively Weighted Least Squares (IWLS). That is, in complete analogy to the GLM estimation routines in Section 2.8, one has

  $$\hat{\boldsymbol{\beta}}_{m+1} = (\boldsymbol{X}^T\boldsymbol{D}\boldsymbol{\Sigma}^{-1}\boldsymbol{D}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{D}\boldsymbol{\Sigma}^{-1}\boldsymbol{D}\hat{\boldsymbol{Y}}_m$$

  where $\hat{\boldsymbol{Y}}_m$ is a vector of working observations (which is also the same as in Section 2.8). If $\boldsymbol{Y}$ is in fact multivariate normal and $h(\cdot)$ the identity link (so $\boldsymbol{D} = \boldsymbol{I}$), then *one* iteration $(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}$ is sufficient. In either case, estimation of $\boldsymbol{\beta}$ does not depend on ϕ, so ϕ can be estimated separately after the last iteration.

- Otherwise (if $\boldsymbol{\Sigma}$ depends on unknown parameters $\boldsymbol{\alpha}$), cycle between:

(1) Given current $\hat{\boldsymbol{\alpha}}$ and $\hat{\phi}$, estimate $\hat{\boldsymbol{\beta}}$ by (one iteration of) IWLS;
(2) Given current $\hat{\boldsymbol{\beta}}$, estimate $\hat{\boldsymbol{\alpha}}$ and $\hat{\phi}$ [explicit formulas in Fahrmeir and Tutz [2001]; page 125].

Variance estimation (under either scenario):

- "naïve":

  $$\text{Var}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X}^T\boldsymbol{D}\boldsymbol{\Sigma}^{-1}\boldsymbol{D}\boldsymbol{X})^{-1} \equiv \boldsymbol{F}^{-1}$$

- "robust": Sandwich variance estimator

  $$\text{Var}_s(\hat{\boldsymbol{\beta}}) = \boldsymbol{F}^{-1}\boldsymbol{V}\boldsymbol{F}^{-1}$$

where $\boldsymbol{V} = \boldsymbol{X}^T\boldsymbol{D}\boldsymbol{\Sigma}^{-1}\boldsymbol{S}\boldsymbol{\Sigma}^{-1}\boldsymbol{D}\boldsymbol{X}$, and $\boldsymbol{S}$ is the so-called "true" variance matrix estimated as

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_1 & & \\ & \ddots & \\ & & \boldsymbol{S}_n \end{pmatrix} \in \mathbb{R}^{N\times N}.$$

with $\boldsymbol{S}_i = (\boldsymbol{Y}_i - \hat{\boldsymbol{\mu}}_i)(\boldsymbol{Y}_i - \hat{\boldsymbol{\mu}}_i)^T$.

Theoretical properties [Fahrmeir and Tutz [2001]; page 126/127]:

Under some regularity conditions, $\hat{\boldsymbol{\beta}}$ is consistent (i.e. $\hat{\boldsymbol{\beta}} \longrightarrow \boldsymbol{\beta}$ for $N \longrightarrow \infty$) and asymptotically normal,

$$\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{F}^{-1}\boldsymbol{V}\boldsymbol{F}^{-1})$$

even if the specification of $\boldsymbol{\Sigma}$ is wrong. Correct specification of $\boldsymbol{\mu}$ is therefore more important than that of $\boldsymbol{\Sigma}$.

6.3.1 Example

GEE for Mathematics achievement data:

```r
require(gee)
hsb.gee <- gee(mathach ~ ses, data = sub_hsb,
               id=school.id, corstr = "exchangeable")
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
```

```
## (Intercept)         ses
##   12.886358    3.453019
```

```r
hsb.gee
```

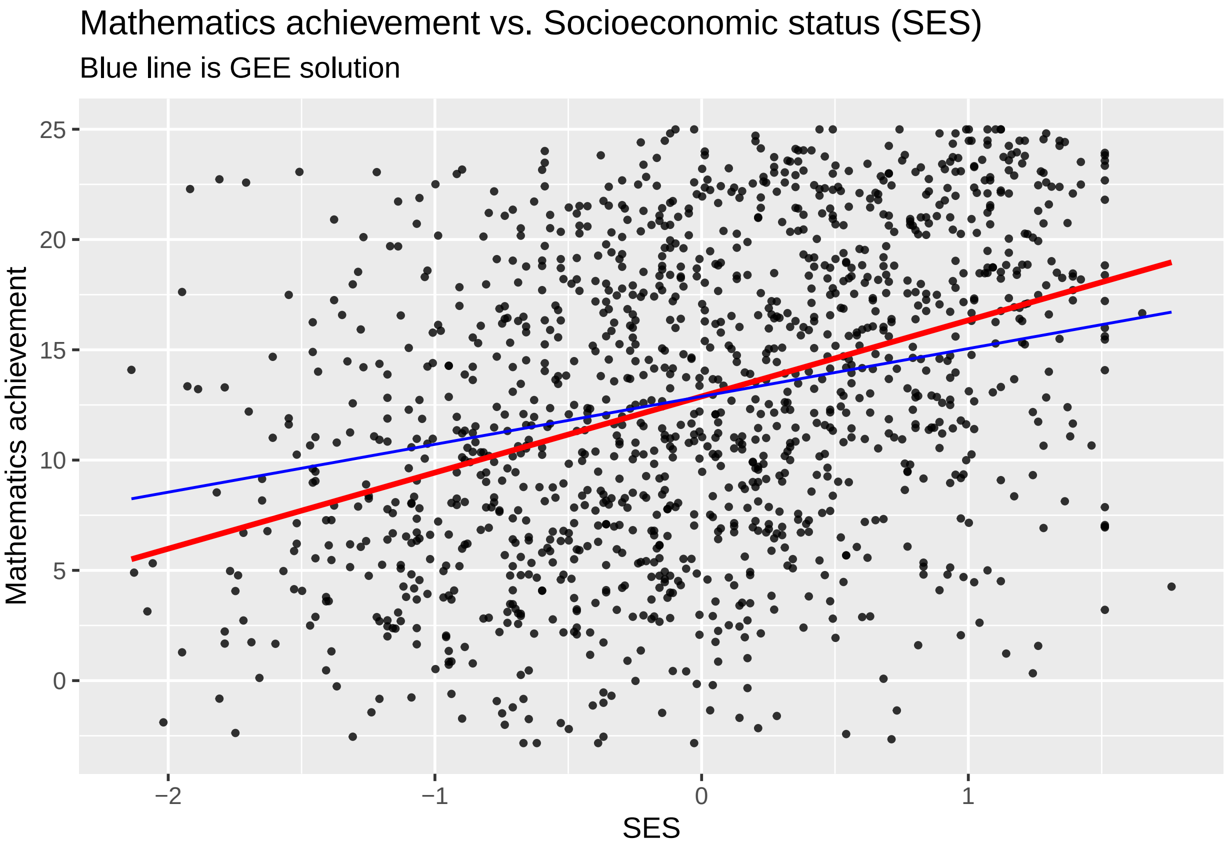
```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Identity
##  Variance to Mean Relation: Gaussian
##  Correlation Structure:     Exchangeable
##
## Call:
## gee(formula = mathach ~ ses, id = school.id, data = sub_hsb,
##     corstr = "exchangeable")
##
## Number of observations :  1329
##
## Maximum cluster size   :  67
##
```

```
##
## Coefficients:
## (Intercept)         ses
##   12.884541    2.170503
##
## Estimated Scale Parameter:  41.85127
## Number of Iterations:  4
##
## Working Correlation[1:4,1:4]
##           [,1]      [,2]      [,3]      [,4]
## [1,] 1.0000000 0.1253383 0.1253383 0.1253383
## [2,] 0.1253383 1.0000000 0.1253383 0.1253383
## [3,] 0.1253383 0.1253383 1.0000000 0.1253383
## [4,] 0.1253383 0.1253383 0.1253383 1.0000000
##
##
## Returned Error Value:
## [1] 0
```

```
# Compare to linear regression
hsb.lm<- lm(mathach ~ ses, data = sub_hsb)
```

```
sub_hsb$pred1 <- predict(hsb.gee)

ggplot(data = sub_hsb, aes(x = ses, y = mathach)) +
  geom_point(size = 0.8, alpha = .8) +
  geom_smooth(method = "lm", se = FALSE, col = "Red") +
  geom_line(aes(x = ses, y = pred1), col = "Blue") +
  ggtitle("Mathematics achievement vs. Socioeconomic status (SES)",
          subtitle = "Blue line is GEE solution") +
  xlab("SES") +
  ylab("Mathematics achievement")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



What about standard errors?

