apts.ac.uk

Statistical Modelling

Notes

April 19, 2024

Ioannis Kosmidis
University of Warwick
ioannis.kosmidis@warwick.ac.uk

Table of contents

Introduction

In order to get the most out of the APTS module on Statistical Modelling, students should have, at the start of the module, a sound knowledge of the principles of statistical inference and the theory of linear and generalised linear models. Students should also have some experience of statistical modelling in R.

The following reading and activities are recommended to all students to (re)-familiarise themselves with those topics.

Statistical inference: It is recommended that students (re)-read the notes of the APTS module on Statistical Inference, available from the APTS website, and complete the assessment exercise (if they have not already done so). No further material is provided here.

Linear and generalised linear models: A student who has covered Davison (2003, Chapter 8 and 10.1-10.4) will be more than adequately prepared for the APTS module. For students without access to this book, the main theory is repeated in the Preliminary Material. The inference methodology described there is largely based on classical statistical theory. Although prior experience of Bayesian statistical modelling would be helpful, it will not be assumed.

Preliminary material exercises: Nine exercises are included in the Preliminary Material.

R practicals: Some practical exercises are also provided at the end of the preliminary material (see here) to enable students to familiarise themselves with statistical modelling in R.

Typos and issues

You can report and suggest fixes to typos and issues by email to ioannis.kosmidis@warwick.ac.uk.

Acknowledgements

This set of notes is an edited and enriched version of original material developed by previous module leaders of the APTS Statistical Modelling module. These are (in reverse chronological order)

Name	Affiliation
Helen Ogden	University of Southampton
Antony Overstall	University of Southampton
Dave Woods	University of Southampton
Jon Forster	University of Warwick
Anthony Davison	EPFL

Chapter 1

Model selection

> Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.
>
> — *George Box (1919 – 2013)*
>
> in Box and Draper (1987). Empirical Model-Building and Response Surfaces, p. 74

1.1 Introduction

Statisticians construct models to simplify reality, to gain understanding, to compare scientific, economic, or other theories, and to predict future events or data. We rarely believe in our models, but regard them as temporary constructs, which should be subject to improvement. Often we have several models and must decide which, if any, is preferable.

Principles for model selection include:

- Substantive knowledge, from previous studies, theoretical arguments, dimensional or other general considerations.
- Robustness to departures from assumptions: we prefer models that provide valid inference even if some of their assumptions are invalid.
- Quality of fit: we prefer models that perform well in terms of informal devices such as residuals and graphical assessments, or more formal or goodness-of-fit tests.
- Parsimony: for reasons of economy we seek the simplest possible models that are adequate descriptions of the data.

There may be a very large number of plausible models for us to compare. For instance, in a linear regression with p covariates, there are 2^p possible combinations of covariates: for each covariate, we need to decide whether or not to include that variable in the model. If $p = 20$ we have over a million possible models to consider, and the problem becomes even more complex if we allow for transformations and interactions in the model.

To focus and simplify discussion we will consider model selection among parametric models, but the ideas generalize to semi-parametric and non-parametric settings.

Example 1.1 (Nodal involvement data)**.** A logistic regression model for binary responses assumes that $Y_i \mid x_i \sim \text{Bernoulli}(\mu_i)$, with $\mu_i = P(Y_i = 1 \mid x_i)$, and a linear model for log odds

$$\log\left(\frac{\mu_i}{1-\mu_i}\right) = x_i^\top \beta\,.$$

The log-likelihood about β, assuming that $Y_1, \ldots, Y_n$ are independent conditionally on the covariate vectors $x_1, \ldots, x_n$, is

$$\ell(\beta) = \sum_{i=1}^{n} y_i x_i^\top \beta - \sum_{i=1}^{n} \log\left\{1 + \exp(x_i^\top \beta)\right\} .$$

A good fit gives large maximized log-likelihood $\hat{\ell} = \ell(\hat{\beta})$ where $\hat{\beta}$ is the maximum likelihood estimator.

The `SMPracticals` R package contains a dataset called `nodal`, which relates to the nodal involvement (`r`) of 53 patients with prostate cancer, with five binary covariates `aged`, `stage`, `grade`, `xray` and `acid`.

Considering only the models without any interaction between the 5 binary covariates, results in $2^5 = 32$ possible logistic regression models for this data. We can rank these models according to the value of the maximized log-likelihood $\hat{\ell}$. Figure 1.1 summarizes such a ranking through a plot of the maximized log-likelihood of each of the 32 models under consideration against the number of unknown parameters in each model.

```
library("SMPracticals")
library("MuMIn")
mod_full <- glm(r ~ aged + stage + grade + xray + acid,
                data = nodal, family = "binomial", na.action = "na.fail")
mod_table <- dredge(mod_full, rank = logLik)
plot(logLik ~ df, data = mod_table,
     xlab = "Number of parameters",
     ylab = "Maximized log-likelihood",
     bg = "#ff7518", pch = 21)
```

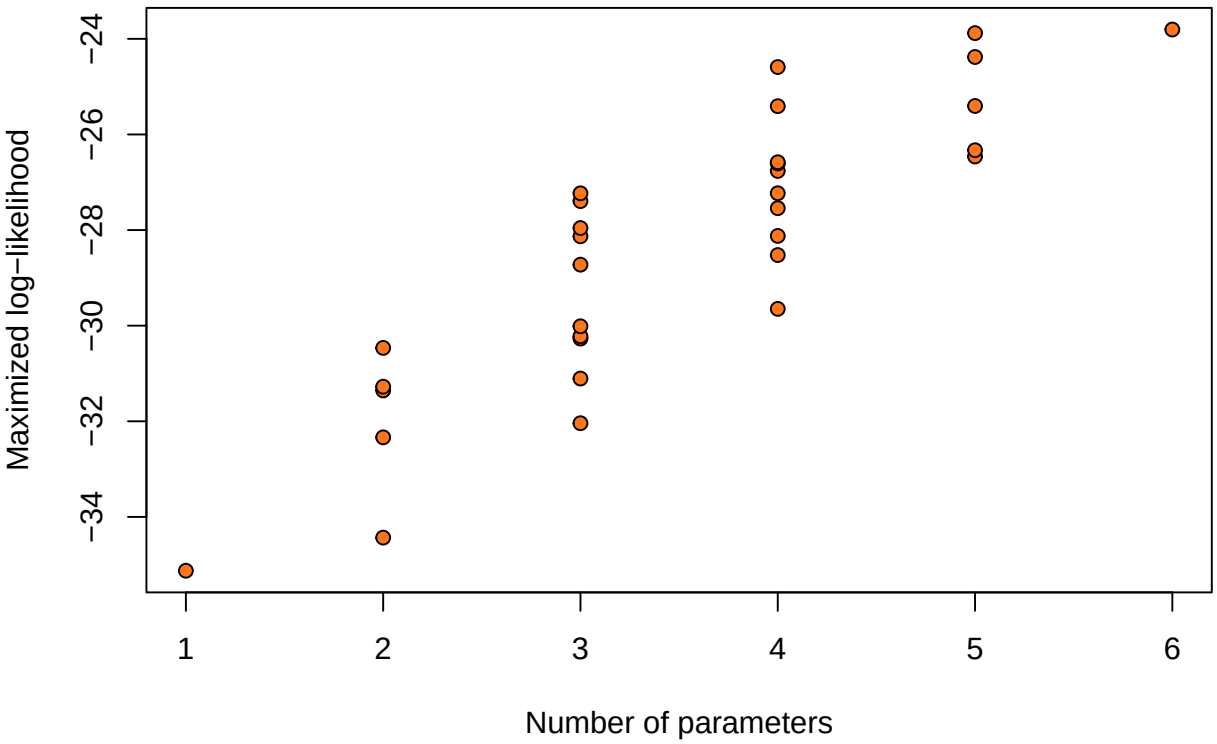


Figure 1.1: Maximized log-likelihoods for 32 possible logistic regression models for the `nodal` data.

Adding terms always increases the maximized log-likelihood $\hat{\ell}$. So, taking the model with highest $\hat{\ell}$ would give the full model. We need to a different way to compare models, which should trade off quality of fit (measured by $\hat{\ell}$) and model complexity (number of parameters or, more generally, degrees of freedom).

1.2 Criteria for model selection

1.2.1 Likelihood inference under the wrong model

Suppose the (unknown) *true model* has independent $Y_1, \ldots, Y_n$, where Y_i has a density or probability mass function $g(y)$. Suppose we have a *candidate model* that assumes $Y_1, \ldots, Y_n$ are independent where Y_i has a density of probability mass function $f(y\,;\,\theta)$. We wish to compare the candidate model against other candidate models. For each candidate model, we first find the maximum likelihood estimate $\hat{\theta}$ of the model parameters, and, then, use criteria based on the maximized log-likelihood $\hat{\ell} = \ell(\hat{\theta})$ to compare candidate models.

We do not assume that any of the candidate models are correct; there may be no value of θ such that $f(\cdot\,;\,\theta) = g(\cdot)$. Before we can decide on an appropriate criterion for choosing between models, we first need to understand the asymptotic behaviour of $\hat{\theta}$ and $\ell(\hat{\theta})$ without the usual assumption that the model is correctly specified.

The log-likelihood $\ell(\theta)$ of the candidate model is maximized at $\hat{\theta}$, and

$$\bar{\ell}(\hat{\theta}) = n^{-1}\ell(\hat{\theta}) \to \int \log f(y\,;\,\theta_g) g(y)\, dy, \quad \text{almost surely as } n \to \infty\,,$$

where θ_g minimizes the Kullback-Leibler divergence

$$KL(f_\theta, g) = \int \log\left\{\frac{g(y)}{f(y\,;\,\theta)}\right\} g(y)\, dy\,.$$

Theorem 1.1. *Suppose the true model has $Y_1, \ldots, Y_n$ independent with Y_i having a density or probability mass function $g(y)$, but, instead, we assume that Y_i has a density or probability mass function $f(y\,;\,\theta)$. Then under mild regularity conditions, the maximum likelihood estimator $\hat{\theta}$ satisfies*

$$\sqrt{n}(\hat{\theta} - \theta_g) \xrightarrow{d} \mathrm{N}\left(0, nI(\theta_g)^{-1}K(\theta_g)I(\theta_g)^{-1}\right)\,, \tag{1.1}$$

where

$$K(\theta) = n\int \frac{\partial \log f(y\,;\,\theta)}{\partial\theta}\frac{\partial \log f(y\,;\,\theta)}{\partial\theta^\top} g(y)\, dy,$$
$$I(\theta) = -n\int \frac{\partial^2 \log f(y\,;\,\theta)}{\partial\theta\partial\theta^\top} g(y)\, dy\,.$$

The likelihood ratio statistic converges in distribution as

$$W(\theta_g) = 2\left\{\ell(\hat{\theta}) - \ell(\theta_g)\right\} \xrightarrow{d} \sum_{r=1}^{p} \lambda_r V_r\,,$$

where $V_1, \ldots, V_p$ are independent with $V_i \sim \chi^2_1$, and λ_r are eigenvalues of $K(\theta_g)^{1/2}I(\theta_g)^{-1}K(\theta_g)^{1/2}$. Thus, $E\{W(\theta_g)\} \to \mathrm{tr}\{I(\theta_g)^{-1}K(\theta_g)\}$.

Under the true model, θ_g is the 'true' value of θ, $K(\theta) = I(\theta)$, $\lambda_1 = \cdots = \lambda_p = 1$, and we recover the usual results.

In practice $g(y)$ is, of course, unknown, and then $K(\theta_g)$ and $I(\theta_g)$ may be estimated by

$$\hat{K} = \sum_{i=1}^{n} \frac{\partial \log f(y_i\,;\,\hat{\theta})}{\partial\theta}\frac{\partial \log f(y_i\,;\,\hat{\theta})}{\partial\theta^\top}, \quad \hat{J} = -\sum_{i=1}^{n} \frac{\partial^2 \log f(y_i\,;\,\hat{\theta})}{\partial\theta\partial\theta^\top}\,.$$

The latter is just the observed information matrix. We can then construct confidence regions and hypothesis tests about θ_g, using the fact that, from (1.1), the approximate distribution of $\hat{\theta}$ is $\mathrm{N}\left(\theta_g, I(\theta_g)^{-1}K(\theta_g)I(\theta_g)^{-1}\right)$ and replacing the variance covariance matrix with $\hat{J}^{-1}\hat{K}\hat{J}^{-1}$.

1.2.2 Information criteria

Using the average log-likelihood $\bar{\ell}(\hat{\theta})$ to choose between models leads to overfitting, because we use the data twice: first to estimate θ, then again to evaluate the model fit.

If we had another independent sample $Y_1^+, \ldots, Y_n^+ \sim g$ and computed

$$\bar{\ell}^+(\hat{\theta}) = n^{-1} \sum_{i=1}^{n} \log f(Y_i^+ ; \hat{\theta}) ,$$

we would choose choose the candidate model that maximizes

$$\Delta = \mathrm{E}_g \left[\mathrm{E}_g^+ \left\{ \bar{\ell}^+(\hat{\theta}) \right\} \right] , \tag{1.2}$$

where the inner expectation is over the distribution of Y_i^+, and the outer expectation is over the distribution of $\hat{\theta}$.

Since $g(.)$ is unknown, we cannot compute Δ directly. We will show that $\bar{\ell}(\hat{\theta})$ is a biased estimator of Δ, but by adding an appropriate penalty term we can obtain an approximately unbiased estimator of Δ, which we can use for model comparison.

We write

$$\mathrm{E}_g\{\bar{\ell}(\hat{\theta})\} = \underbrace{\mathrm{E}_g\{\bar{\ell}(\hat{\theta}) - \bar{\ell}(\theta_g)\}}_{a} + \underbrace{\mathrm{E}_g\{\bar{\ell}(\theta_g)\} - \Delta}_{b} + \Delta .$$

Then, $a + b$ is the bias in using $\bar{\ell}(\hat{\theta})$ to estimate Δ. Hence, finding expressions for a and b would allow us to correct for that bias. We have

$$a = \mathrm{E}_g\{\bar{\ell}(\hat{\theta}) - \bar{\ell}(\theta_g)\} = \frac{1}{2n} E_g\{W(\theta_g)\} \approx \frac{1}{2n} \operatorname{tr}\{I(\theta_g)^{-1} K(\theta_g)\} .$$

Results on inference under the wrong model (we will not prove this here) may be used to show that

$$b = \mathrm{E}_g\{\bar{\ell}(\theta_g)\} - \Delta \approx \frac{1}{2n} \operatorname{tr}\{I(\theta_g)^{-1} K(\theta_g)\} .$$

Putting the latter two expressions together, we have

$$\mathrm{E}_g\{\bar{\ell}(\hat{\theta})\} = \Delta + a + b = \Delta + \frac{1}{n} \operatorname{tr}\{I(\theta_g)^{-1} K(\theta_g)\} .$$

So, in order to correct the bias in using $\bar{\ell}(\hat{\theta})$ to estimate Δ, we can aim to maximize

$$\bar{\ell}(\hat{\theta}) - \frac{1}{n} \operatorname{tr}(\hat{J}^{-1} \hat{K}) ,$$

over the candidate models. Equivalently, we can maximize

$$\hat{\ell} - \operatorname{tr}(\hat{J}^{-1} \hat{K}) ,$$

or, equivalently, minimize

$$2\{\operatorname{tr}(\hat{J}^{-1} \hat{K}) - \hat{\ell}\} ,$$

The latter expression is called the Takeuchi Information Criterion and has also been refereed to as the Network Information Criterion.

Let $p = \dim(\theta)$ be the number of parameters in a candidate model, and $\hat{\ell}$ the corresponding maximized log likelihood. There are many other information criteria with a variety of penalty terms:

Name	Acronym	Criterion
Akaike Information Criterion	AIC	$2(p - \hat{\ell})$

Name	Acronym	Criterion
Corrected AIC	AIC_c	$2(p + (p^2 + p)/(n - p - 1) - \widehat{\ell})$
Bayesian Information Criterion	BIC	$2(p \log n/2 - \widehat{\ell})$
Deviance Information Criterion	DIC	
Extended Information Criterion	EIC	
Generalized Information Criterion	GIC	
	⋮	

Another popular model selection criterion for regression problems is Mallows' $C_p = \text{RSS}/s^2 + 2p - n$, where RSS is the residual sum of squares of the candidate model, and s^2 is an estimate of the error variance σ^2.

Example 1.2 (Nodal involvement data (revisited))**.** AIC and BIC can both be used to choose between the 2^5 models that we fitted to the nodal involvement data in Example 1.1.

Both criteria prefer a model with four parameters, which includes three of the five covariates: `acid`, `stage` and `xray`.

```
mods_AIC <- dredge(mod_full, rank = AIC)
head(mods_AIC)
```

```
Global model call: glm(formula = r ~ aged + stage + grade + xray + acid, family = "binomial",
    data = nodal, na.action = "na.fail")
---
Model selection table
   (Intrc) acid aged grade stage xray df  logLik  AIC delta weight
26  -3.052    +                +    +  4 -24.590 57.2  0.00  0.319
30  -3.262    +          +     +    +  5 -23.880 57.8  0.58  0.239
28  -2.778    +    +           +    +  5 -24.380 58.8  1.58  0.145
22  -2.734    +          +          +  4 -25.409 58.8  1.64  0.141
32  -3.079    +    +     +     +    +  6 -23.805 59.6  2.43  0.095
25  -2.082                     +    +  3 -27.231 60.5  3.28  0.062
Models ranked by AIC(x)
```

```
mods_BIC <- dredge(mod_full, rank = BIC)
head(mods_BIC)
```

```
Global model call: glm(formula = r ~ aged + stage + grade + xray + acid, family = "binomial",
    data = nodal, na.action = "na.fail")
---
Model selection table
   (Intrc) acid grade stage xray df  logLik  BIC delta weight
26  -3.052    +           +    +  4 -24.590 65.1  0.00  0.341
25  -2.082                +    +  3 -27.231 66.4  1.31  0.177
22  -2.734    +     +           +  4 -25.409 66.7  1.64  0.150
18  -2.176    +                 +  3 -27.394 66.7  1.64  0.150
30  -3.262    +     +     +     +  5 -23.880 67.6  2.55  0.095
10  -2.509    +           +        3 -27.957 67.8  2.76  0.086
Models ranked by BIC(x)
```

Figure 1.2 shows the AIC and BIC for each of the 32 models, against the number of free parameters. As is apparent, BIC increases more rapidly than AIC after the minimum, as it penalizes more strongly against model complexity, as measured by the number of free parameters.

```
par(mfrow = c(1, 2))
plot(AIC ~ df, data = mods_AIC, xlab = "Number of parameters",
     bg = "#ff7518", pch = 21)
```

```
plot(BIC ~ df, data = mods_BIC, xlab = "Number of parameters",
     bg = "#ff7518", pch = 21)
```

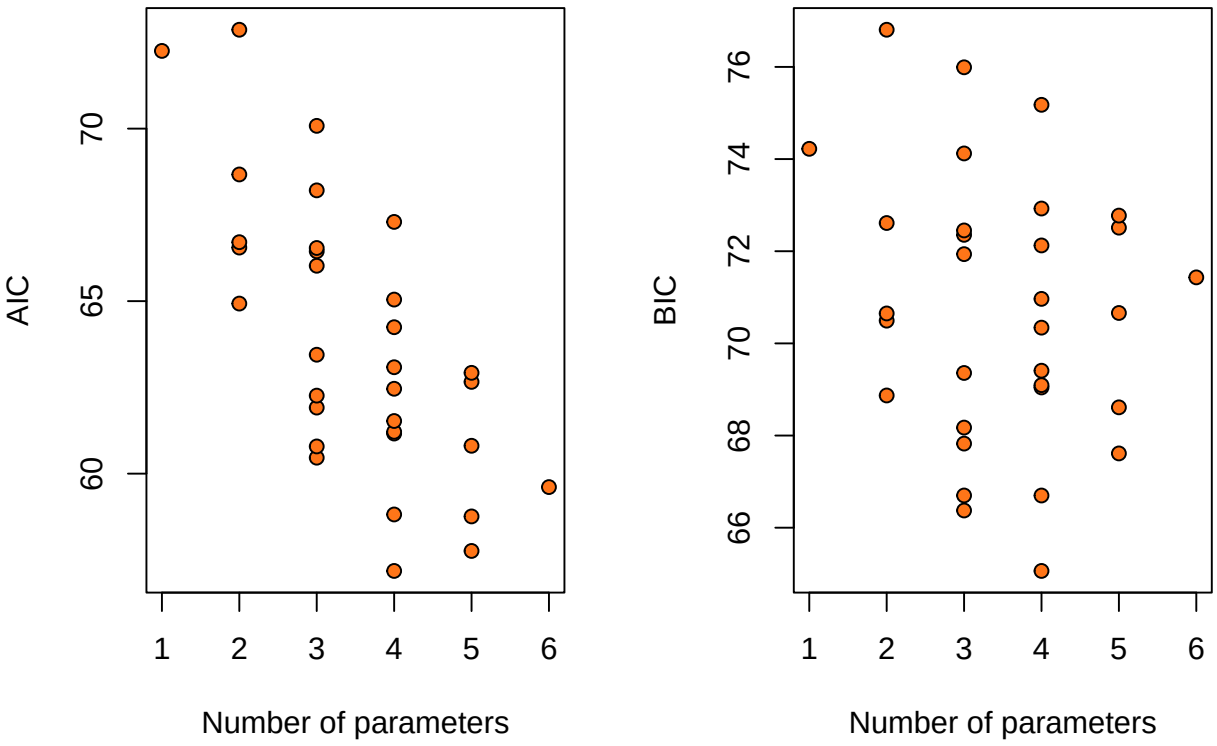


Figure 1.2: AIC and BIC for 32 logistic regression models for the `nodal` data.

1.2.3 Theoretical properties of information criteria

We may assume that the true model is of infinite dimension, and that by choosing among our candidate models we hope to get as close as possible to this ideal model, using the available data. We need some measure of distance between a candidate and the true model, and we aim to minimize that distance. A model selection procedure that selects the candidate closest to the truth for large n is called *asymptotically efficient*.

An alternative is to suppose that the true model is among the candidate models. If so, then a model selection procedure that selects the true model with probability tending to one as $n \to \infty$ is called *consistent*.

We seek to find the correct model by minimizing and information criterion $\text{IC} = c(n,p) - 2\hat{\ell}$, where the penalty $c(n,p)$ depends on sample size n and the dimension p of the parameter space.

A crucial aspect in the behaviour of model selection procedures is the differences in IC. Let IC be an information criterion for the true model, and IC_+ an information criterion for a model with one extra parameter.

Then,

$$\begin{aligned} P(\text{IC}_+ < \text{IC}) &= P\left\{c(n,p+1) - 2\hat{\ell}_+ < c(n,p) - 2\hat{\ell}\right\} \\ &= P\left\{2(\hat{\ell}_+ - \hat{\ell}) > c(n,p+1) - c(n,p)\right\}. \end{aligned}$$

Table 1.2 lists the value of $c(n,p+1) - c(n,p)$ for AIC, TIC, and BIC.

Table 1.2: Difference in IC penalties

Criterion	$c(n, p+1) - c(n, p)$
AIC	$= 2$
TIC	≈ 2 for large n
BIC	$= \log n$

Under regularity conditions about the model and as $n \to \infty$, $2(\hat{\ell}_+ - \hat{\ell})$ converges in distribution to a χ^2_1 random variable. So, as $n \to \infty$,

$$P(\mathrm{IC}_+ < \mathrm{IC}) \to \begin{cases} 0.157, & \text{if AIC or TIC is used} \\ 0, & \text{if BIC is used} \end{cases} .$$

Thus, in contrast to BIC, AIC and TIC have non-zero probability of selecting a model with an extra parameter (over-fitting), even in very large samples.

1.3 Variable selection for linear models

Consider a linear regression model

$$Y = X^*\beta + \epsilon\,,$$

with $\mathrm{E}(\epsilon) = 0$ and $\mathrm{cov}(\epsilon) = \sigma^2 I_n$, where X^* is an $n \times p^*$ model matrix with columns v_j, for $j \in \mathcal{J} = \{1, \ldots, p^*\}$, $Y = (Y_1, \ldots, Y_n)^\top$, and $\beta = (\beta_1, \ldots, \beta_{p^*})^\top$ is the vector of regression parameters.

Assume that the data generating process is a linear regression model of Y on a subset $\mathcal{J} \subseteq \mathcal{J}$ of the columns of X^* with $|\mathcal{J}| = p_0 \leq p^*$, and the goal is to estimate $\mathcal{J}$ based on data.

The parameters β enter the model through a linear predictor. So, selecting columns of X^* is formally equivalent to estimating the sets

$$\mathcal{J} = \{j \in \mathcal{J} : \beta_j \neq 0\} \quad \text{and} \quad \mathcal{K} = \{j \in \mathcal{J} : \beta_j = 0\}\,,$$

indicating which covariates should and should not be in the model, respectively.

A selected model can be either *true*, *correct*, or *wrong*. A *true* model has only those columns of $X^* with indices in $\mathcal{J}$. A *correct* model has the columns of $X^* with indices in $\mathcal{S}$, where $\mathcal{J} \subseteq \mathcal{S} \subseteq \mathcal{J}$. A *wrong* model has the columns of $X^* with indices $\mathcal{S}$, where $\mathcal{S} \not\subset \mathcal{J}$.