```
summary(hsb.gee)$coef
```

```
##             Estimate Naive S.E.   Naive z Robust S.E. Robust z
## (Intercept) 12.884541  0.4524909 28.474697   0.4784090 26.93206
## ses          2.170503  0.2538904  8.548976   0.3576248  6.06922
```

```
# Compare to linear regression
summary(hsb.lm)$coef
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 12.886358  0.1752970 73.51156 0.000000e+00
## ses          3.453019  0.2222153 15.53907 3.738528e-50
```

Observations:

- Both the actual estimates, and their standard errors, are quite different for the GEE and the LM.
- The robust standard errors are still a bit larger than the naive ones.

GEE for Oxford boys data:

```
data(Oxboys, package="nlme")
oxboys.gee <- gee(height~age, data=Oxboys, id=Subject, corstr="AR-M", Mv=1)
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
## (Intercept)         age
##  149.371801    6.521022
```

```
oxboys.gee
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Identity
##  Variance to Mean Relation: Gaussian
##  Correlation Structure:     AR-M , M = 1
##
## Call:
## gee(formula = height ~ age, id = Subject, data = Oxboys, corstr = "AR-M",
##     Mv = 1)
##
## Number of observations :  234
##
## Maximum cluster size   :  9
##
##
## Coefficients:
## (Intercept)         age
##  149.719096    6.547328
##
## Estimated Scale Parameter:  65.41743
## Number of Iterations:  2
##
## Working Correlation[1:4,1:4]
##           [,1]      [,2]      [,3]      [,4]
## [1,] 1.0000000 0.9892949 0.9787045 0.9682274
## [2,] 0.9892949 1.0000000 0.9892949 0.9787045
## [3,] 0.9787045 0.9892949 1.0000000 0.9892949
## [4,] 0.9682274 0.9787045 0.9892949 1.0000000
##
##
## Returned Error Value:
## [1] 0
```

```
# Compare to linear regression
oxboys.lm <- lm(height~age, data=Oxboys)
oxboys.lm
```

```
##
## Call:
## lm(formula = height ~ age, data = Oxboys)
```

```
##
## Coefficients:
## (Intercept)          age
##     149.372        6.521
```

```
summary(oxboys.gee)$coef
```

```
##               Estimate Naive S.E.  Naive z Robust S.E. Robust z
## (Intercept) 149.719096  1.5531285 96.39840   1.5847569 94.47449
## age           6.547328  0.3177873 20.60286   0.3042478 21.51972
```

```
summary(oxboys.lm)$coef
```

```
##               Estimate Std. Error    t value      Pr(>|t|)
## (Intercept) 149.371801  0.5285648 282.598864 1.987406e-296
## age           6.521022  0.8169867   7.981797  6.635134e-14
```

Standard errors under the GEE can also *decrease*!

Chapter 7

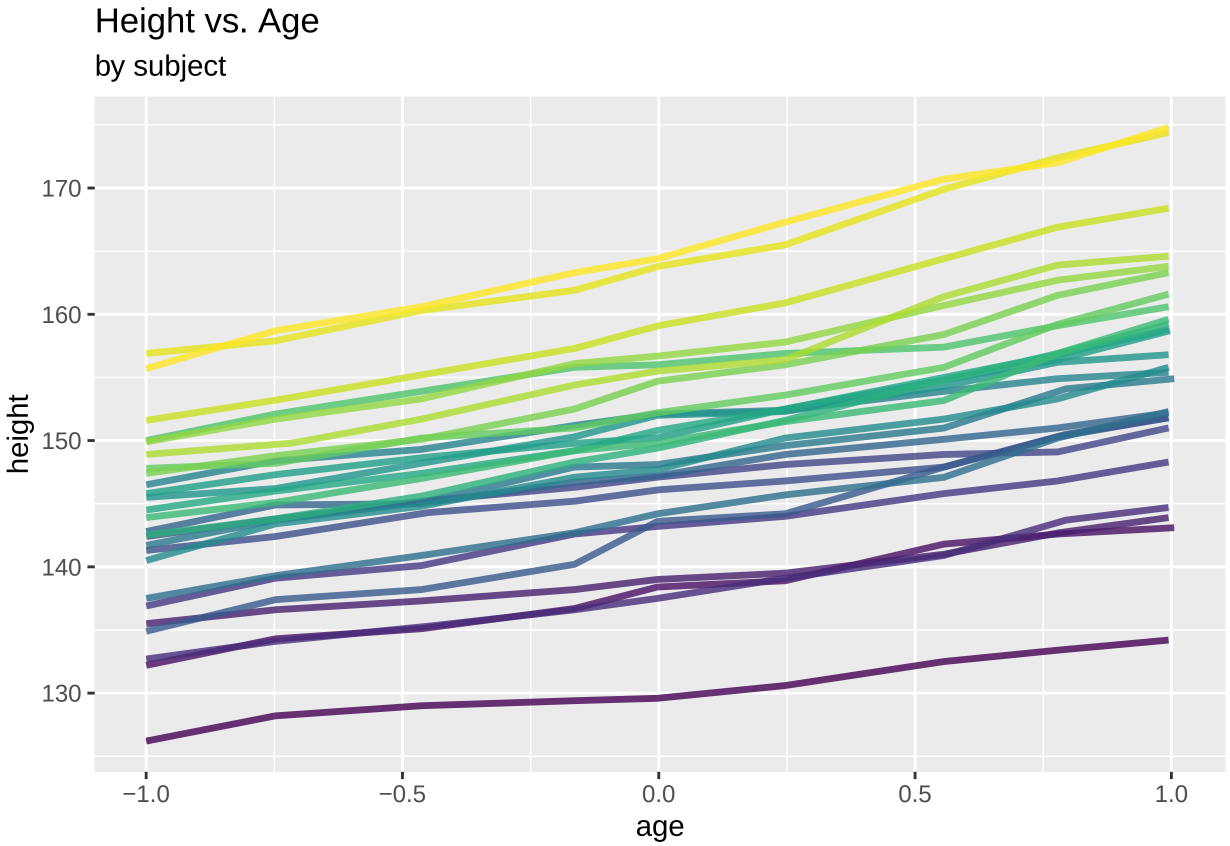
Linear mixed models

7.1 Random intercept models

7.1.1 Example: Oxford boys data

We revisit the Oxford boys data from Section 6.1.1.

```r
require(nlme)
require(ggplot2)
ggplot(data = Oxboys, aes(x = age, y = height, col = Subject)) +
       geom_line(linewidth = 1.2, alpha = .8) +
       labs(title = "Height vs. Age", subtitle="by subject") +
       theme(legend.position = "none")
```



The growth of the boys appears to be governed by an (overall) linear trend with subject-specific intercepts. In this section, we are interested in modelling these subject-specific effects explicitly, not just the marginal, population-averaged effects as in the previous section.

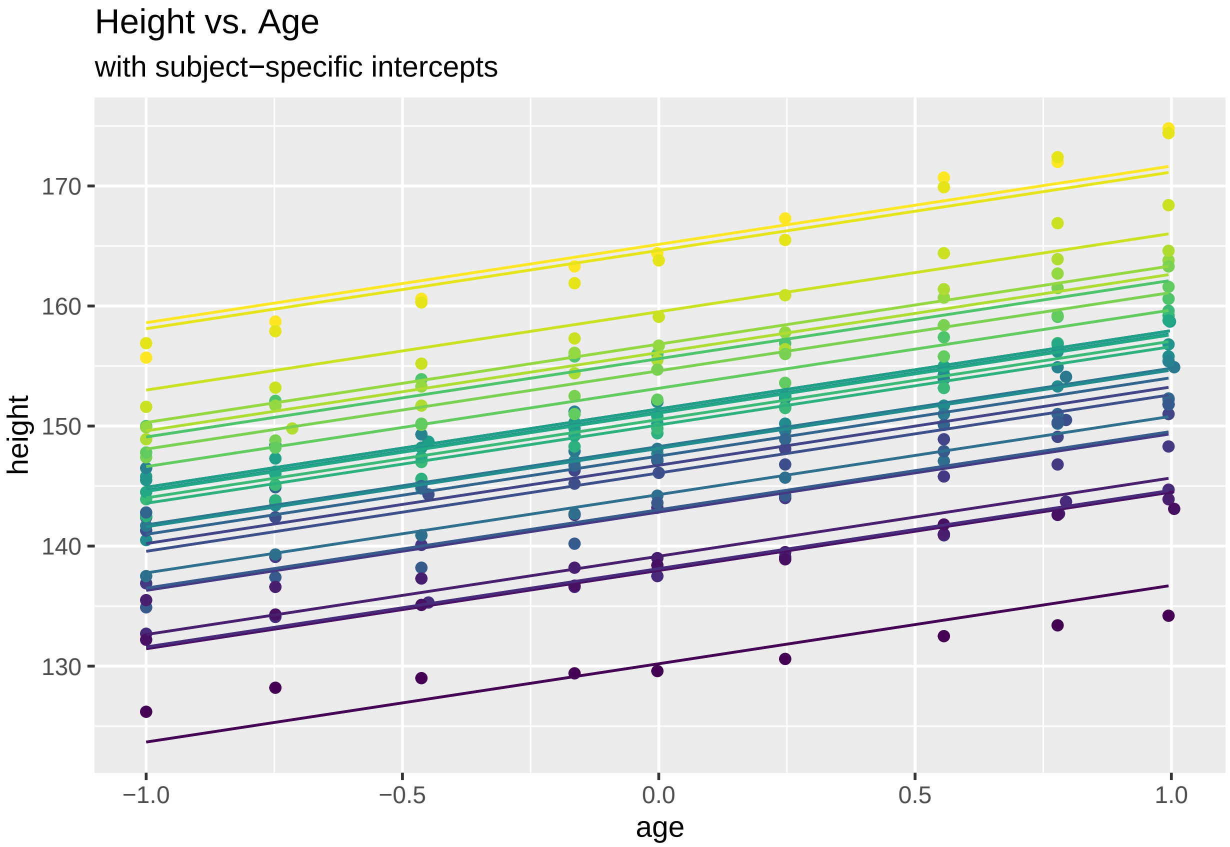
In order to do this, one could consider the following modelling strategies:

- Option 1: Fit a traditional regression model into which we include as many levels as subjects.