Suppose we fit a candidate model $Y = X\beta + \epsilon$, with X having the columns of X^* with indices in $\mathcal{S} \subseteq \mathcal{J}$ with $|\mathcal{S}| = p \leq p^*$. The fitted values are

$$X\hat{\beta} = X\{(X^\top X)^{-1} X^\top Y\} = HY = H\mu + H\epsilon\,,$$

where $\mu = X^*\beta$ is the expectation of Y, and $H = X(X^\top X)^{-1}X^\top$ is the *hat matrix*. It is a simple exercise to show that $H\mu = \mu$ if the model is correct.

As with AIC, suppose we have an independent set of responses Y_+ with $Y_+ = \mu + \epsilon_+$, where ϵ_+ and ϵ are independent, and $\mathrm{E}(\epsilon_+) = 0$ and $\mathrm{cov}(\epsilon_+) = \sigma^2 I_n$. A natural measure of prediction error in linear regression is the mean squared error

$$\Delta = n^{-1}\mathrm{E}\left[\mathrm{E}_+\left\{(Y_+ - X\hat{\beta})^\top (Y_+ - X\hat{\beta})\right\}\right]\,,$$

where expectations are taken over both Y and Y_+.

Theorem 1.2.

$$\Delta = \begin{cases} n^{-1}\mu^\top (I_n - H)\mu + (1 + p/n)\sigma^2\,, & \textit{if the model is wrong} \\ (1 + p/n)\sigma^2\,, & \textit{if the model is correct}\,. \\ (1 + p_0/n)\sigma^2\,, & \textit{if the model is true} \end{cases} \tag{1.3}$$

Proof. We have

$$Y_+ - X\hat{\beta} = Y_+ - HY = \mu + \epsilon_+ - H\mu - H\epsilon = (I_n - H)\mu + (\epsilon_+ - H\epsilon)\,.$$

Hence,

$$(Y_+ - X\hat{\beta})^\top (Y_+ - X\hat{\beta}) = \mu(I_n - H)\mu + \epsilon_+^\top \epsilon_+ + \epsilon^\top H\epsilon + Z\,,$$

where Z collects all terms with $\mathrm{E}[\mathrm{E}_+(Z)] = 0$. From the assumptions on the errors, we have $\mathrm{E}[\mathrm{E}_+(\epsilon_+^\top \epsilon_+)] = n\sigma^2$, and $\mathrm{E}[\mathrm{E}_+(\epsilon^\top H\epsilon)] = \operatorname{tr} H\sigma^2 = p\sigma^2$. Collecting terms,

$$\Delta = \underbrace{\mu(I_n - H)\mu/n}_{\text{Bias}} + \overbrace{(1 + p/n)\sigma^2}^{\text{Variance}}\,.$$

If the model is correct then $H\mu = \mu$, and the bias term is zero. If the model is also true, then $p = p_0$. □

The *bias* term $n^{-1}\mu^\top (I_n - H)\mu = n^{-1}\|\mu - H\mu\|_2^2$ is positive, unless the model is correct, in which case it is zero. Its size is reduced the closer μ is to the space spanned by the columns of X (or, equivalently, the closer μ is to its projected value $H\mu$), and, hence we would expect a reduction in bias when useful covariates are added to the model. The *variance* term $(1 + p/n)\sigma^2$ increases as p increases, for example whenever useless terms are included. Ideally, we would choose a model matrix X to minimize Δ, but this is impossible, because Δ depends on the unknowns μ and σ. We will have to estimate Δ.

Example 1.3 (Polynomial regression)**.** Consider the candidate models

$$Y_i = \sum_{j=0}^{p-1} \beta_j x_i^j + \epsilon_i \quad (i = 1, \ldots, n)\,,$$

where $x_1, \ldots, x_n$ have been generated from n independent standard normal random variables. Assume that the true model is $Y_i = \sum_{j=0}^{5} x_i^j + \epsilon_i$, hence $\mu_i = \sum_{j=0}^{5} x_i^j$ is a degree five polynomial, so $p_0 = 6$.

Let $n = 20$, and $\sigma^2 = 1$. Figure 1.3 shows $\sqrt{\Delta}$ for models of increasing polynomial degree, from the intercept only model ($p = 1$), to a linear model ($p = 2$), to a quadratic model ($p = 3$), up to a degree 14 polynomial ($p = 15$). The minimum of Δ is achieved at $p = p_0 = 6$. There is a sharp decrease in bias as useful covariates are added, and a slow increase with variance as the number of variables p increases.

```
Delta <- function(p, p0, x, sigma2) {
    cols <- 0:(p - 1)
    cols0 <- 0:(p0 - 1)
    n <- length(x)
    X <- matrix(rep(x, p)^rep(cols, each = n), nrow = n)
    X0 <- matrix(rep(x, p0)^rep(cols0, each = n), nrow = n)
    mu <- rowSums(X0)
    H <- tcrossprod(qr.Q(qr(X)))
    bias <- sum(((diag(n) - H) %*% mu)^2) / n
    variance <- sigma2 * (1 + p / n)
    c(p = p, bias = bias, variance = variance)
}

n <- 20
sigma2 <- 1
p_max <- 15
set.seed(1)
x <- rnorm(n)
D <- data.frame(t(sapply(1:p_max, Delta, p0 = 6, x = x, sigma2 = 1)))

par(mfrow = c(2, 2))
plot(sqrt(bias) ~ p, data = D, subset = p > 2,
```

```
     main = expression(sqrt("Bias")), ylab = "",
     type = "b", bg = "#ff7518", pch = 21)
plot(sqrt(bias + variance) ~ p, data = D, subset = p > 2,
     main = expression(sqrt(Delta)), ylab = "",
     type = "b", bg = "#ff7518", pch = 21)
plot(sqrt(variance) ~ p, data = D, subset = p > 2,
     main = expression(sqrt("Variance")), ylab = "",
     type = "b", bg = "#ff7518", pch = 21)
```

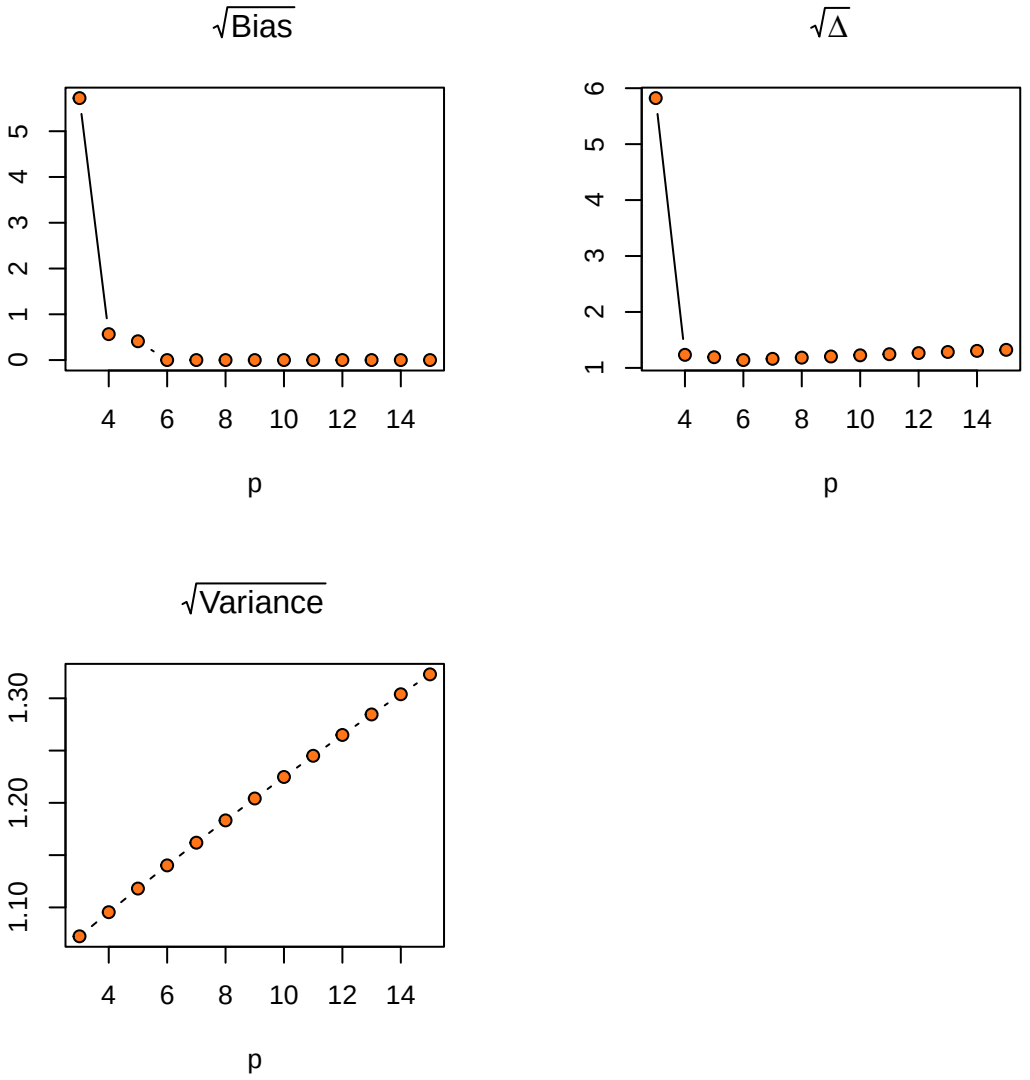


Figure 1.3: Δ for models with varying polynomial degree.

One approach to estimate Δ is to split the data into two parts, (X, y) and (X_+, y_+), where X_+ is an $n_+ \times p$ matrix and y_+ is the vector of the corresponding n_+ response values. Then, we use the former part to estimate the candidate models, and the latter part to compute the prediction error. We compute

$$\hat{\Delta} = \frac{1}{n_+}(y_+ - X_+\hat{\beta})^\top(y_+ - X_+\hat{\beta}) = \frac{1}{n_+}\sum_{i=1}^{n_+}(y_{+,i} - x_{+,i}^\top\hat{\beta})^2\,,$$

where $\hat{\beta}$ is the least squares estimator of the candidate model.

The available data may be either small for splitting, and, more generally, data splitting is not the most

efficient use of the available information. For this reason, we often use *leave-one-out cross-validation* to estimate Δ as

$$\hat{\Delta}_{\mathrm{CV}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^\top \hat{\beta}_{-i})^2\,, \tag{1.4}$$

where $\hat{\beta}_{-j}$ is the estimate computed without the ith observation. At first glance, (1.4) seems to require n fits of model. However, it can be shown that

$$\hat{\Delta}_{\mathrm{CV}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - x_i^\top \hat{\beta})^2}{(1-h_{ii})^2}\,,$$

where $h_{11}, \ldots, h_{nn}$ are diagonal elements of H. So, (1.4) can be obtained from one fit.

A simpler, and often more stable, estimator than $\hat{\Delta}_{\mathrm{CV}}$ uses *generalized cross-validation* and has the form

$$\hat{\Delta}_{\mathrm{GCV}} = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - x_i^\top \hat{\beta})^2}{\{1-\mathrm{tr}(H)/n\}^2}\,.$$

Theorem 1.3.

$$\mathrm{E}(\hat{\Delta}_{GCV}) = \frac{1}{n(1-p/n)^2}\mu^\top(I_n - H)\mu + \frac{1}{1-p/n}\sigma^2\,.$$

Furthermore, for large n, $\mathrm{E}(\hat{\Delta}_{GCV}) \approx \Delta$, with Δ as in (1.3).

Proof. It holds that $Y - X\hat{\beta} = (I_n - H)Y = (I_n - H)\mu + (I_n - H)\epsilon$. So,

$$(Y - X\hat{\beta})^\top(Y - X\hat{\beta}) = \mu^\top(I_n - H)\mu + \epsilon^\top(I_n - H)\epsilon + Z\,,$$

where the terms collected in Z have expectation zero. Using the fact that $\mathrm{tr}\, H = p$,

$$\begin{aligned}\mathrm{E}(\hat{\Delta}_{\mathrm{GCV}}) &= \frac{1}{n(1-p/n)^2}\mu^\top(I_n - H)\mu + \frac{(n-p)}{n(1-p/n)^2}\sigma^2 \\ &= \frac{1}{n(1-p/n)^2}\mu^\top(I_n - H)\mu + \frac{1}{1-p/n}\sigma^2\,.\end{aligned} \tag{1.5}$$

For large n or small $z = 1/n$, a Taylor expansion of $(1-pz)^{-1}$ about $z = 0$ gives $(1-pz)^{-1} = 1+pz+O(z^2)$. Also, $z(1-zp)^{-2} = z + O(z^2)$. Replacing $(1-p/n)^{-1}$ and $n^{-1}(1-p/n)^{-2}$ in the last expression in (1.5) with their first order approximations gives the right hand side of (1.3). □

We can minimize either $\hat{\Delta}_{\mathrm{CV}}$ or $\hat{\Delta}_{\mathrm{GCV}}$. Model selection based on leave-one-out cross validation has been found to be less stable than generalized cross-validation. Another estimator of Δ is obtained using k-fold cross-validation. k-fold cross-validation operates by splitting the data into k roughly equal parts (say $k = 10$), predicting the response for each part based on the model fit from the other $k-1$ parts, and, then, selecting the model that minimizes an aggregate estimate of prediction error.

1.4 A Bayesian perspective on model selection

In a parametric model, the data y is assumed to be a realization of Y with density or probability mass function $f(y \mid \theta)$, where $\theta \in \Omega_\theta$.

Separate from data, the prior information about the parameter θ is summarized in a prior density or probability mass function $\pi(\theta)$. The posterior density for θ is given by Bayes' theorem as

$$\pi(\theta \mid y) = \frac{\pi(\theta)f(y \mid \theta)}{\int \pi(\theta)f(y \mid \theta)\,d\theta}\,.$$

Here $\pi(\theta \mid y)$ contains all information about θ, conditional on the observed data y. If $\theta = (\psi^\top, \lambda^\top)^\top$, then inference for ψ is based on the *marginal posterior density or probability mass function*

$$\pi(\psi \mid y) = \int \pi(\theta \mid y)\, d\lambda\,.$$

Now, suppose we have M alternative models for the data, with respective parameters $\theta_1 \in \Omega_{\theta_1}, \ldots, \theta_M \in \Omega_{\theta_M}$. The spaces Ω_{θ_m} may have different dimensions.

We enlarge the parameter space to define an *encompassing model* with parameter

$$\theta \in \Omega = \bigcup_{m=1}^{M} \{m\} \times \Omega_{\theta_m}\,.$$

We need priors $\pi_m(\theta_m \mid m)$ for the parameters of each model, plus a prior $\pi(m)$ giving pre-data probabilities for each of the models. Then, for each model we have

$$\pi(m, \theta_m) = \pi(\theta_m \mid m)\pi(m)\,.$$

Inference about model choice is based on the marginal posterior

$$\begin{aligned}\pi(m \mid y) &= \frac{\int f(y \mid \theta_m)\pi_m(\theta_m)\pi(m)\, d\theta_m}{\sum_{m'=1}^{M} \int f(y \mid \theta_{m'})\pi_{m'}(\theta_{m'})\pi(m')\, d\theta_{m'}} \\ &= \frac{\pi(m) f(y \mid m)}{\sum_{m'=1}^{M} \pi(m') f(y \mid m')}\,.\end{aligned} \tag{1.6}$$

For each model, we can write the joint posterior of model and parameters as

$$\pi(m, \theta_m \mid y) = \pi(\theta_m \mid m, y)\pi(m \mid y)\,,$$

so Bayesian updating corresponds to the map

$$\pi(\theta_m \mid m)\pi(m) \mapsto \pi(\theta_m \mid m, y)\pi(m \mid y)\,.$$

So, for each model $m \in \{1, \ldots, M\}$, it is necessary to compute

- the posterior probability $\pi(m \mid y)$, which involves the marginal likelihood $f(y \mid m) = \int f(y \mid \theta_m, m)\pi(\theta_m \mid m)\, d\theta_m$; and
- the posterior density $\pi(\theta_m \mid y, m)$.

If there are just two models, we can write

$$\frac{\pi(1 \mid y)}{\pi(2 \mid y)} = \frac{\pi(1)}{\pi(2)} \frac{f(y \mid 1)}{f(y \mid 2)},$$

so the posterior odds on model 1 equal the prior odds on model 1 multiplied by the *Bayes factor* $B_{12} = f(y \mid 1)/f(y \mid 2)$.

Example 1.4 (Lindley's paradox)**.** Suppose the prior for each θ_m is $N(0, \sigma^2 I_{p_m})$, where $p_m = \dim(\theta_m)$. Then,

$$\begin{aligned} f(y \mid m) &= \sigma^{-p_m}(2\pi)^{-p_m/2} \int f(y \mid m, \theta_m) \prod_{r=1}^{p_m} \exp\left\{-\theta_{m,r}^2/(2\sigma^2)\right\} d\theta_{m,r} \\ &\approx \sigma^{-p_m}(2\pi)^{-p_m/2} \int f(y \mid m, \theta_m) \prod_{r=1}^{p_m} d\theta_{m,r}, \end{aligned}$$

for a highly diffuse prior distribution (large σ^2). The Bayes factor for comparing the models is then approximately

$$\frac{f(y \mid 1)}{f(y \mid 2)} \approx \sigma^{p_2 - p_1} g(y),$$

where $g(y)$ depends on the two likelihoods but is independent of σ^2. Hence, *whatever the data tell us about the relative merits of the two models*, the Bayes factor in favour of the simpler model can be made arbitrarily large by increasing σ. This illustrates **Lindley's paradox**, and highlights that we must be careful when specifying prior dispersion parameters when comparing models.

If a quantity Z has the same interpretation for all models, it is desirable to allow for model uncertainty when constructing inferences or making predictions about it.

If prediction is the aim, the each model may be just a vehicle that provides a future value, and not of interest on its own.

If Z corresponds to physical parameters (e.g. means, variances, etc.) *that have the same interpretation across models*, then inferences can be constructed accounting for model uncertainty, but care is needed with prior choice.

The predictive distribution for Z may be written

$$f(z \mid y) = \sum_{m=1}^{M} f(z \mid m, y)\pi(m \mid y)\,,$$

where $\pi(m \mid y)$ is as in (1.6).

Chapter 2

Beyond generalized linear models

> We must be careful not to confuse data with the abstractions we use to analyse them.
>
> — *William James (1842 – 1910)*

2.1 Generalized linear models

Suppose that $y_1, \ldots, y_n$ are observations of response variables $Y_1, \ldots, Y_n$, which are assumed to be independent conditionally on covariates $x_1, \ldots, x_n$. Furthermore, assume that Y_i has an exponential family distribution. A generalized linear models links the mean $\mu_i = \mathrm{E}(Y_i \mid x_i)$ to a linear combination of covariates and regression parameters through a link function $g(\cdot)$ as

$$g(\mu_i) = \eta_i = x_i^\top \beta \,.$$

Generalized linear models (GLMs) have proved effective at modelling real-world variation in a wide range of application areas. However, situations frequently arise where GLMs do not adequately describe observed data. This can be due to a number of reasons including:

- The mean model cannot be appropriately specified due to dependence on an unobserved or unobservable covariate.
- There is excess variability between experimental units beyond what is implied by the mean/variance relationship of the chosen response distribution.
- The assumption of independence is not appropriate.
- Complex multivariate structures in the data require a more flexible model.

2.2 Overdispersion

2.2.1 An example

Example 2.1 (Toxoplasmosis data)**.** The `toxo` dataset in the `SMPracticals` R package provides data on the number of people testing positive for toxoplasmosis (`r`) out of the number of people tested (`m`) in 34 cities in El Salvador, along with the annual rainfall in mm (`rain`) in those cities.

We consider logistic regression models that assume that the numbers $r_1, \ldots, r_n$ of people testing positive for toxoplasmosis in the $n = 34$ cities are realizations of independent random variables $R_1, \ldots, R_n$, conditionally on a function of the annual rainfall x_i, and that $R_i \mid x_i \sim \text{Binomial}(m_i, \mu_i)$, where

$$\log \frac{\mu_i}{1-\mu_i} = \eta_i = \beta_1 + f(x_i) \,.$$

Let $Y_i = R_i/m_i$ be the proportion of people testing positive in the ith city. Because of the assumption of a binomial response distribution, a logistic regression model sets the response variance to $\text{var}(Y_i \mid x_i) = \mu_i(1-\mu_i)/m_i$.

If we consider logistic models with a linear predictor that implies a polynomial dependence on rainfall, AIC and stepwise selection methods both prefer a cubic model. For simplicity here, we compare a cubic model and an intercept-only model, in which there is no dependence on rainfall. Figure 2.1 shows the fitted proportions testing positive under the two models.

```
data("toxo", package = "SMPracticals")
mod_const <- glm(r/m ~ 1, data = toxo, weights = m,
                 family = "binomial")
mod_cubic <- glm(r/m ~ poly(rain, 3), data = toxo, weights = m,
                 family = "binomial")
plot(r/m ~ rain, data = toxo, xlab = "Annual rainfall (mm)",
     ylab = "Proportion testing positive for toxoplasmosis",
     bg = "#ff7518", pch = 21)
pred_cubic <- function(x)
    predict(mod_cubic, newdata = list(rain = x), type = "response")
abline(h = plogis(coef(mod_const)), lty = 2)
curve(pred_cubic, add = TRUE, lty = 3)
legend("topleft", legend = c("intercept only", "cubic"), lty = c(2, 3))
```

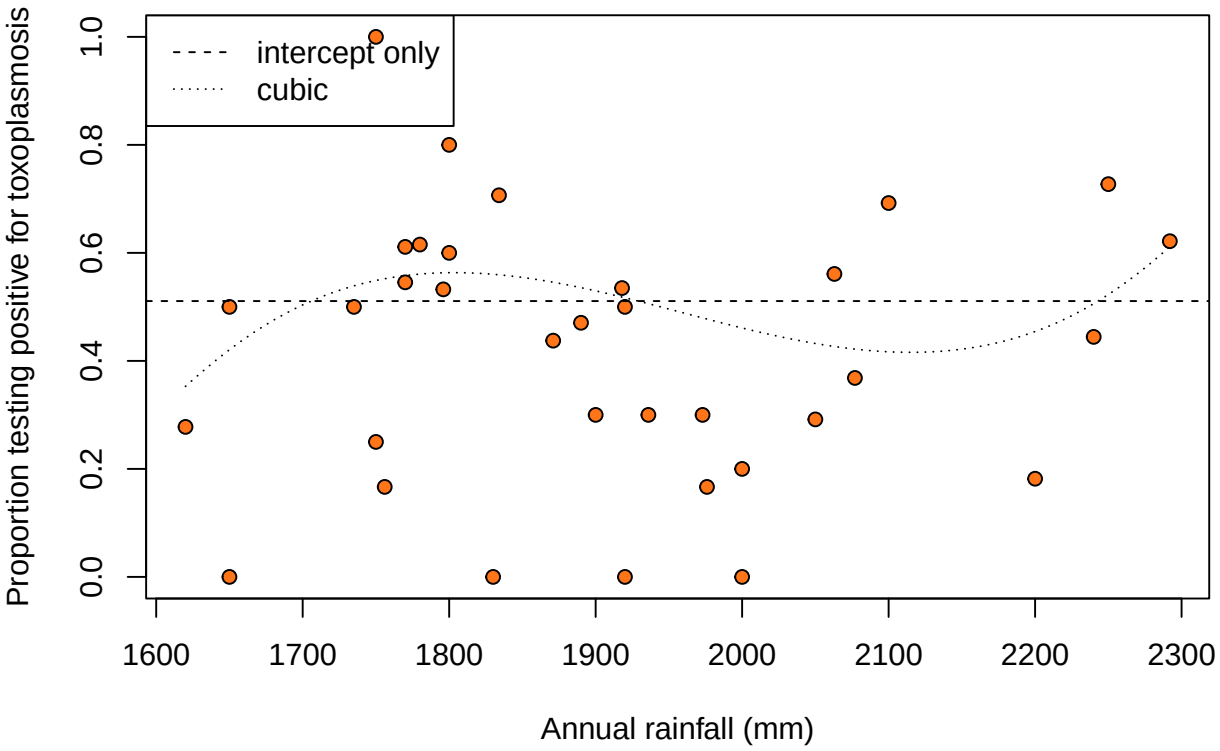


Figure 2.1: Proportion of people testing positive for toxoplasmosis against rainfall, with fitted proportions under an intercept-only (dashed line) and a cubic (dotted line) logistic regression model.