```
oxboys.int.lm <- lm(height~age+Subject, data=Oxboys)
oxboys.int.lm$coef
```

```
## (Intercept)         age   Subject.L   Subject.Q   Subject.C   Subject^4
## 149.3717351   6.5239272  38.2711944  -1.0176524   9.7862876  -0.2592347
##   Subject^5   Subject^6   Subject^7   Subject^8   Subject^9  Subject^10
##   2.4811449  -1.7440210  -0.1519776  -0.8033613   0.0949727  -5.1229859
##  Subject^11  Subject^12  Subject^13  Subject^14  Subject^15  Subject^16
##   1.4768831  -0.1076707  -1.4004605   1.5915615  -1.9752706   0.6604833
##  Subject^17  Subject^18  Subject^19  Subject^20  Subject^21  Subject^22
##   1.3405697   2.0599463   1.7441624  -2.3586126  -1.3409881   1.3729270
##  Subject^23  Subject^24  Subject^25
##   3.4359538   1.1353750  -1.8758449
```

```
oxboys.int.pred <- predict(oxboys.int.lm)

ggplot(data = Oxboys, aes(x = age, y = height)) +
       geom_point(aes(col=Subject)) +
       geom_line(aes(y=oxboys.int.pred, col=Subject)) +
       labs(title = "Height vs. Age", subtitle="with subject-specific intercepts") +
       theme(legend.position = "none")
```



While this seems to fit well, the approach does not appear very practicable, for two reasons: Firstly, one potentially needs very many parameters (one for each subject/cluster $i = 1, \ldots, n$). Secondly, the approach is useless for prediction of a new subject (since the intercept of that new subject will be unknown)

- Option 2: Consider the subject-specific intercepts to be drawn from a distribution centered at the overall intercept. This view implies a "hierarchical" model. One also speaks of a "two-level model" (or more generally multilevel models), where however this notion of levels has nothing to do with the notion of levels of a factor! Specifically, one has:
  - Lower level (observations/repeated measurements):

$$y_{ij} = a_i + \beta x_{ij} + \epsilon_{ij} \quad \text{with } \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

  - Upper level (clusters/subjects):

$$a_i = \alpha + u_i \quad \text{with } u_i \sim \mathcal{N}(0, \sigma_u^2)$$

    where all "random effects" u_i and model errors ϵ_{ij} are independent.

In the above, α and β indicate fixed effect parameters, while u_i and ϵ_{ij} are random quantities.

Random effect model with subjects-specific random intercepts and a single covariate $x_{ij} = \texttt{age}_{ij}$:

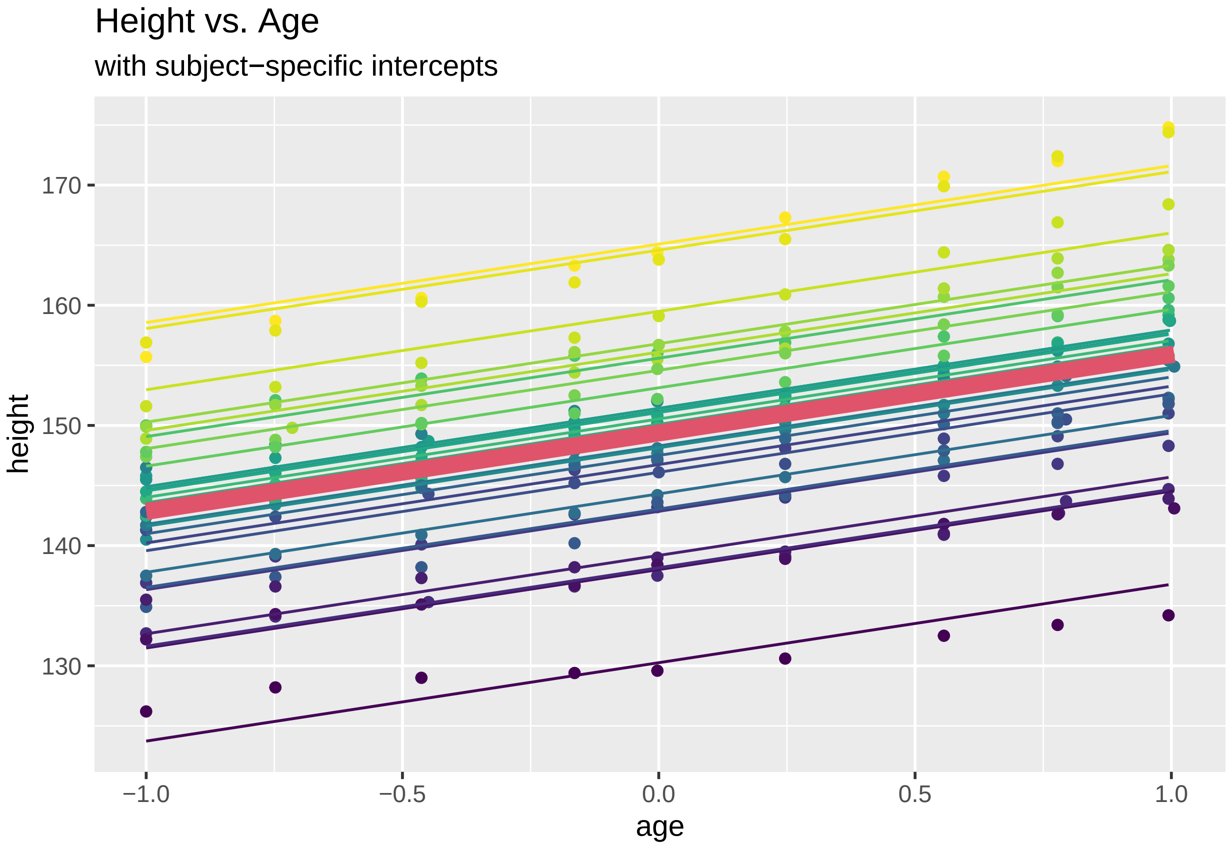
```
require(lme4)
```

```
## Loading required package: lme4

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:npmlreg':
##
##     expand

##
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':
##
##     lmList
```

```
oxboys.lmm <- lmer(height ~ age + (1 | Subject), data=Oxboys)
oxboys.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ age + (1 | Subject)
##    Data: Oxboys
## REML criterion at convergence: 940.0297
## Random effects:
##  Groups   Name        Std.Dev.
##  Subject  (Intercept) 8.097
##  Residual             1.311
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)          age
##     149.372        6.524
```

```
oxboys.lmm.pred <- predict(oxboys.lmm)  # predict xi^T beta + zi_T u_i
                                        # will study later how u_i are predicted!

oxboys.lmm.marg <- predict(oxboys.lmm, re.form=NA) # predict xi^T beta
                              # corresponds to predicting the marginal model fit

ggplot(data = Oxboys, aes(x = age, y = height)) +
      geom_point(aes(col=Subject)) +
      geom_line(aes(y=oxboys.lmm.pred, col=Subject)) +
      geom_line(aes(y=oxboys.lmm.marg), lwd=3, colour=2) +
      labs(title = "Height vs. Age", subtitle="with subject-specific intercepts")+
      theme(legend.position = "none")
```



Note that we can combine the two-level representation of the model displayed above into a single model (we also slightly generalize the notation here to allow for more than one covariates):

$$\begin{aligned} y_{ij} &= \alpha + \boldsymbol{\beta}^T \boldsymbol{x}_{ij} + u_i + \epsilon_{ij} \\ &= \alpha + u_i + \boldsymbol{\beta}^T \boldsymbol{x}_{ij} + \epsilon_{ij} \end{aligned}$$

where the first representation is useful as it highlights the separation of the model into a "fixed part" (first two terms) and a "random part" (last two terms), and the second representation is useful because it highlights its role as a **random intercept model**.

Also, it is interesting to look at the *implied* marginal effects of this model. Specifically, the marginal means are

$$E(y_{ij}) = \alpha + \boldsymbol{\beta}^T \boldsymbol{x}_{ij}$$

and the marginal variances are

$$\text{Var}(y_{ij}) = \sigma_u^2 + \sigma^2.$$

We see that

- fixed effects specify the marginal mean;
- random effects specify the marginal variance.

Since models of the type above contain a mixture of fixed effect parameters and random effects, they are also often termed "mixed effect models".

What can we say about marginal covariances?

$$\begin{aligned}
\mathrm{Cov}(y_{ij}, y_{ik}) &= E\left((y_{ij} - E(y_{ij}))(y_{ik} - E(y_{ik}))\right) \\
&= E\left((u_i + \epsilon_{ij})(u_i + \epsilon_{ik})\right) \\
&= E(u_i^2) + E(u_i)E(\epsilon_{ij}) + E(u_i)E(\epsilon_{ik}) + E(\epsilon_{ij})E(\epsilon_{ik}) \\
&= \sigma_u^2
\end{aligned}$$

This implies also

$$\mathrm{Corr}(y_{ij}, y_{ik}) = \frac{\sigma_u^2}{\sigma_u^2 + \sigma^2}$$

This quantity is commonly known as the **intra-class correlation** (ICC). It can be interpreted as the proportion of "total" variation explained by the cluster structure. Alternatively it can be interpreted as the correlation of two items randomly drawn from the same cluster.

In this context, one can show easily that

$$\mathrm{Corr}(y_{ij}, y_{i'k}) = 0 \quad \text{for } i \neq i'$$

that is, observations from different clusters are uncorrelated.

It is clear from the equations above that covariates have not played a role in this derivation. In fact, it is common to compute the ICC for a model which does not contain any fixed-effect parameters at all, i.e. $\boldsymbol{\beta} \equiv 0$. However, while the equation for ICC is then unchanged, the estimates of σ^2 and σ_u^2 may still be different for the "empty random intercept" model $y_{ij} = \alpha + u_i + \epsilon_{ij}$ and the "random intercept model with fixed effect covariates". In the literature, ICCs which are based on the empty model (without fixed effect covariates) are sometimes called unconditional ICC, and the ones including fixed effects "conditional ICC", with terminology not being entirely consistent across resources.

ICCs are often the "first shot" when assessing whether or not a repeated measure structure needs to be explicitly addressed through a (say) two-level model.

7.1.2 Example: Oxford boys data

We compute the intra-class correlation for this data set; firstly "conditional":

```
oxboys.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ age + (1 | Subject)
##    Data: Oxboys
## REML criterion at convergence: 940.0297
## Random effects:
##  Groups   Name        Std.Dev.
##  Subject  (Intercept) 8.097
##  Residual             1.311
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)          age
##     149.372        6.524
```

```
oxboys.v <- as.data.frame(summary(oxboys.lmm)$varcor)
oxboys.v
```

```
##        grp        var1 var2      vcov    sdcor
## 1  Subject (Intercept) <NA> 65.554956 8.096602
## 2 Residual        <NA> <NA>  1.718066 1.310750
```

```
icc <- oxboys.v[1,4]/(oxboys.v[1,4]+oxboys.v[2,4])
icc
```

```
## [1] 0.9744613
```

Then “unconditional”:

```
oxboys.int_only.lmm <- lmer(height ~ (1 | Subject), data=Oxboys)
oxboys.int_only.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ (1 | Subject)
##    Data: Oxboys
## REML criterion at convergence: 1466.59
## Random effects:
##  Groups   Name        Std.Dev.
##  Subject  (Intercept) 7.957
##  Residual             4.661
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)
##       149.5
```

```
oxboys.int_only.v <- as.data.frame(summary(oxboys.int_only.lmm)$varcor)
icc <- oxboys.int_only.v[1,4]/(oxboys.int_only.v[1,4]+oxboys.int_only.v[2,4])
icc
```

```
## [1] 0.7445153
```

Automated (confusing output!):

```
require(performance)
```

```
## Loading required package: performance
```

```
icc(oxboys.lmm)
```

```
## # Intraclass Correlation Coefficient
```

```
##
##     Adjusted ICC: 0.974
##   Unadjusted ICC: 0.770
```

```
icc(oxboys.int_only.lmm)
```

```
## # Intraclass Correlation Coefficient
##
##     Adjusted ICC: 0.745
##   Unadjusted ICC: 0.745
```

7.2 Random slope models

What if not only the intercept, but also the slopes subject-specific?

For ease of presentation, let us now just consider a single covariate x_{ij}. In this case, we have

- lower level:
$$y_{ij} = a_i + b_i x_{ij} + \epsilon_{ij}$$
- upper level:
$$\begin{aligned} a_i &= \alpha + u_i \\ b_i &= \beta + v_i \end{aligned}$$
where $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$, $u_i \sim \mathcal{N}(0, \sigma_u^2)$, $v_i \sim \mathcal{N}(0, \sigma_v^2)$, and ϵ_{ij} is independent with u_i and v_i. However, u_i and v_i for the same cluster may *not* be independent.

Combined this gives

$$y_{ij} = \alpha + \beta x_{ij} + u_i + v_i x_{ij} + \epsilon_{ij}$$

where $\alpha + \beta x_{ij}$ is the fixed part and $u_i + v_i x_{ij} + \epsilon_{ij}$ is the random part of the model.

Marginally this implies

$$\begin{aligned} E(y_{ij}) &= \alpha + \beta x_{ij} \\ \text{Var}(y_{ij}) &= \sigma^2 + \sigma_u^2 + \sigma_v^2 x_{ij}^2 + 2r\sigma_u\sigma_v x_{ij} \end{aligned}$$

where we define $r = \text{Corr}(u_j, v_j)$, which is sometimes assumed to be 0 [Dobson and Barnett [2018]; page 221].

7.2.1 Example: Oxford boys data

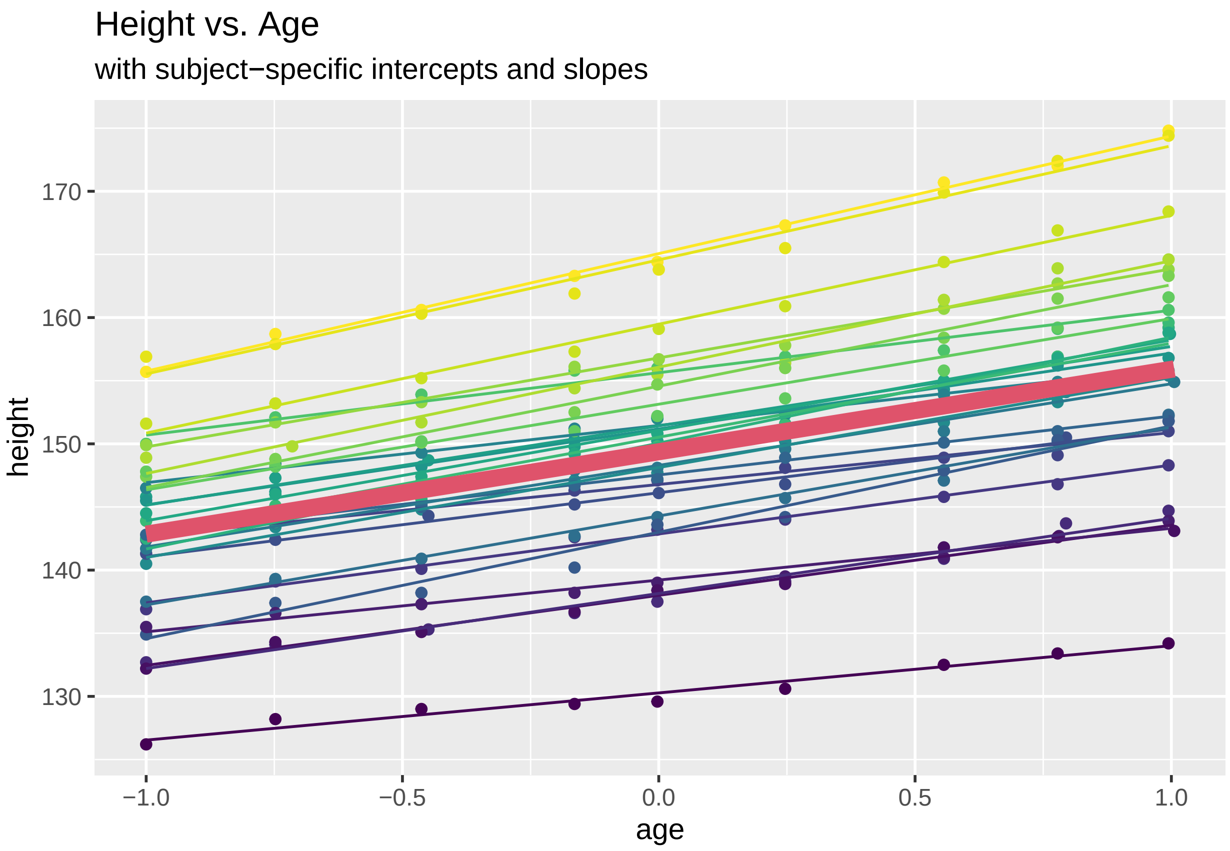
```
require(lme4)
oxboys.slope.lmm <- lmer(height ~ age + (age | Subject), data=Oxboys)
oxboys.slope.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ age + (age | Subject)
##    Data: Oxboys
## REML criterion at convergence: 724.091
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Subject  (Intercept) 8.0811
##           age         1.6807   0.64
##  Residual             0.6599
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)          age
##     149.372        6.525
```

```
oxboys.slope.lmm.pred <- predict(oxboys.slope.lmm)
oxboys.slope.lmm.marg <- predict(oxboys.slope.lmm, re.form=NA)
```

So, here $r = 0.64$.