We can also compare the models using a hypothesis test:

```
anova(mod_const, mod_cubic, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: r/m ~ 1
```

```
Model 2: r/m ~ poly(rain, 3)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        33     74.212
2        30     62.635  3   11.577 0.008981 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is evidence against the intercept-only model, which implies no effect of rain on the probability of testing positive for toxoplasmosis, in favour of the cubic model.

However, we find that the residual deviance for the cubic model is 62.63, which is much larger than the residual degrees of freedom (30). That is evidence of a poor fit, and may be due to *overdispersion* of the responses. Overdispersion results in the residual variability being greater than what is prescribed by the mean / variance relationship of logistic regression.

2.2.2 Quasi-likelihood

A quasi-likelihood approach to accounting for overdispersion models the mean and variance, but stops short of a full probability model for the responses.

For a model specified by the mean relationship $g(\mu_i) = \eta_i$, and variance $\text{var}(Y_i \mid x_i) = \sigma^2 V(\mu_i)/m_i$, the quasi-likelihood equations are

$$\sum_{i=1}^{n} x_i \frac{y_i - \mu_i}{\sigma^2 V(\mu_i) g'(\mu_i)/m_i} = 0\,, \tag{2.1}$$

which can be solved with respect to β without knowledge of σ^2.

If $V(\mu_i)$ is the same function as in the definition of $\text{var}(Y_i \mid x_i)$ for an exponential family distribution, then it may be possible to solve (2.1) using standard GLM routines.

It can be shown that provided the mean and variance functions are correctly specified, asymptotic normality for $\widehat{\beta}$ still holds.

The dispersion parameter σ^2 can be estimated after estimating β, as

$$\hat{\sigma}^2 \equiv \frac{1}{n-p} \sum_{i=1}^{n} \frac{m_i (y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}\,.$$

Example 2.2 (Quasi-likelihood for the toxoplasmosis data)**.** In order to fit the same models as before, but with $\text{var}(Y_i \mid x_i) = \sigma^2 \mu_i (1 - \mu_i)/m_i$, we do

```
mod_const_quasi <- glm(r/m ~ 1, data = toxo, weights = m,
                       family = "quasibinomial")
mod_cubic_quasi <- glm(r/m ~ poly(rain, 3), data = toxo, weights = m,
                       family = "quasibinomial")
```

Comparing the output from fitting the logistic regression with a cubic relationship to rainfall

```
summary(mod_cubic)
```

```
Call:
glm(formula = r/m ~ poly(rain, 3), family = "binomial", data = toxo,
    weights = m)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.02427    0.07693   0.315 0.752401
poly(rain, 3)1 -0.08606    0.45870  -0.188 0.851172
poly(rain, 3)2 -0.19269    0.46739  -0.412 0.680141
```

```
poly(rain, 3)3  1.37875    0.41150   3.351 0.000806 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 74.212  on 33  degrees of freedom
Residual deviance: 62.635  on 30  degrees of freedom
AIC: 161.33

Number of Fisher Scoring iterations: 3
```

to the corresponding output from solving the quasi-likelihood equations

```
summary(mod_cubic_quasi)
```

```
Call:
glm(formula = r/m ~ poly(rain, 3), family = "quasibinomial",
    data = toxo, weights = m)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.02427    0.10716   0.226   0.8224
poly(rain, 3)1 -0.08606    0.63897  -0.135   0.8938
poly(rain, 3)2 -0.19269    0.65108  -0.296   0.7693
poly(rain, 3)3  1.37875    0.57321   2.405   0.0225 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.940446)

    Null deviance: 74.212  on 33  degrees of freedom
Residual deviance: 62.635  on 30  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 3
```

we observe that the estimates of the β coefficients are the same, but the estimated standard errors from quasi-likelihood are inflated by a factor of 1.39, which is equal to the square root of the estimate $\hat{\sigma}^2 = 1.94$ of σ^2. Note that the value of $\hat{\sigma}^2$ is about double than $\sigma^2 = 1$, implied by logistic regression.

A comparison of two quasi-likelihood fits is usually performed by mimicking the F test for nested linear regression models, using residual deviances in place of residual sums of squares. The resulting F statistic is asymptotically valid. To illustrate, we compare the value of F statistic

$$F = \frac{(74.21 - 62.63)/(33 - 30)}{1.94},$$

to quantiles of its asymptotic F distribution with 33 - 30 and 30 degrees of freedom. In R, we get

```
anova(mod_const_quasi, mod_cubic_quasi, test = "F")
```

```
Analysis of Deviance Table

Model 1: r/m ~ 1
Model 2: r/m ~ poly(rain, 3)
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1        33     74.212
2        30     62.635  3   11.577 1.9888 0.1369
```

After accounting for overdispersion, the evidence in favour of an effect of rainfall on toxoplasmosis incidence is less compelling.

2.2.3 Parametric models for overdispersion

To construct a full probability model in the presence of overdispersion, it is necessary to consider the reasons for the presence of overdispersion. Possible reasons include:

- There may be important covariates, other than rainfall, which are not observed.
- There may be many other features of the cities, possibly unobservable, all having a small individual effect on incidence, but a larger effect in combination. Such effects may be individually undetectable, a phenomenon sometimes described as *natural excess variability between units*.

Suppose that part of the linear predictor is missing from the model, that is the actual predictor is

$$\eta_i^* = \eta_i + \zeta_i \,,$$

instead of just η_i, where ζ_i may involve covariates z_i that are different than those in x_i. We can compensate for the missing term ζ_i by assuming that it has a distribution F in the population. Hence,

$$\mu_i = \mathrm{E}(Y_i \mid x_i, \zeta_i) = g^{-1}(\eta_i + \zeta_i) \sim G \,,$$

where G is the distribution induced by F. Then,

$$\begin{aligned} \mathrm{E}(Y_i \mid x_i) &= \mathrm{E}_G(\mu_i) \,, \\ \mathrm{var}(Y_i \mid x_i) &= \mathrm{E}_G(\mathrm{var}(Y_i \mid x_i, \zeta_i)) + \mathrm{var}_G(\mu_i) \,. \end{aligned}$$

For exponential family models, we get $\mathrm{var}(Y_i \mid x_i) = \sigma^2 \mathrm{E}_G(V(\mu_i))/m_i + \mathrm{var}_G(\mu_i)$.

One approach is to model the Y_i directly, by specifying an appropriate form for G. For example, for the toxoplasmosis data, instead of a quasi-likelihood approach, we can use a *beta-binomial* model, where

$$\begin{aligned} Y_i &= R_i/m_i \,, \\ R_i \mid \mu_i &\overset{\text{ind}}{\sim} \mathrm{Binomial}(m_i, \mu_i) \,, \\ \mu_i \mid x_i &\overset{\text{ind}}{\sim} \mathrm{Beta}(k\mu_i^*, k(1-\mu_i^*)) \,, \\ \log\{\mu_i^*/(1-\mu_i^*)\} &= \eta_i \,, \end{aligned}$$

leading to

$$\mathrm{E}(Y_i \mid x_i) = \mu_i^* \quad \text{and} \quad \mathrm{var}(Y_i \mid x_i) = \frac{\mu_i^*(1-\mu_i^*)}{m_i}\left(1 + \frac{m_i - 1}{k+1}\right) \,,$$

with $(m_i - 1)/(k+1)$ representing the overdispersion factor.

Another, popular in practice, model that accounts for overdispersion in count responses assumes a Poisson distribution for the responses and that the Poisson mean has a gamma distribution. This leads to a *negative binomial* marginal distribution for the responses.

Models that explicitly account for overdispersion can, in principle, be fitted with popular estimation methods, such as maximum likelihood. For example, the beta-binomial model has likelihood proportional to

$$\prod_{i=1}^{n} \frac{\Gamma(k\mu_i^* + m_i y_i)\Gamma(k(1-\mu_i^*) + m_i(1-y_i))\Gamma(k)}{\Gamma(k\mu_i^*)\Gamma(k(1-a\mu_i^*))\Gamma(k+m_i)} \,.$$

However, these models tend to have limited flexibility, and maximization of the likelihood can be difficult. For those reasons, practitioners typically resort to alternative approaches.

A more flexible, and extensible approach models the excess variability by including an extra term in the linear predictor

$$\eta_i = x_i^\top \beta + b_i \tag{2.2}$$

where $b_1, \ldots, b_n$ can be thought of as representing the extra, unexplained by the covariates, variability between units, and are called *random effects*. The model is completed by specifying a distribution F for the random effects in the population. A typical assumption is that $b_1, \ldots, b_n$ are independent with $b_i \sim \mathrm{N}(0, \sigma_b^2)$, for some unknown σ_b^2. We set $\mathrm{E}(b_i) = 0$, as an unknown mean for b_i would be unidentifiable in the presence of the intercept parameter in η_i.

Let $f(y_i \mid x_i, b_i\,;\, \theta)$ be the density or probability mass function of the chosen exponential family distribution, with linear predictor (2.2), and $f(b_i \mid \sigma_b^2)$ the density function of a univariate normal distribution with mean 0 and variance σ_b^2. The likelihood about the parameters $(\theta^\top, \sigma_b^2)^\top$ of the random effects model is

$$\begin{aligned} f(y \mid X\,;\, \theta, \sigma_b^2) &= \int f(y \mid X, b\,;\, \theta, \sigma_b^2) f(b \mid X\,;\, \theta, \sigma_b^2) db \\ &= \int f(y \mid X, b\,;\, \theta) f(b\,;\, \sigma_b^2) db \\ &= \int \prod_{i=1}^{n} f(y_i \mid x_i, b_i\,;\, \theta) f(u_i\,;\, \sigma_b^2) db_i\,. \end{aligned} \tag{2.3}$$

Depending on what $f(y_i \mid x_i, b_i\,;\, \theta)$ is, no further simplification of (2.3) may be possible, and computation needs careful consideration. We will briefly discuss such points later.

2.3 Dependence

2.3.1 An example

Example 2.3 (Toxoplasmosis data (revisited))**.** We can think of the toxoplasmosis cases in the ith city arising as $R_i = \sum_{j=1}^{m_i} Y_{ij}$, where Y_{ij} is a Bernoulli random variable, representing the toxoplasmosis status of individual j, with probability

$$\log \frac{\mu_{ij}}{1 - \mu_{ij}} = \eta_i = \beta_1 + f(x_i)\,. \tag{2.4}$$

From the properties of the binomial distirbution, if $\{Y_{ij}\}$ are independent, then the logistic regresion model on $\{R_i\}$ in Example 2.1 is the same to the Bernoulli model on $\{Y_{ij}\}$. However, suppose that the only assumptions we can confidently make is that $\{R_i\}$ are conditionally independent given the covariates and that $\mathrm{cov}(Y_{ij}, Y_{ik} \mid x_i) \neq 0$. Then,

$$\begin{aligned} \mathrm{var}(Y_i \mid x_i) &= \frac{1}{m_i^2} \left\{ \sum_{j=1}^{m_i} \mathrm{var}(Y_{ij} \mid x_i) + \sum_{j \neq k} \mathrm{cov}(Y_{ij}, Y_{ik} \mid x_i) \right\} \\ &= \frac{\mu_i(1 - \mu_i)}{m_i} + \frac{1}{m_i^2} \sum_{j \neq k} \mathrm{cov}(Y_{ij}, Y_{ik} \mid x_i)\,. \end{aligned}$$

As a result, positive correlation between individuals in the same city induces overdispersion in the number of positive cases.

There may be a number of plausible reasons why there is dependence between responses corresponding to units within a given *cluster* (in the toxoplasmosis example, clusters are cities). One compelling reason, that we discussed already, is unobserved heterogeneity.

In the correct model (corresponding to η_i^*), the toxoplasmosis status of individuals, Y_{ij}, may be independent, so

$$Y_{ij} \perp\!\!\!\perp Y_{ik} \mid x_i, \zeta_i \quad (j \neq k)\,.$$

However, in the absence of knowledge of ζ_i, it may be the case that

$$Y_{ij} \not\perp\!\!\!\perp Y_{ik} \mid x_i \quad (j \neq k)\,.$$

Hence conditional (given ζ_i) independence between units in a common cluster i becomes marginal dependence, when marginalised over the population distribution F of unobserved ζ_i.

The correspondence between positive intra-cluster correlation and unobserved heterogeneity suggests that intra-cluster dependence might be effectively modelled using random effects. For example, for the individual-level toxoplasmosis data

$$\begin{aligned} Y_{ij} \mid x_i, b_i &\overset{\text{ind}}{\sim} \text{Bernoulli}(\mu_{ij}) \,, \\ \log \frac{\mu_{ij}}{1-\mu_{ij}} &= \beta_1 + f(x_i) + b_i \,, \\ b_i &\overset{\text{ind}}{\sim} \text{N}(0, \sigma_b^2) \,, \end{aligned}$$

which, for $\sigma^2 > 0$, implies

$$Y_{ij} \not\perp\!\!\!\perp Y_{ik} \mid x_i \,.$$

Intra-cluster dependence arises in many applications, and random effects provide an effective way of modelling it.

2.3.2 Marginal models

Another way to account for intra-cluster dependence are marginal models. A *marginal model* expresses $\mu_{ij} = \text{E}(Y_{ij} \mid \eta_{ij})$ as a function of explanatory variables, through $g(\mu_{ij}) = \eta_{ij} = x_{ij}^\top \beta$, specifies a variance relationship $\text{var}(Y_{ij} \mid x_{ij}) = \sigma^2 V(\mu_{ij})/m_{ij}$, and models $\text{corr}(Y_{ij}, Y_{ik} \mid x_{ij}, x_{ik})$, as a function of μ and possibly additional parameters.

It is important to note that the parameters β in a marginal model have a different interpretation from those in a random effects model, because for the latter

$$\text{E}(Y_{ij} \mid x_{ij}) = \text{E}(g^{-1}[x_{ij}^\top \beta + b_i]) \neq g^{-1}(x_{ij}^\top \beta) \quad \text{if } g \text{ is not linear} \,.$$

A random effects model describes the mean response at the subject level (*subject specific*), while a marginal model describes the mean response across the population (*population averaged*).

As with the quasi-likelihood approach above, marginal models do not generally provide a full probability model for the responses. Nevertheless, β can be estimated using *generalized estimating equations (GEEs)*.

The GEE estimator of $ in a marginal model is the solution of of the form

$$\sum_i \left(\frac{\partial \mu_i}{\partial \beta} \right)^\top \text{var}(Y_i)^{-1} (Y_i - \mu_i) = 0 \,,$$

where $Y_i = (Y_{i1}, \ldots, Y_{in_i})^\top$ and $\mu_i = (\mu_{i1}, \ldots, \mu_{in_i})^\top$, with n_i indicating the number of observations in the ith response vector.

There are several consistent estimators for the covariance of GEE estimators. Furthermore, the GEE approach is generally robust to misspecification of the correlation structure.

2.3.3 Clustered data

Examples where data are collected in clusters include:

- Studies in biometry where *repeated measurements* are made on experimental units. Such studies can effectively mitigate the effect of between-unit variability on important inferences.
- Agricultural field trials, or similar studies, for example in engineering, where experimental units are arranged within *blocks*.
- Sample surveys where collecting data within clusters or *small areas* can save costs.

Of course, other forms of dependence exist, for example spatial or serial dependence induced by the arrangement of units of observation in space or time.

Example 2.4. The `rat.growth` data in the `SMPracticals` R package gives the weekly weights (`y`) of 30 young rats. Figure 2.2 shows the weight evolution for each rat. While the weights of each rat appears to grow roughly linearly with time, the intercept and slope of that weight evolution seem to vary between rats.

```
data("rat.growth", package = "SMPracticals")
plot(y ~ week, data = rat.growth, type = "n", xlab = "Week", ylab = "Weight")
for (i in 1:30) {
    dat_i <- subset(rat.growth, rat == i)
    lines(y ~ week, data = dat_i, col = "grey")
}
```

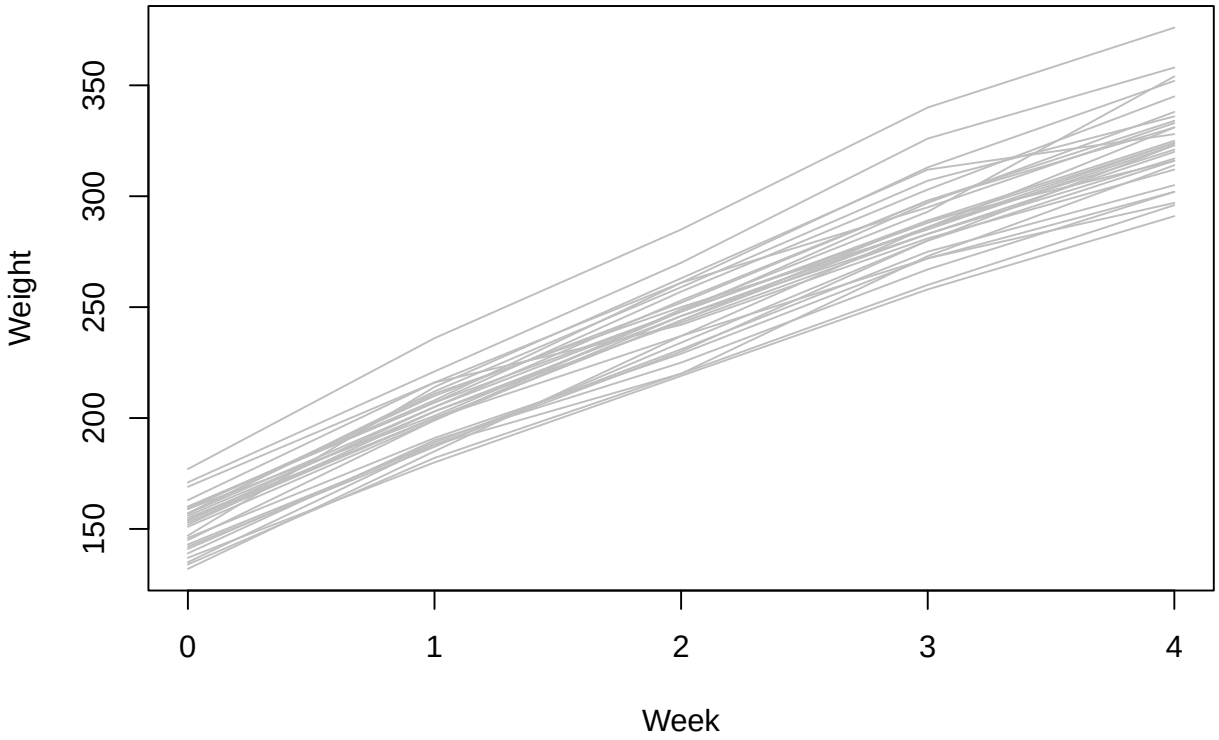


Figure 2.2: Individual rat weight by week, for the rat growth data.

Writing y_{ij} for the weight of rat i at week x_{ij}, we consider the simple linear regression

$$Y_{ij} = \beta_1 + \beta_2 x_{ij} + \epsilon_{ij}\,,$$

and fit it using R:

```
rat_lm <- lm(y ~ week, data = rat.growth)
(rat_lm_sum <- summary(rat_lm))
```

```
Call:
lm(formula = y ~ week, data = rat.growth)

Residuals:
   Min     1Q Median     3Q    Max
-38.12 -11.25   0.28   7.68  54.15

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  156.053      2.246   69.47   <2e-16 ***
```

```
week          43.267      0.917   47.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.88 on 148 degrees of freedom
Multiple R-squared:  0.9377,    Adjusted R-squared:  0.9372
F-statistic:  2226 on 1 and 148 DF,  p-value: < 2.2e-16
```

The resulting estimates are $\hat{\beta}_1 = 156.05$ and $\hat{\beta}_2 = 43.27$, with estimated standard errors 2.25 and 0.92, respectively.

Figure 2.3 shows boxplots of the residuals from this model, separately for each rat. There is clear evidence of unexplained differences between rats.