```
ggplot(data = Oxboys, aes(x = age, y = height)) +
       geom_point(aes(col=Subject)) +
       geom_line(aes(y=oxboys.slope.lmm.pred,  col=Subject)) +
       geom_line(aes(y=oxboys.slope.lmm.marg), lwd=3, colour=2) +
       labs(title = "Height vs. Age", subtitle="with subject-specific intercepts and slo
       theme(legend.position = "none")
```



7.3 The linear mixed model (LMM)

General framework encompassing all previous models (but still only Gaussian response):

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}$$

where we have

$$\begin{aligned}\boldsymbol{Y} &= (y_{11}, \dots, y_{1n_1}, y_{21}, \dots, y_{2n_2}, \dots, y_{n1}, \dots, y_{nn_n})^T \in \mathbb{R}^N \\ \boldsymbol{\epsilon} &= (\epsilon_{11}, \dots, \epsilon_{1n_1}, \epsilon_{21}, \dots, \epsilon_{2n_2}, \dots, \epsilon_{n1}, \dots, \epsilon_{nn_n})^T \in \mathbb{R}^N\end{aligned}$$

where we we recall that $N = \sum_{i=1}^n n_i$, and

- p fixed effects; i.e. $\boldsymbol{\beta} \in \mathbb{R}^p$, with design matrix $\boldsymbol{X} \in \mathbb{R}^{N \times p}$;
- q random effects, i.e. $\boldsymbol{u} = (\tilde{\boldsymbol{u}}_1, \dots, \tilde{\boldsymbol{u}}_n)^T$ with $\tilde{\boldsymbol{u}}_1 \in \mathbb{R}^q$, and random-efffects design matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times nq}$.

7.3.1 Examples

- For the "empty model" with random intercept,

$$y_{ij} = \alpha + u_i + \epsilon_{ij},$$

one has

$$\begin{aligned}\boldsymbol{\beta} &= \alpha \in \mathbb{R}, \\ \boldsymbol{u} &= (u_1, \dots, u_n)^T \in \mathbb{R}^n, \\ \boldsymbol{X} &= (1, \dots, 1)^T \in \mathbb{R}^N,\end{aligned}$$

as well as

$$\boldsymbol{Z} = \begin{pmatrix} 1 & & \\ \vdots & & \\ 1 & & \\ & \ddots & \\ & & 1 \\ & & \vdots \\ & & 1 \end{pmatrix} \in \mathbb{R}^{N \times n}.$$

- For the random intercept model with fixed slope and one covariate,

$$y_{ij} = \alpha + \beta x_{ij} + u_i + \epsilon_{ij},$$

one has

$$\begin{aligned}
\boldsymbol{\beta} &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{R}^2, \\
\boldsymbol{u} &= (u_1, \dots, u_n)^T \in \mathbb{R}^n, \\
\boldsymbol{X} &= \begin{pmatrix} 1 & x_{11} \\ \vdots & \vdots \\ 1 & x_{nn_n} \end{pmatrix} \in \mathbb{R}^{N \times 2}
\end{aligned}$$

and $\boldsymbol{Z}$ as above.

- For the random slope model with a single covariate,

$$y_{ij} = \alpha + \beta x_{ij} + u_i + v_i x_{ij} + \epsilon_{ij},$$

one has

$$\begin{aligned}
\boldsymbol{\beta} &= \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{R}^2, \\
\boldsymbol{u} &= (u_1, v_1, \dots, u_n, v_n)^T \in \mathbb{R}^{2n},
\end{aligned}$$

$\boldsymbol{X}$ as above, and

$$\boldsymbol{Z} = \begin{pmatrix} 1 & x_{11} & & & \\ \vdots & \vdots & & & \\ 1 & x_{1n_1} & & & \\ & & \ddots & & \\ & & & 1 & x_{n1} \\ & & & \vdots & \vdots \\ & & & 1 & x_{nn_n} \end{pmatrix} \mathbb{R}^{N \times 2n}.$$

In the LMM, one (commonly) assumes:

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_N)$$

(this is because correlation structures have already been induced by the random effect, hence there will be rarely a reason to make further specifications of such correlations) and

$$\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{Q})$$

where $\boldsymbol{\epsilon}$ and $\boldsymbol{u}$ are independent, and

$$\boldsymbol{Q} = \mathrm{Var}(\boldsymbol{u}) = \begin{pmatrix} \mathrm{Var}(\tilde{\boldsymbol{u}}_1) & & \\ & \ddots & \\ & & \mathrm{Var}(\tilde{\boldsymbol{u}}_n) \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{Q}} & & \\ & \ddots & \\ & & \tilde{\boldsymbol{Q}} \end{pmatrix}$$

That is, the $\tilde{\boldsymbol{Q}}$ is the variance matrix of the random effects of the ith cluster (which usually does not depend on i).

This implies marginally

$$\begin{aligned} E(\boldsymbol{Y}) &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}E(\boldsymbol{u}) + E(\boldsymbol{\epsilon}) = \boldsymbol{X}\boldsymbol{\beta} \\ \operatorname{Var}(\boldsymbol{Y}) &= \boldsymbol{Z}\operatorname{Var}(\boldsymbol{u})\,\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N = \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N \end{aligned}$$

Summarizing, this gives us the "structured" marginal variance matrix

$$\boldsymbol{\Sigma} = \operatorname{Var}(\boldsymbol{Y}) = \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N.$$

7.4 Estimation of fixed effects

Recall our modelling framework:

$$\begin{aligned} \boldsymbol{u} &\sim \mathcal{N}(0, \boldsymbol{Q}), \\ \boldsymbol{Y}|\boldsymbol{u} &\sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \sigma^2\boldsymbol{I}_N), \\ \boldsymbol{Y} &\sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N). \end{aligned}$$

Denote the set of variance parameters ("variance components") by $\boldsymbol{\gamma} = \{\boldsymbol{Q}, \sigma^2\}$. Then $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\gamma})$; i.e. $\boldsymbol{\Sigma}$ is only fully known when $\boldsymbol{\gamma}$ is known.

For the estimation of the fixed effect parameters $\boldsymbol{\beta}$, we distinguish several cases:

1. $\boldsymbol{\gamma}$ known (hence $\boldsymbol{\Sigma}$ known). Then the solution is the same as for the marginal model with fixed $\boldsymbol{\Sigma}$, i.e.
$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}$$
Note that this is just the solution corresponding to the GEE $\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}) = 0$ with known (and correctly specified) variance matrix $\boldsymbol{\Sigma}$ (noting that in the current context we have $\boldsymbol{D} = \boldsymbol{I}$ since our setup is fully Gaussian, without link functions).
2. If $\boldsymbol{\gamma}$ is unknown, a possible approach is to estimate it through maximization of the (marginal) likelihood

$$L^*(\boldsymbol{\beta}, \boldsymbol{\gamma}) \propto \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\right)$$

In order to maximize this likelihood, one typically employs a profile-likelihood-type approach. Consider therefore the same estimator as in case 1 above, but evaluated at $\boldsymbol{\gamma}$, i.e.

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\gamma}) = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\boldsymbol{Y}$$

which can be plugged into $L^*(\boldsymbol{\beta}, \boldsymbol{\gamma})$, yielding

$$L(\boldsymbol{\gamma}) = L^*(\hat{\boldsymbol{\beta}}(\boldsymbol{\gamma}), \boldsymbol{\gamma})$$

Maximizing $L(\boldsymbol{\gamma})$ w.r.t. $\boldsymbol{\gamma}$ yields

$$\hat{\boldsymbol{\gamma}}_{ML} = \text{arg max}_{\boldsymbol{\gamma}} L(\boldsymbol{\gamma}).$$

3. REML estimation addresses the following problem in the ML solution: Just like, in the linear model (LM), the Maximum Likelihood estimate of the error variance σ^2 is biased, in the LMM the estimator $\hat{\gamma}_{ML}$ is biased for $\boldsymbol{\gamma}$, due to a "loss" of degrees of freedom in the estimation of $\boldsymbol{\beta}$. The idea of REML (Restricted Maximum Likelihood estimation) is to multiply the model equation $\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Zu} + \boldsymbol{\epsilon}$ by any matrix $\boldsymbol{A}$ which is orthogonal to $\boldsymbol{X}$, i.e. $\boldsymbol{A}^T\boldsymbol{X} = 0$, then

   $$\boldsymbol{A}^T\boldsymbol{Y} = \boldsymbol{A}^T\boldsymbol{Zu} + \boldsymbol{A}^T\boldsymbol{\epsilon}$$

   and based on this one can find a likelihood (of $\boldsymbol{A}^T\boldsymbol{Y}$) which does not depend on $\boldsymbol{\beta}$ [Fahrmeir and Tutz [2001]; page 290/291]. This "restricted likelihood" (the logarithm of which is called REML criterion in `lmer` output) does not depend on $\boldsymbol{A}$ and takes the shape

   $$L_{REML}(\boldsymbol{\gamma}) \propto |\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\gamma})\boldsymbol{X}|^{-1/2}L(\boldsymbol{\gamma})$$

   Then the REML estimator of $\boldsymbol{\gamma}$ is

   $$\hat{\gamma}_{REML} = \arg\max_{\gamma} L_{REML}(\boldsymbol{\gamma}).$$

7.4.1 Examples

REML and ML estimates for Oxford boys data

```
oxboys.slope.lmm <- lmer(height ~ age + (age | Subject), data=Oxboys)
oxboys.slope.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: height ~ age + (age | Subject)
##    Data: Oxboys
## REML criterion at convergence: 724.091
## Random effects:
##  Groups   Name        Std.Dev. Corr
##  Subject  (Intercept) 8.0811
##           age         1.6807   0.64
##  Residual             0.6599
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)          age
##     149.372        6.525
```

```
oxboys.slope.lmm.ml <- lmer(height ~ age + (age | Subject), data=Oxboys, REML=FALSE)
oxboys.slope.lmm.ml
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: height ~ age + (age | Subject)
##    Data: Oxboys
##       AIC       BIC    logLik  deviance  df.resid
##  737.9677  758.6996 -362.9838  725.9677       228
## Random effects:
```

```
##  Groups   Name        Std.Dev. Corr
##  Subject  (Intercept) 7.9240
##           age         1.6467   0.64
##  Residual             0.6599
## Number of obs: 234, groups:  Subject, 26
## Fixed Effects:
## (Intercept)          age
##     149.372        6.525
```

REML and ML estimates for mathematics achievement data

```
load("Datasets/sub_hsb.RData")
school.id <- as.factor(sub_hsb$schid)
hsb.lmm <- lmer(mathach~ses + (1|school.id), data=sub_hsb)
hsb.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ ses + (1 | school.id)
##    Data: sub_hsb
## REML criterion at convergence: 8601.028
## Random effects:
##  Groups    Name        Std.Dev.
##  school.id (Intercept) 2.518
##  Residual              6.010
## Number of obs: 1329, groups:  school.id, 30
## Fixed Effects:
## (Intercept)          ses
##       12.89         2.12
```

```
hsb.lmm.ml <- lmer(mathach~ses + (1|school.id), data=sub_hsb, REML=FALSE)
hsb.lmm.ml
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: mathach ~ ses + (1 | school.id)
##    Data: sub_hsb
##       AIC       BIC    logLik  deviance  df.resid
##  8608.516  8629.284 -4300.258  8600.516      1325
## Random effects:
##  Groups    Name        Std.Dev.
##  school.id (Intercept) 2.462
##  Residual              6.008
## Number of obs: 1329, groups:  school.id, 30
## Fixed Effects:
## (Intercept)          ses
##      12.886        2.131
```

7.5 Inference for fixed effects

Recall that, for known $\boldsymbol{\Sigma} = \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N$, one has

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\gamma}) = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}$$

which means that

$$\begin{aligned}\mathrm{Var}(\hat{\boldsymbol{\beta}}) &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\mathrm{Var}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\\ &= (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\end{aligned}$$

However, if $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\gamma})$ needs to be estimated by $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\gamma}})$, this variance estimator can be poor. Therefore, it has been suggested in the literature to also use the sandwich variance estimator (as we have seen for GEEs in Section 6.3) here. However, R function `lmer` does not actually do this. The LMM implementation in SAS does. Asymptotic normality and unbiasedness of the $\hat{\boldsymbol{\beta}}$ hold approximately. So, for some fixed effects coefficient β_j, $j = 1, \ldots, p$,

- p-values for $H_0 : \beta_j = 0$ can be approximately based on t-values $\hat{\beta}_j/SE(\hat{\beta}_j)$;
- $\hat{\beta}_j \pm z_{\alpha/2}SE(\hat{\beta}_j)$ will give reasonable confidence intervals,

where $z_{\alpha/2}$ is the right-hand tail $\alpha/2$ quantile of the standard normal distribution.

7.5.1 Example: Mathematics achievement data

We consider the random intercept model for mathematics achievement with fixed effect for socioeconomic status (SES) as fitted in Section 7.4.1:

```r
summary(hsb.lmm)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ ses + (1 | school.id)
##    Data: sub_hsb
## 
## REML criterion at convergence: 8601
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max 
## -2.81268 -0.70959 -0.03616  0.76678  2.74101 
## 
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  school.id (Intercept)  6.339   2.518   
##  Residual              36.119   6.010   
## Number of obs: 1329, groups:  school.id, 30
## 
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  12.8865     0.4908  26.256
## ses          2.1202    0.2536   8.359
##
## Correlation of Fixed Effects:
##     (Intr)
## ses -0.010
```

The t-value for the fixed effect slope `ses` is given by $2.1202/0.2536 = 8.359$, which is clearly $\gg 2$ and hence significantly different from 0 at the 5% (or any other reasonable) level of significance.

We can easily obtain an approximate 95% confidence interval for the fixed effect coefficient `ses`:

```
CI <- 2.1202 + qnorm(0.975)*c(-1,1)*0.2536
CI
```

```
## [1] 1.623153 2.617247
```

However, as stated the methods mentioned above have only approximate character. A more principled approach is to use likelihood ratio (LR)– based methods.

Therefore, assume there is a smaller model M_0 and a larger model M_1, in the sense that the smaller model is nested in the larger model M_1, but with the only difference being in the fixed effects. Let us further denote the likelihoods (of the fitted models, evaluated at the respective MLEs) by L_0 and L_1 respectively, so that clearly $L_0 < L_1$. Finally, let $D_i = -2\log L_i + c$, with c denoting a constant depending on the saturated likelihood. Then

$$D_0 - D_1 = -2\log L_0 + 2\log L_1 = -2\log\frac{L_0}{L_1} \sim \chi^2(df)$$

where df is the difference in the number of fixed effect parameters of the two models (it is not allowed here to have a difference in the number of random effect parameters). That is, for the test problem

$$H_0 : M_0 \quad \text{versus} \quad H_1 : M_1$$

one needs to fit both models and then reject H_0 if

$$D_0 - D_1 > \chi^2_\alpha(df)$$

where $\chi^2_\alpha(df)$ is the right-tail α quantile of the χ^2 distribution with df degrees of freedom.

Consider now the problem of finding a $1-\alpha$ confidence interval (or region) for some fixed effect parameters $\boldsymbol{\beta}$. (We may be interested in the whole parameter vector, or a subset of it, or just a single coefficient. We assume that $k \le p$ parameters are needed to estimate $\hat{\boldsymbol{\beta}}$.)

Then we find the confidence interval (region) by identifying the range of $\boldsymbol{\beta}$ values for which

$$\log L(\boldsymbol{\beta}) \ge \log L(\hat{\boldsymbol{\beta}}) - \frac{1}{2}\chi^2_\alpha(k)$$

where

$$L(\boldsymbol{\beta}) = L^*(\boldsymbol{\beta}, \gamma(\boldsymbol{\beta}))$$

(Such a function does not really exist, it is evaluated by software, purely computationally).

7.5.2 Example: Mathematics achievement data

We are interested in testing H_0: "no linear trend for SES" versus H_1: "There is a linear trend for SES". To carry out the test, we need to fit both models (the one under the alternative is already available, via `hsb.lmm`) and find the difference in deviances using the `anova` command. Note here that the models will be refitted with ML.

```
hsb.flat.lmm <- lmer(mathach~ 1 + (1|school.id), data=sub_hsb)
anova(hsb.flat.lmm, hsb.lmm)
```

```
## refitting model(s) with ML (instead of REML)

## Data: sub_hsb
## Models:
## hsb.flat.lmm: mathach ~ 1 + (1 | school.id)
## hsb.lmm: mathach ~ ses + (1 | school.id)
##              npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## hsb.flat.lmm    3 8670.6 8686.2 -4332.3   8664.6
## hsb.lmm         4 8608.5 8629.3 -4300.3   8600.5 64.102  1  1.182e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also obtain the confidence intervals via

```
confint(hsb.lmm)
```

```
## Computing profile confidence intervals ...

##                 2.5 %    97.5 %
## .sig01       1.830620  3.380955
## .sigma       5.783941  6.246918
## (Intercept) 11.910502 13.864699
## ses          1.613066  2.649187
```

(This also gives confidence intervals for the random effects but we have not studied these methods yet.)

Comparison of LMM to GEE

```
require(gee)
hsb.gee <- gee(mathach~ses, data=sub_hsb, id=school.id, corstr="exchangeable")
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27

## running glm to get initial regression estimate

## (Intercept)         ses
##   12.886358    3.453019
```

```
summary(hsb.gee)$coef
```

```
##              Estimate Naive S.E.   Naive z Robust S.E. Robust z
## (Intercept) 12.884541  0.4524909 28.474697   0.4784090 26.93206
## ses          2.170503  0.2538904  8.548976   0.3576248  6.06922
```

```
summary(hsb.lmm)$coef
```

```
##              Estimate Std. Error   t value
## (Intercept) 12.886541  0.4907986 26.256272
## ses          2.120208  0.2536467  8.358904
```

7.6 Prediction of random effects

Recall again

$$\begin{aligned}\boldsymbol{u} &\sim \mathcal{N}(0, \boldsymbol{Q}),\\ \boldsymbol{Y}|\boldsymbol{u} &\sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \sigma^2 \boldsymbol{I}_N),\\ \boldsymbol{Y} &\sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2 \boldsymbol{I}_N)\end{aligned}$$

where $\boldsymbol{u} \in \mathbb{R}^{nq}$ contains all random effects, so for instance in the case of the random intercept model, one has $\boldsymbol{u} = (u_1, \ldots, u_n)^T$, with $q = 1$.

What can we say about the distribution of $\boldsymbol{u}|\boldsymbol{Y}$?

In principle, this is fully available from Bayes' theorem,

$$f(\boldsymbol{u}|\boldsymbol{Y}) = \frac{f(\boldsymbol{Y}|\boldsymbol{u})f(\boldsymbol{u})}{\int(\boldsymbol{Y}|\boldsymbol{u})f(\boldsymbol{u})d\boldsymbol{u}}$$

In order to work out this posterior, we get help by a general result:

If

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}\right)$$

then

$$\boldsymbol{y}_1|\boldsymbol{y}_2 \sim \mathcal{N}\left(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{y}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right)$$

so here with

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{Y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \boldsymbol{X}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{\Sigma} \end{pmatrix}\right)$$

we obtain

$$\boldsymbol{u}|\boldsymbol{Y} \sim \mathcal{N}\left(\boldsymbol{C}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}), \boldsymbol{Q} - \boldsymbol{C}\boldsymbol{\Sigma}^{-1}\boldsymbol{C}^T\right)$$

with a covariance matrix $\boldsymbol{C}$ that we will work out later.

So in summary we can predict $\boldsymbol{u}$ by

$$E(\boldsymbol{u}|\boldsymbol{Y}) = \boldsymbol{C}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}).$$

It remains to work out $\boldsymbol{C} = \text{Cov}(\boldsymbol{u}, \boldsymbol{Y}) \in \mathbb{R}^{nq \times N}$.

Here again we make use of a general result. According to the law of total covariance, one has for any random vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$,

$$\text{Cov}(\boldsymbol{x}, \boldsymbol{y}) = E\left(\text{Cov}(\boldsymbol{x}, \boldsymbol{y})|\boldsymbol{z}\right) + \text{Cov}(E(\boldsymbol{x}|\boldsymbol{z}), E\left(\boldsymbol{y}|\boldsymbol{z}\right))$$

so when using $\boldsymbol{z} = \boldsymbol{y}$ this gives

$$\text{Cov}(\boldsymbol{x}, \boldsymbol{y}) = E\left(\text{Cov}(\boldsymbol{x}, \boldsymbol{y})|\boldsymbol{y})\right) + \text{Cov}(E(\boldsymbol{x}|\boldsymbol{y}), E\left(\boldsymbol{y}|\boldsymbol{y}\right)) = \text{Cov}(E(\boldsymbol{x}|\boldsymbol{y}), \boldsymbol{y})$$

Thus,

$$\begin{aligned}\text{Cov}(\boldsymbol{Y}, \boldsymbol{u}) &= \text{Cov}(E(\boldsymbol{Y}|\boldsymbol{u}), \boldsymbol{u}) = \text{Cov}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \boldsymbol{u}) \\ &= \text{Cov}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{u}) + \text{Cov}(\boldsymbol{Z}, \boldsymbol{u}) = \boldsymbol{Z}\text{Cov}(\boldsymbol{u}, \boldsymbol{u}) \\ &= \boldsymbol{Z}\,\text{Var}(\boldsymbol{u}) = \boldsymbol{Z}\boldsymbol{Q}\end{aligned}$$

i.e. $\boldsymbol{C}^T = \boldsymbol{Z}\boldsymbol{Q}$ and therefore $\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{Z}^T$.

Putting everything together we obtain

$$E(\boldsymbol{u}|\boldsymbol{Y}) = \boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}).$$

Now recall that $\boldsymbol{\Sigma} = \boldsymbol{Z}\boldsymbol{Q}\boldsymbol{Z}^T + \sigma^2\boldsymbol{I}_N = \Sigma(\gamma)$ with $\boldsymbol{\gamma} = \{\boldsymbol{Q}, \boldsymbol{\sigma^2}\}$. If $\boldsymbol{\gamma}$ is known and hence $\boldsymbol{Q}$ and $\boldsymbol{\Sigma}$ known, then plugging $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}$ into the expression for $E(\boldsymbol{u}|\boldsymbol{Y})$ is called the **Best Linear Unbiased Predictor (BLUP)** of $\boldsymbol{u}$,

$$\hat{\boldsymbol{u}} = \boldsymbol{Q}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}).$$

If $\boldsymbol{Q}$ and $\boldsymbol{\Sigma}$ are unknown they can be replaced by estimates, $\hat{\boldsymbol{Q}}$ and $\hat{\boldsymbol{\Sigma}}$, resulting in

$$\hat{\boldsymbol{u}} = \hat{\boldsymbol{Q}}\boldsymbol{Z}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

which is still often called BLUP (despite not being necessarily unbiased) and which we therefore do not distinguish notationally. Details can be found in McCulloch et al. [2008].