```
res <- residuals(rat_lm, type = "pearson")
ord <- order(ave(res, rat.growth$rat))
rats <- rat.growth$rat[ord]
rats <- factor(rats, levels = unique(rats), ordered = TRUE)
plot(res[ord] ~ rats,
     xlab = "Rat (ordered by mean residual)",
     ylab = "Pearson residual",
     col = "#ff7518", pch = 21)
abline(h = 0, lty = 2)
```

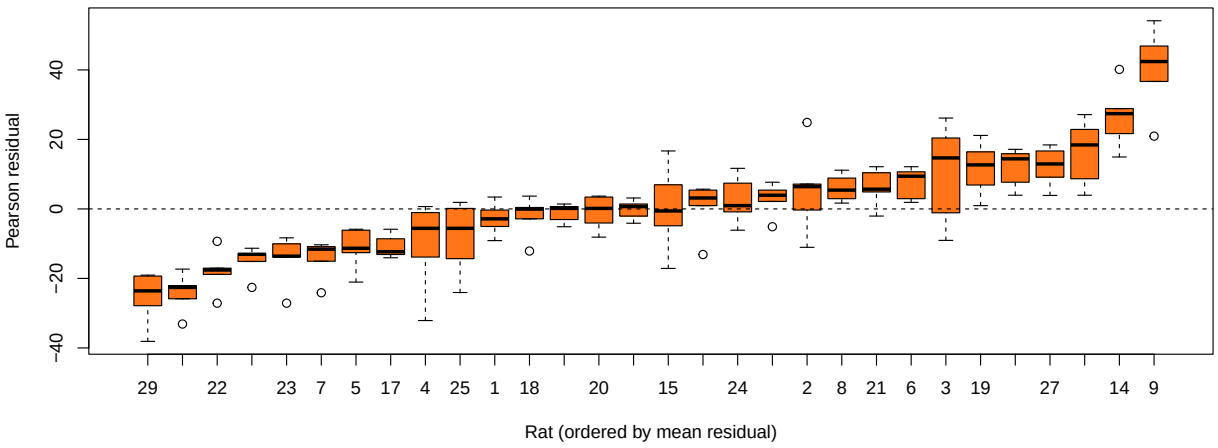


Figure 2.3: Boxplots of residuals from a simple linear regression, for each rat in the rat growth data.

2.4 Linear mixed models

2.4.1 Model definition

A linear mixed model (LMM) for observations $y = (y_1, \ldots, y_n)^\top$ has the general form

$$\begin{aligned} Y \mid X, Z, b &\sim \mathrm{N}(\mu, \Sigma)\,, \\ \mu &= X\beta + Zb\,, \\ b &\sim \mathrm{N}(0, \Sigma_b)\,. \end{aligned} \tag{2.5}$$

where X and Z are matrices containing covariate values. Usually, $\Sigma = \sigma^2 I_n$. The unknown parameters to be estimated are β, Σ, and Σ_b. The term *mixed model* highlights that the linear predictor $X\beta + Zb$ contains both the fixed effects β and the random effects b.

A typical example for clustered data is

$$\begin{aligned} Y_{ij} \mid x_{ij}, z_{ij}, b_i &\overset{\text{ind}}{\sim} \mathrm{N}(\mu_{ij}, \sigma^2)\,, \\ \mu_{ij} &= x_{ij}^\top \beta + z_{ij}^\top b_i\,, \\ b_i &\overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b^*)\,, \end{aligned} \tag{2.6}$$

where x_{ij} contains the covariates for observation j of the ith cluster, and z_{ij} the covariates which are allowed to exhibit extra between-cluster variation in their relationship with the response. Typically, z_{ij} is a sub-vector of x_{ij}, but this is not necessary.

The simplest case of a mixed effects linear predictor arises when $z_{ij} = 1$, which results in a random-intercept model, as in (2.2).

A plausible LMM for k clusters with n_i observations in the ith cluster, and a single explanatory variable (see, for example, Example 2.4) has

$$Y_{ij} = \beta_1 + b_{1i} + (\beta_2 + b_{2i})x_{ij} + \epsilon_{ij}\,, \quad (b_{1i}, b_{2i})^\top \overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b^*)\,.$$

This fits into the general LMM definition in (2.5) with $\Sigma = \sigma^2 I_n$, and

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_k \end{bmatrix}, \quad Y_i = \begin{bmatrix} Y_{i1} \\ \vdots \\ Y_{in_i} \end{bmatrix},$$

$$X = \begin{bmatrix} X_1 \\ \cdots \\ X_k \end{bmatrix}, \quad Z = \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & X_k \end{bmatrix}, \quad X_i = \begin{bmatrix} 1 & x_{i1} \\ \vdots & \vdots \\ 1 & x_{in_i} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{1k} \\ b_{2k} \end{bmatrix}, \quad \Sigma_b = \begin{bmatrix} \Sigma_b^* & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_b^* \end{bmatrix},$$

where Σ_b^* is an unknown 2×2 positive definite matrix, and 0 denotes a matrix of zeros of appropriate dimension.

Under an LMM, we can write the marginal distribution of Y directly as

$$Y \mid X, Z \sim \mathrm{N}(X\beta, \Sigma + Z\Sigma_b Z^\top) \tag{2.7}$$

Hence, $\mathrm{var}(Y \mid X, Z)$ is comprised of two *variance components*.

LMMs for clustered data, such as (2.6) are also known as *hierarchical* or *multilevel* models. This reflects the two-stage structure of the model definition; a conditional model for the responses given covariates and the random effects, followed by a marginal model for the random effects.

Sometimes the hierarchy can have further levels, corresponding to clusters nested within clusters. Common practical settings of that kind is patients within wards within hospitals, or pupils within classes within schools.

2.4.2 Random effects or cluster-specific fixed effects

Instead of including random effects for clusters, e.g.

$$Y_{ij} = \beta_1 + b_{1i} + (\beta_2 + b_{2i})x_{ij} + \epsilon_{ij}\,,$$

we could use separate fixed effects for each cluster, e.g.

$$Y_{ij} = \beta_{1i} + \beta_{2i}x_{ij} + \epsilon_{ij}\,.$$

However, inferences can then only be made about those clusters present in the observed data. Random effects models allow inferences to be extended to a wider population. It also can be the case that fixed effects are not identifiable. That is the case in the setting of Example 2.1 where there is only one observation per cluster. In contrast, random effects are identifiable and can be estimated. Random effects also allow *borrowing strength* across clusters by shrinking fixed effects towards a common mean.

2.4.3 Estimation

The likelihood about β, Σ, Σ_b is available from (2.7) as

$$f(y \mid X, Z\,;\, \beta, \Sigma, \Sigma_b) \propto |V|^{-1/2} \exp\left(-\frac{1}{2}(y - X\beta)^\top V^{-1}(y - X\beta)\right) \tag{2.8}$$

where $V = \Sigma + Z\Sigma_b Z^\top$, and can be directly maximized.

However, the maximum likelihood estimators for the variance parameters of LMMs can have considerable downward bias. The bias is more profound in cluster models with a small number of observed clusters, and can affect the performance of inferential procedures. Hence estimation by *REML* (*REstricted* or *REsidual* Maximum Likelihood) is usually preferred.

REML proceeds by estimating the variance parameters Σ and Σ_b using a *marginal likelihood* based on the residuals from a least squares fit of the model $E(Y \mid X) = X\beta$.

Consider the vector of residuals $(I_n - H)Y$, where $H = X(X^\top X)^{-1}X^\top$ is the usual hat matrix. The distribution of $(I_n - H)Y$ does not depend of β, but because $(I_n - H)$ has rank $n - p$, that distribution is degenerate. From the spectral decomposition theorem, we can always define a vector of $(n - p)$ random variables $U = B^\top Y$, where B is any $n \times (n-p)$ matrix of rank $(n-p)$ with $BB^\top = I_n - H$ and $B^\top B = I_{n-p}$. Then, $B^\top X = B^\top BB^\top X = B(I_n - H)X = 0$, and

$$U = B^\top Y = B^\top(X\beta + A) = B^\top A\,,$$

where $A \mid Z \sim \mathrm{N}(0, V)$. Hence, $U \mid X, Z \sim \mathrm{N}(0, B^\top V B)$, which does not depend on β. That observation may, at first sight, appear as simply trading the dependence on β with the dependence on B, which would be useless for practical purposes because B is not uniquely defined. However, it can be shown that the distribution of U depends neither on β nor on the choice of B!

To see that, note that the least squares estimator of β for known V is

$$\hat{\beta}_V = (X^\top V^{-1} X)^{-1} X^\top V^{-1} Y = \beta + (X^\top V^{-1} X)^{-1} X^\top V^{-1} A\,. \tag{2.9}$$

So, $\hat{\beta}_V - \beta \mid X, Z \sim \mathrm{N}(0, (X^\top V^{-1} X)^{-1})$. Also,

$$\mathrm{E}(U(\hat{\beta}_V - \beta)^\top \mid X, Z) = B^\top \mathrm{E}(AA^\top \mid Z) V^{-1} X (X^\top V^{-1} X)^{-1} = 0$$

Hence, since U and $\hat{\beta}_V - \beta$ are both normally distributed, they are independent.

Temporarily suppressing the conditioning on X and Z in the notation, we can write

$$\begin{aligned} f(u\,;\, \Sigma, \Sigma_b) f(\hat{\beta}_V\,;\, \beta, \Sigma, \Sigma_b) &= f(u, \hat{\beta}_V\,;\, \beta, \Sigma, \Sigma_b) \\ &= f(Qy\,;\, \beta, \Sigma, \Sigma_b) = f(y\,;\, \beta, \Sigma, \Sigma_b) |Q^\top|^{-1}\,, \end{aligned} \tag{2.10}$$

where $Q = \begin{bmatrix} B^\top \\ G^\top \end{bmatrix}$, $G^\top = (X^\top V^{-1} X)^{-1} X^\top V^{-1}$, and $|Q^\top|$ is the Jacobian determinant of the transformation Qy. We have

$$\begin{aligned} |Q^\top| &= \left|\begin{bmatrix} B & G \end{bmatrix}\right| \\ &= \left|\begin{bmatrix} B^\top \\ G^\top \end{bmatrix} \begin{bmatrix} B & G \end{bmatrix}\right|^{1/2} = \left|\begin{bmatrix} B^\top B & B^\top G \\ G^\top B & G^\top G \end{bmatrix}\right|^{1/2} \\ &= |B^\top B|^{1/2} |G^\top G - G^\top B (B^\top B)^{-1} B^\top G|^{1/2} \\ &= |G^\top G - G^\top (I_n - H) G|^{1/2} \\ &= |(X^\top V^{-1} X)^{-1} X^\top V^{-1} X (X^\top X)^{-1} X^\top V^{-1} X (X^\top V^{-1} X)^{-1}|^{1/2} \\ &= |X^\top X|^{-1/2}\,. \end{aligned}$$

So, from expression (2.10), a completion of the square in the ratio of the normal densities of y and $\hat{\beta}_V$ gives

$$\begin{aligned} f(u\,;\,\Sigma,\Sigma_b) &= \frac{f(y\,;\,\beta,\Sigma,\Sigma_b)}{f(\hat{\beta}_V\,;\,\beta,\Sigma,\Sigma_b)}|X^\top X|^{1/2} \\ &\propto \frac{|X^\top X|^{1/2}}{|V|^{1/2}|X^\top V^{-1}X|^{1/2}} \exp\left(-\frac{1}{2}(y-X\hat{\beta}_V)^\top V^{-1}(y-X\hat{\beta}_V)\right), \end{aligned} \tag{2.11}$$

which does not involve B. Note that the maximized marginal likelihood cannot be used to compare different fixed effects specifications, due to the dependence of U on X.

2.4.4 Estimating random effects

A natural predictor $\tilde{b}$ of the random effect vector b is obtained by minimizing the mean squared prediction error $\mathrm{E}((\tilde{b}-b)^\top(\tilde{b}-b)\mid X,Z)$ where the expectation is over both b and Y. This is achieved by

$$\tilde{b} = \mathrm{E}(b\mid Y,X,Z) = (Z^\top\Sigma^{-1}Z+\Sigma_b^{-1})^{-1}Z^\top\Sigma^{-1}(Y-X\beta)\,, \tag{2.12}$$

which is the *Best Linear Unbiased Predictor* (BLUP) for b, with corresponding variance

$$\mathrm{var}(b\mid Y,X,Z) = (Z^\top\Sigma^{-1}Z+\Sigma_b^{-1})^{-1}\,. \tag{2.13}$$

We can obtain estimates of (2.13) and (2.13) by plugging in $\hat{\beta}$, $\hat{\Sigma}$, and $\hat{\Sigma}_b$. The estimates of $\tilde{b}$ are typically *shrunk* towards 0 relative to the corresponding fixed effects estimates.

Any component b_k of b with no relevant data (for example, a cluster effect for an as yet unobserved cluster) corresponds to a null column of Z. In that case, $\tilde{b}_k = 0$ and $\mathrm{var}(b_k\mid Y,X,Z) = [\Sigma_b]_{kk}$, which can be estimated in the common case that b_k shares a variance with other random effects.

Example 2.5 (LMM for rat growth data)**.** Here, we consider the model

$$Y_{ij} = \beta_1 + b_{1i} + (\beta_2+b_{2i})x_{ij} + \epsilon_{ij}\,, \quad (b_{1i},b_{2i})^\top \overset{\text{ind}}{\sim} \mathrm{N}(0,\Sigma_b)\,,$$

where $\epsilon_{ij} \overset{\text{ind}}{\sim} \mathrm{N}(0,\sigma^2)$ and Σ_b is an unspecified covariance matrix. This model allows for random, cluster-specific slope and intercept.

We may fit the model in `R` using the methods in the `lme4` package:

```
library("lme4")
```

```
Loading required package: Matrix
```

```
rat_rs <- lmer(y ~ week + (week | rat), data = rat.growth)
rat_rs
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ week + (week | rat)
   Data: rat.growth
REML criterion at convergence: 1084.58
Random effects:
 Groups   Name        Std.Dev. Corr
 rat      (Intercept) 10.933
          week         3.535   0.18
 Residual              5.817
Number of obs: 150, groups:  rat, 30
Fixed Effects:
(Intercept)         week
     156.05        43.27
```

Let's also consider the simpler random intercept model

$$Y_{ij} = \beta_1 + b_{1i} + \beta_2 x_{ij} + \epsilon_{ij}\,, \quad b_{1i} \overset{\text{ind}}{\sim} \mathrm{N}(0,\sigma_b^2)\,.$$

```
rat_ri <- lmer(y ~ week + (1 | rat), data = rat.growth)
rat_ri
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ week + (1 | rat)
   Data: rat.growth
REML criterion at convergence: 1127.169
Random effects:
 Groups   Name        Std.Dev.
 rat      (Intercept) 13.851
 Residual              8.018
Number of obs: 150, groups:  rat, 30
Fixed Effects:
(Intercept)         week
     156.05        43.27
```

We can compare the two models using AIC or BIC, but in order to do so we need to refit the models with maximum likelihood rather than REML.

```
rat_rs_ML <- lmer(y ~ week + (week | rat), data = rat.growth, REML = FALSE)
rat_ri_ML <- lmer(y ~ week + (1 | rat), data = rat.growth, REML = FALSE)
anova(rat_rs_ML, rat_ri_ML)
```

```
Data: rat.growth
Models:
rat_ri_ML: y ~ week + (1 | rat)
rat_rs_ML: y ~ week + (week | rat)
          npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
rat_ri_ML    4 1139.2 1151.2 -565.60   1131.2
rat_rs_ML    6 1101.1 1119.2 -544.56   1089.1 42.079  2  7.288e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By either criterion, there is evidence for the random slopes model.

An alternative model would be a fixed effects model with separate intercepts and slopes for each rat

$$Y_{ij} = \beta_{1i} + \beta_{2i} x_{ij} + \epsilon_{ij} \,.$$

Figure Figure 2.4 shows parameter estimates from the random effects model against those from the fixed effects model, illustrating the shrinkage of the random effect estimates towards a common mean. Random effects estimates 'borrow strength' across clusters, due to the common Σ_b^{-1} term in (2.12). The extent of borrowing strength is determined by cluster similarity.

```
ranef_est <- coef(rat_rs)$rat
rat_lm3 <- lm(y ~ rat * week, data = rat.growth)

rats <- factor(1:30, levels = 1:30)
pred_rat_0 <- predict(rat_lm3,
                      newdata = data.frame(rat = rats, week = 0))
pred_rat_1 <- predict(rat_lm3,
                      newdata = data.frame(rat = rats, week = 1))
fixef_est <- data.frame("(Intercept)" = pred_rat_0,
                        "week" = pred_rat_1 - pred_rat_0)
intercept_range <- range(c(fixef_est[,1], ranef_est[,1]))
slope_range <- range(c(fixef_est[,2], ranef_est[,2]))

par(mfrow = c(1, 2))
plot(fixef_est[,1], ranef_est[,1],
```

```
     xlab = "Intercept estimate (fixed effects)",
     ylab = "Intercept estimate (random effects)",
     xlim = intercept_range, ylim = intercept_range,
     bg = "#ff7518", pch = 21)
abline(lm(ranef_est[,1] ~ fixef_est[,1]), col = "grey")
abline(a = 0, b = 1, lty = 2)

plot(fixef_est[,2], ranef_est[,2],
     xlab = "Slope estimate (fixed effects)",
     ylab = "Slope estimate (random effects)",
     xlim = slope_range,
     ylim = slope_range,
     bg = "#ff7518", pch = 21)
abline(lm(ranef_est[,2] ~ fixef_est[,2]), col = "grey")
abline(a = 0, b = 1, lty = 2)
```

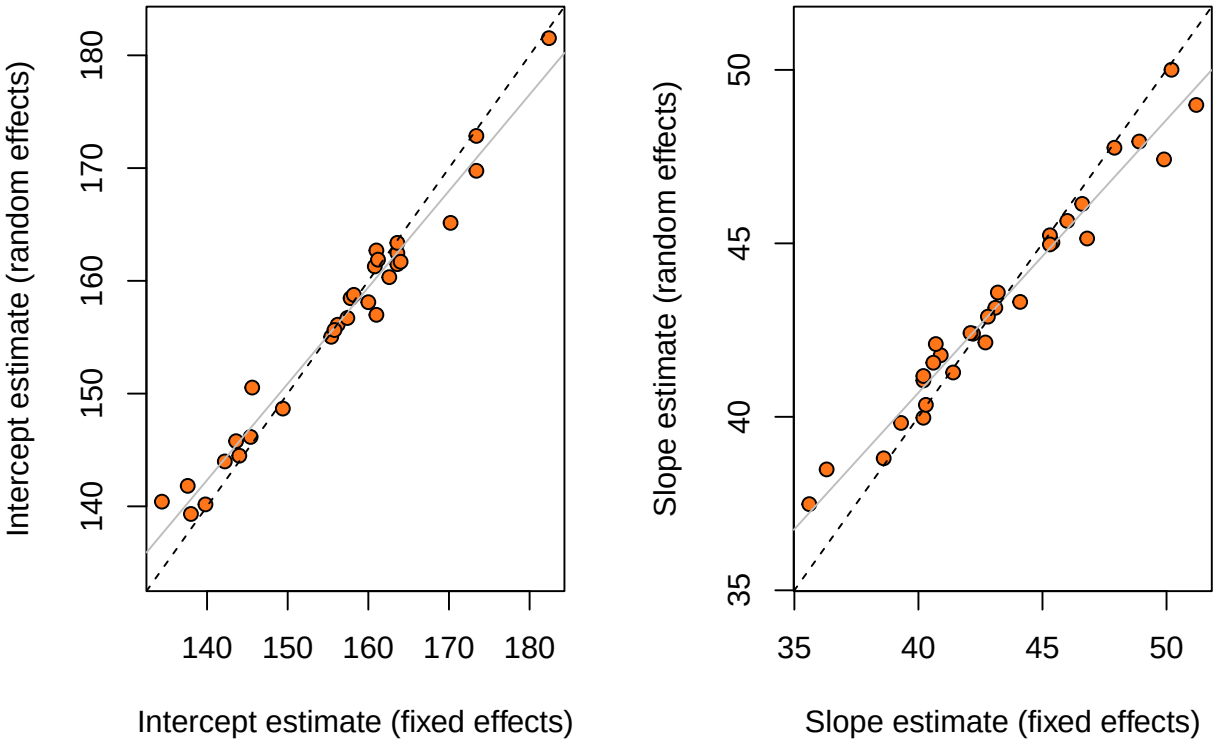


Figure 2.4: Estimates of random and fixed effects for the rat growth data. The dashed line is a line with intercept zero and slope 1, and the solid line is from the least squares fit of the observed points.