Based on the predicted random effects $\hat{\boldsymbol{u}}$, we can also straightforwardly define and compute *fitted values*

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{Z}\hat{\boldsymbol{u}}$$

and *residuals*

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} = \boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{u}}.$$

7.6.1 Example: Mathematics achievement data

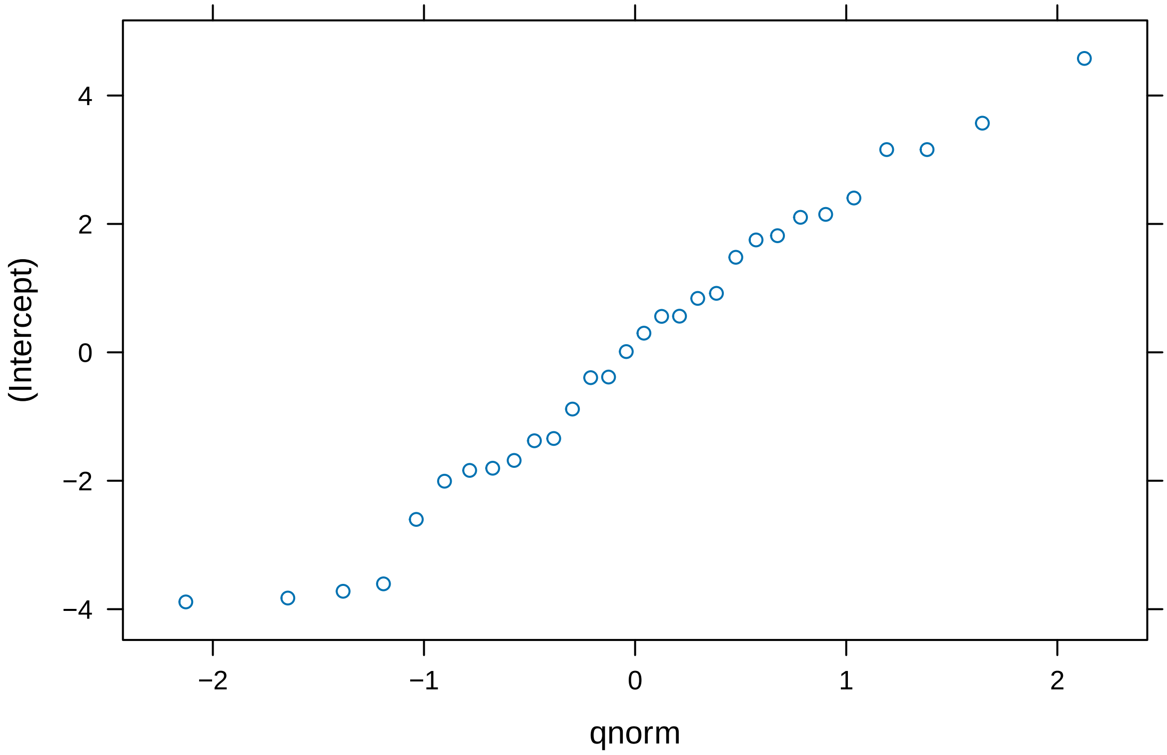
Random effects (predicted via BLUP) can be extracted from the fitted model via `ranef`:

```
hsb.ran <- ranef(hsb.lmm)
hsb.ran
```

```
## $school.id
##      (Intercept)
## 1224 -2.00682516
## 1288  0.29842627
## 1296 -3.88702782
## 1308  1.75072442
## 1317 -0.39423508
## 1358 -1.37706846
## 1374 -2.60181956
## 1433  4.57777678
## 1436  3.56937021
## 1461  2.14874621
## 1462 -0.88339941
## 1477  0.91911536
## 1499 -3.82691109
## 1637 -3.60587318
## 1906  1.81658824
## 1909  0.83990262
## 1942  3.15791804
## 1946  0.01160886
## 2030 -1.34191054
## 2208  1.48034521
## 2277 -1.80488696
## 2305 -0.38458386
## 2336  2.40354897
## 2458  0.56018478
## 2467 -1.83847667
## 2526  3.15768920
## 2626  0.56310879
## 2629  2.10286933
## 2639 -3.72142061
## 2651 -1.68348489
##
## with conditional variances for "school.id"
```

```
plot(hsb.ran)
```

```
## $school.id
```



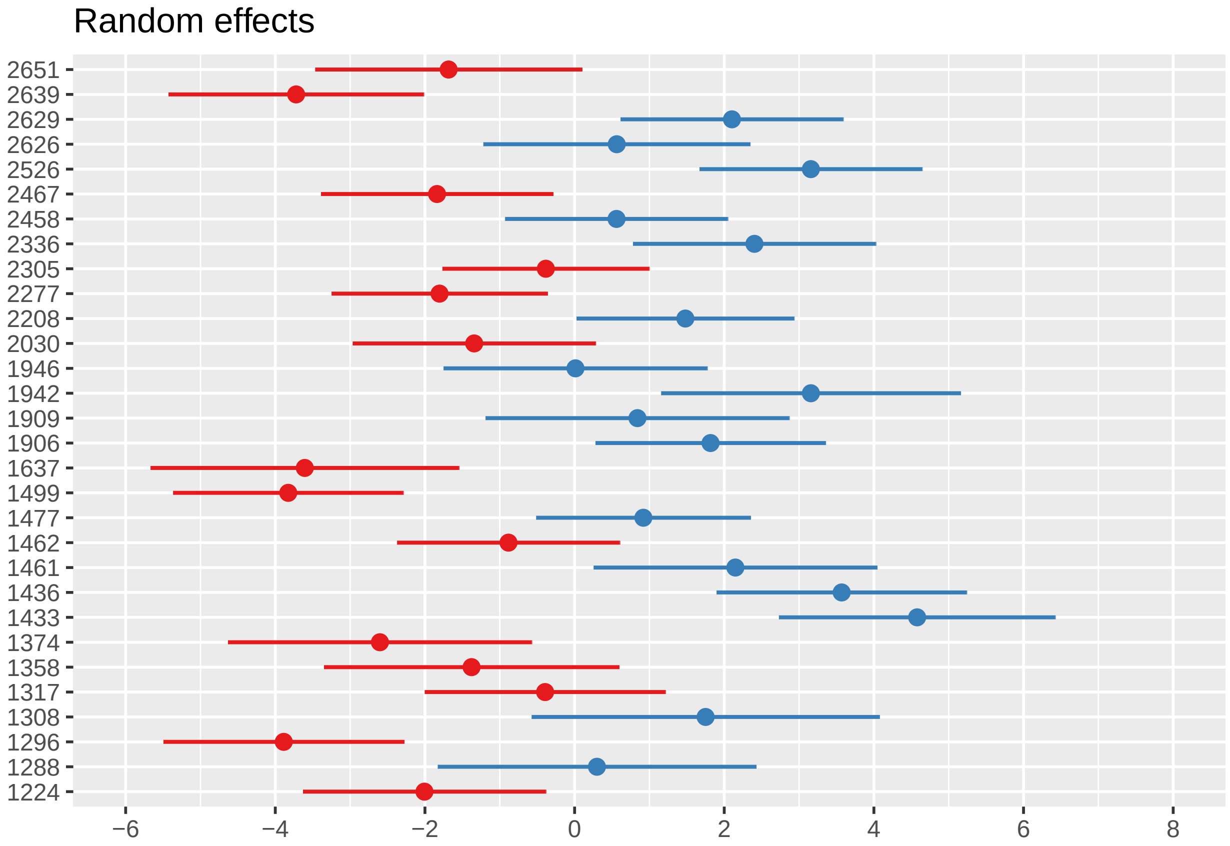
A bit nicer:

```
require(sjPlot)
```

```
## Loading required package: sjPlot
```

```
plot_model(hsb.lmm, type="re", transform=NULL)
```

```
## Warning in checkDepPackageVersion(dep_pkg = "TMB"): Package version inconsistency det
## glmmTMB was built with TMB version 1.9.6
## Current TMB version is 1.9.10
## Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinst
```



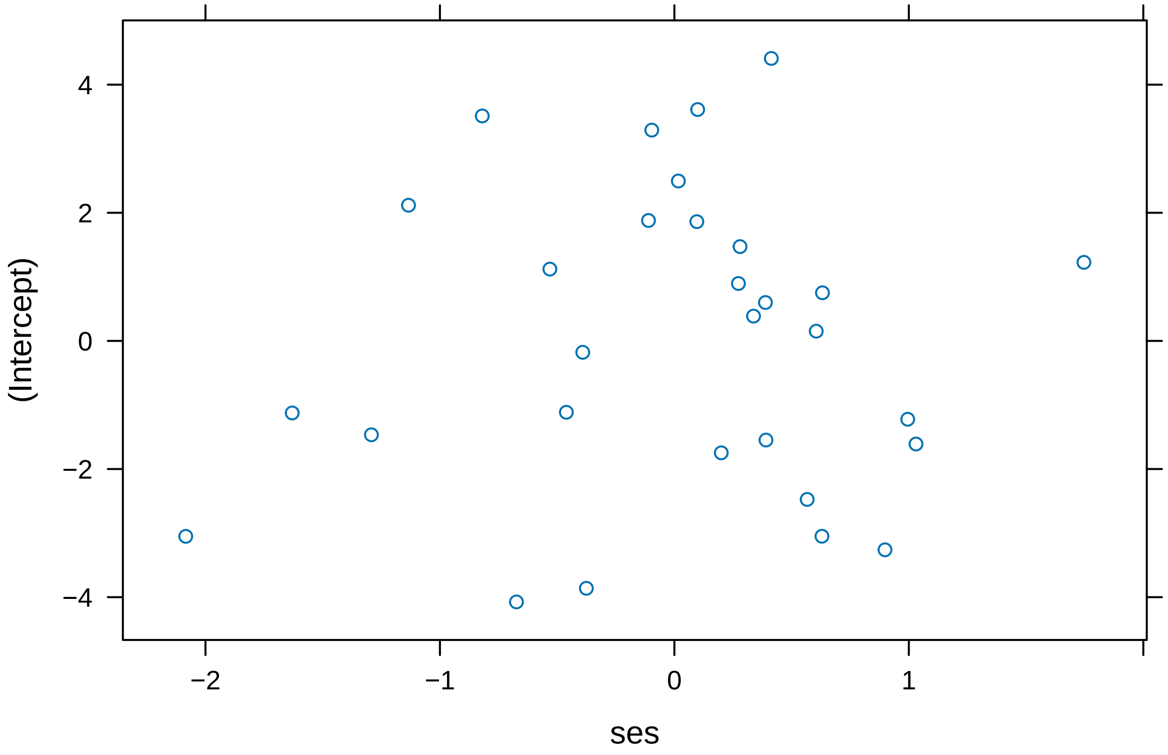
Repeat for random slope model

```
hsb.slope.lmm <- lmer(mathach~ses+ (ses|school.id), data=sub_hsb)
hsb.slope.lmm
```

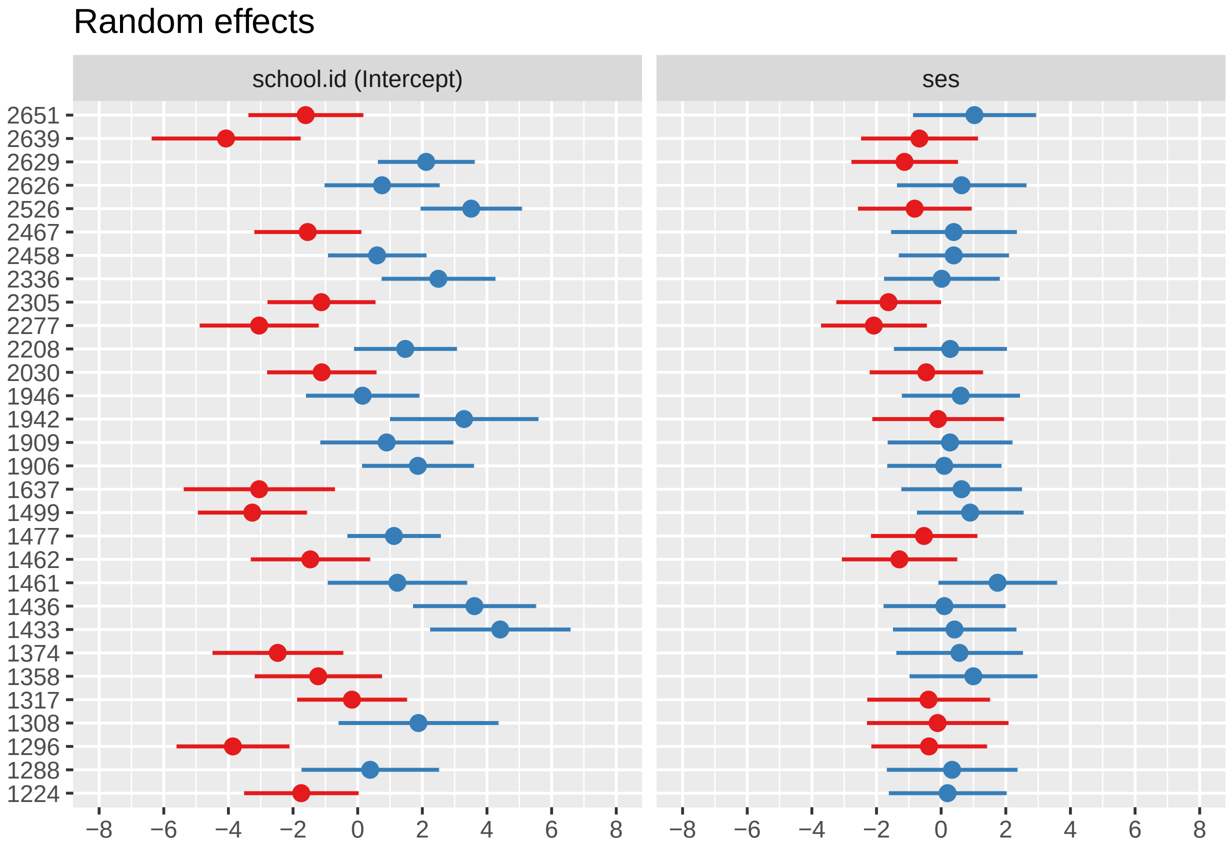
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ ses + (ses | school.id)
##    Data: sub_hsb
## REML criterion at convergence: 8593.115
## Random effects:
##  Groups    Name        Std.Dev. Corr
##  school.id (Intercept) 2.546
##            ses         1.253    0.04
##  Residual              5.951
## Number of obs: 1329, groups:  school.id, 30
## Fixed Effects:
## (Intercept)          ses
##      12.731        2.247
```

```
plot(ranef(hsb.slope.lmm))
```

```
## $school.id
```



```r
plot_model(hsb.slope.lmm, type="re", transform=NULL)
```



7.7 Inference for random effects

Recall $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}$, where

$$\boldsymbol{u} = \begin{pmatrix} \tilde{\boldsymbol{u}}_1 \\ \vdots \\ \tilde{\boldsymbol{u}}_n \end{pmatrix}$$

with $\tilde{\boldsymbol{u}}_i = (u_i, v_i, \ldots)^T$ comprising of the q random effects for cluster i, with

$$\begin{aligned} u_i &\sim \mathcal{N}(0, \sigma_u^2) \\ v_i &\sim \mathcal{N}(0, \sigma_v^2) \\ &\vdots \end{aligned}$$

Usually, the u_i, $i = 1, \ldots, n$ will correspond to a random intercept, and the v_i to a random slope for a particular coefficient. In principle, one can have one random slope for each predictor term in the model (but one can also have random slopes just for some or none of them). Let us now assume that we are interested in hypotheses of the type

$$\begin{aligned} H_0^{(u)} : \sigma_u^2 = 0 \quad &\text{versus} \quad H_1^{(u)} : \sigma_u^2 \neq 0 \\ H_0^{(v)} : \sigma_v^2 = 0 \quad &\text{versus} \quad H_1^{(v)} : \sigma_v^2 \neq 0 \\ &\vdots \end{aligned}$$

Clearly, if the null hypothesis say $H_0^{(u)}$ is not rejected, then the random effect for the u_i's is not needed, since they do not have randomness! In this case, a fixed effect, that is a traditional intercept, or in case of the $H_0^{(v)}$ a usual fixed slope, is sufficient.

To carry out these tests, we phrase the test problem again as a model comparison problem. Therefore, denote again

- M_0 as the "smaller" model excluding the random effect in question;
- M_1 as the "larger" model including that random effect.

with L_0, L_1, D_0, and D_1 the associated likelihoods and deviances.

Then consider again the LR statistic

$$D_0 - D_1 = -2 \log \frac{L_0}{L_1} \overset{H_0}{\sim} \chi^2(df)$$

Establishing the *df* does need some care. Since a q-dimensional vector of random effects will induce a $q \times q$ matrix $\tilde{\boldsymbol{Q}}$, removing one random effect will take one row and one column of $\tilde{\boldsymbol{Q}}$ out. This is best illustrated by example (see below).

It is further noted that since the REML likelihood was explicitly produced to enable accurate estimation of the variance components ($\boldsymbol{\gamma}$), here one **can** use *either* REML- or ML- based likelihoods to carry out these tests. In fact, the function `ranova` which we will use for this purpose, does use REML likelihoods.

7.7.1 Example: Mathematics achievement data

Let's begin with the random intercept model. This has only one random effect, namely the random intercept. Let's first look at this model once more.

```
hsb.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ ses + (1 | school.id)
##    Data: sub_hsb
## REML criterion at convergence: 8601.028
## Random effects:
##  Groups    Name        Std.Dev.
##  school.id (Intercept) 2.518
##  Residual              6.010
## Number of obs: 1329, groups:  school.id, 30
## Fixed Effects:
## (Intercept)          ses
##       12.89         2.12
```

Indeed there is only one random effect that can possibly be removed, which corresponds to the variance component with value $\sigma_u = 2.518$. We are now testing whether this value can be considered significantly different from 0.

```
require(lmerTest)
```

```
## Loading required package: lmerTest

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
ranova(hsb.lmm)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## mathach ~ ses + (1 | school.id)
##                  npar  logLik  AIC    LRT Df Pr(>Chisq)
## <none>              4 -4300.5 8609
## (1 | school.id)    3 -4351.0 8708 101.02  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we clearly just have $df = 1$ since `hsb.lmm` just had one random effect. Note also

$-2 \times (-4300.5) = 8601.0$, which corresponds to the value given at `REML criterion` above. We also see that $D_0 - D_1 = 101.02$ based on the difference of the values of the REML criterion and so the random intercepts are clearly needed in the model.

Now let's do the same with the random slope model.

```
hsb.slope.lmm
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ ses + (ses | school.id)
##    Data: sub_hsb
## REML criterion at convergence: 8593.115
## Random effects:
##  Groups    Name        Std.Dev. Corr
##  school.id (Intercept) 2.546
##            ses         1.253    0.04
##  Residual              5.951
## Number of obs: 1329, groups:  school.id, 30
## Fixed Effects:
## (Intercept)          ses
##      12.731        2.247
```

```
ranova(hsb.slope.lmm)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## mathach ~ ses + (ses | school.id)
##                           npar  logLik    AIC   LRT Df Pr(>Chisq)
## <none>                       6 -4296.6 8605.1
## ses in (ses | school.id)    4 -4300.5 8609.0 7.913  2    0.01913 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Removing the random slope requires removing a variance and a covariance term from $\tilde{\boldsymbol{Q}}$, hence $df = 2$. Now $D_0 - D_1 = 7.913$, which is significant at the 5% level but not at the 1% level. So, at the 1% level of significance, we would decide not to include the random slope for `ses`.

Bibliography

Annette J Dobson and Adrian G Barnett. *An introduction to generalized linear models*. CRC Press, 2018.

Ludwig Fahrmeir and Gerhard Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models (2nd edition)*. Springer, 2001.

P. J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *JRSSB*, 46(2):149–192, 1984.

W. W. Hauck and A. Donner. Wald's test as applied to hypotheses in logit analysis. *Journal of the American Statistical Association*, 72(360a):851–853, 1976.

Charles E McCulloch, Shayle R Searle, and John M Neuhaus. *Generalized, linear, and mixed models (2nd edition)*. Wiley, 2008.