2.4.5 Bayesian inference: the Gibbs sampler

Bayesian inference for LMMs (and their generalizations, which we will introduce later) proceeds using *Markov Chain Monte Carlo* (MCMC) methods, such as the *Gibbs sampler*, that have proved very effective in practice.

MCMC computation provides posterior summaries, by *generating* a *dependent* sample from the posterior distribution of interest. Then, any posterior expectation can be estimated by the corresponding Monte Carlo sample mean, densities can be estimated from samples, etc.

MCMC will be covered in detail in the Computer Intensive Statistics APTS module. Here we simply describe the (most basic) Gibbs sampler.

To generate from $f(y_1, \ldots, y_n)$, where the components y_i are allowed to be multivariate, the Gibbs sampler

starts from an arbitrary value of y and updates components, sequentially or otherwise, by generating from the conditional distributions $f(y_i \mid y_{\,i})$ where $y_{\,i}$ are all the variables other than y_i, set at their currently generated values.

Hence, to apply the Gibbs sampler, we require conditional distributions which are available for sampling.

For the linear mixed model

$$Y \mid X, Z, \beta, \Sigma, b \sim \mathrm{N}(\mu, \Sigma), \quad \mu = X\beta + Zb, \quad b \mid \Sigma_b \sim \mathrm{N}(0, \Sigma_b)\,,$$

and prior densities $\pi(\beta)$, $\pi(\Sigma)$, $\pi(\Sigma_b)$, we obtain the *conditional* posterior distributions

$$\begin{aligned} f(\beta \mid y, \text{rest}) &\propto \phi(y - Zb\,;\, X\beta, V)\pi(\beta)\,,\\ f(b \mid y, \text{rest}) &\propto \phi(y - X\beta\,;\, Zb, V)\phi(b\,;\, 0, \Sigma_b)\,,\\ f(\Sigma \mid y, \text{rest}) &\propto \phi(y - X\beta - Zb\,;\, 0, V)\pi(\Sigma)\,,\\ f(\Sigma_b \mid y, \text{rest}) &\propto \phi(b\,;\, 0, \Sigma_b)\pi(\Sigma_b)\,, \end{aligned}$$

where $\phi(y\,;\,\mu, \Sigma)$ is the density of a $\mathrm{N}(\mu, \Sigma)$ random variable evaluated at y.

We can exploit *conditional conjugacy* in the choices of $\pi(\beta)$, $\pi(\Sigma)$, $\pi(\Sigma_b)$ making the conditionals above of known form and, hence, straightforward to sample from. The conditional independence $(\beta, \Sigma) \perp\!\!\!\perp \Sigma_b \mid b$ is also helpful in that direction.

2.5 Generalized linear mixed models

2.5.1 Model setup

Generalized linear mixed models (GLMMs) generalize LMMs to non-normal responses, similarly to how generalized linear models generalize normal linear models. A GLMM has

$$\begin{aligned} Y_i \mid x_i, z_i, b &\overset{\text{ind}}{\sim} \mathrm{EF}(\mu_i, \sigma^2)\,,\\ \begin{bmatrix} g(\mu_1) \\ \vdots \\ g(\mu_n) \end{bmatrix} &= X\beta + Zb\,,\\ b &\sim \mathrm{N}(0, \Sigma_b)\,, \end{aligned} \tag{2.14}$$

where $\mathrm{EF}(\mu_i, \sigma^2)$ is an exponential family distribution with mean μ_i and variance $\sigma^2 V(\mu_i)/m_i$ for known m_i. For many well-used distributions, like binomial and Poisson, $\sigma^2 = 1$. For the shake of not complicating presentation, we shall assume that from here on.

The normality of the random effects b can be relaxed in many ways and to other distributions, but normal random effects usually provide adequate fits. Non-normal random effects distributions are beyond the scope of this module.

Example 2.6. A random-intercept GLMM for binary data in k clusters with $n_1, \ldots, n_k$ observations per cluster, and a single explanatory variable x (e.g. the setting for the toxoplasmosis data at individual level) is

$$\begin{aligned} Y_{ij} \mid x_{ij}, b_i &\overset{\text{ind}}{\sim} \text{Bernoulli}(\mu_{ij})\\ \log \frac{\mu_{ij}}{1 - \mu_{ij}} &= \beta_1 + b_i + \beta_2 x_{ij}\\ b_i &\overset{\text{ind}}{\sim} \mathrm{N}(0, \sigma_b^2) \end{aligned} \tag{2.15}$$

This model fits into the general GLMM framework in (3.1) with

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_k \end{bmatrix}, \quad Y_i = \begin{bmatrix} Y_{i1} \\ \vdots \\ Y_{in_i} \end{bmatrix},$$

$$X = \begin{bmatrix} X_1 \\ \cdots \\ X_k \end{bmatrix}, \quad X_i = \begin{bmatrix} 1 & x_{i1} \\ \vdots & \vdots \\ 1 & x_{in_i} \end{bmatrix},$$

$$Z = \begin{bmatrix} Z_1 & 0 & \cdots & 0 \\ 0 & Z_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_k \end{bmatrix}, \quad Z_i = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}, \quad \Sigma_b = \sigma_b^2 I_k,$$

and $g(\mu) = \log\{\mu/(1-\mu)\}$.

2.5.2 GLMM likelihood

The marginal distribution of Y in a GLMM does not usually have a convenient closed-form representation.

$$\begin{aligned} f(y \mid X, Z\,;\, \beta, \Sigma_b) &= \int f(y \mid X, Z, b\,;\, \beta, \Sigma_b) f(b\,;\, \beta, \Sigma_b) db \\ &= \int f(y \mid X, Z, b\,;\, \beta) f(b\,;\, \Sigma_b) db \\ &= \int \prod_{i=1}^{n} f\left(y_i \mid X, Z, b\,;\, \beta\right) f(b\,;\, \Sigma_b) db\,. \end{aligned} \tag{2.16}$$

For *nested* random effects structures, some simplification is possible. For example, for (2.15),

$$f(y \mid X, Z\,;\, \beta, \sigma_b^2) = \prod_{i=1}^{k} \int \prod_j f\left(y_{ij} \mid x_i, b_i\,;\, \beta\right) \phi(b_i\,;\, 0, \sigma_b^2) db_i\,,$$

which is a product of one-dimensional integrals.

Fitting a GLMM by likelihood methods requires some method for approximating the integrals involved.

When the integrals are of low dimension, a reliable method is to use Gaussian quadrature (see APTS: Statistical Computing). For example, for an one-dimensional cluster-level random effect b_i we might use

$$\begin{aligned} &\int \prod_j f\left(y_{ij} \mid x_i, b_i\,;\, \beta\right) \phi(b_i\,;\, 0, \sigma_b^2) db_i \\ \approx &\sum_{q=1}^{Q} W_q \prod_j f\left(y_{ij} \mid x_i, \sqrt{2}\sigma_b B_q\,;\, \beta\right) / \sqrt{\pi}\,, \end{aligned}$$

for weights W_q and quadrature points B_q $(q = 1, \ldots, Q)$ chosen according to the Gauss-Hermite quadrature rule.

Effective quadrature approaches use information about the mode and dispersion of the integrand, which can be done adaptively. For multi-dimensional b_i, quadrature rules can be applied recursively, but performance in fixed-time diminishes rapidly with dimension.

An alternative approach is to use a Laplace approximation to the likelihood. Writing

$$h(b) = \prod_{i=1}^{n} f\left(y_i \mid X, Z, b\,;\, \beta\right) f(b\,;\, \Sigma_b)$$

for the integrand of the likelihood, a first-order Laplace approximation approximates $h(.)$ as an unnormalised multivariate normal density function

$$\tilde{h}(b) = c\, \phi_k(b\,;\, \hat{b}, Q)\,,$$

where $\hat{b}$ is found by maximizing $\log h(.)$ over b, the variance matrix Q is chosen so that the curvature of $\log h(.)$ and $\log \tilde{h}(.)$ agree at $\hat{b}$, and c is chosen so that $\tilde{h}(\hat{b}) = h(\hat{b})$. The first-order Laplace approximation is equivalent to adaptive Gaussian quadrature with a single quadrature point.

Likelihood inference for GLMMs remains an area of active research and vigorous debate. Quadrature-based procedures provides accurate approximations to the likelihood. For some model structures, particularly those with crossed rather than nested random effects, the likelihood integral may be high-dimensional, and it may be impractical to use quadrature. In such cases, a Laplace approximation has been found to be sufficiently accurate for most purposes, but its accuracy is not guaranteed for every model.

Another alternative is to use Penalized Quasi Likelihood (PQL), which is very fast but often inaccurate. In, particular, PQL can fail badly in some cases, particularly with binary observations, and its use is not recommended.

Example 2.7 (Toxoplasmosis data (revisited))**.** For the toxoplasmosis data in Example 2.1, Table 2.1 gives the estimates and associated standard errors for the parameters of individual-level model (2.15), after dividing the annual rainfall by 10^5. The fits are obtained using maximum likelihood (with 25 quadrature points), Laplace approximation, and PQL.

```
library("MASS")
library("lme4")
library("modelsummary")
toxo$city <- 1:nrow(toxo)
toxo$rain_s <- toxo$rain / 100000
mod_lmm_quad <- glmer(r/m ~ rain_s + (1 | city), weights = m,
                      data = toxo, family = binomial, nAGQ = 25)
mod_lmm_LA <- glmer(r/m ~ rain_s + (1 | city), weights = m,
                    data = toxo, family = binomial)
mod_lmm_PQL <- glmmPQL(fixed = r/m ~ rain_s, random = ~ 1 | city, weights = m,
                       data = toxo, family = binomial)
modelsummary(
    list("GH (25)" = mod_lmm_quad,
         "Laplace" = mod_lmm_LA,
         "PQL" = mod_lmm_PQL),
    output = "markdown",
    escape = FALSE,
    gof_map = "none",
    fmt = fmt_decimal(digits = 3),
    coef_omit = 4,
    coef_rename = c("$\\beta_1$", "$\\beta_2$", "$\\sigma_b$"))
```

Table 2.1: Estimates and associated standard errors for the parameters of the linear individual-level model (2.15). 'GH (25)' is maximum likelihood using 25 Gauss-Hermite quadrature points, 'Laplace' is maximum approximate likelihood based on Laplace approximation, and 'PQL' is maximum PQL.

	GH (25)	Laplace	PQL
β_1	-0.138	-0.134	-0.115
	(1.450)	(1.441)	(1.445)
β_2	0.722	0.592	0.057
	(75.060)	(74.620)	(74.922)
σ_b	0.521	0.513	0.495

Table 2.2 shows the estimates for the corresponding GLMM, when the conditional mean of the incidence in toxoplasmosis is associated with an orthogonal polynomial to rainfall (after division by 10^5); see `?poly`.

```
mod_lmm_cubic_quad <- glmer(r/m ~ poly(rain_s, 3) + (1 | city), weights = m,
                            data = toxo, family = binomial, nAGQ = 25)
mod_lmm_cubic_LA <- glmer(r/m ~ poly(rain_s, 3) + (1 | city), weights = m,
                      data = toxo, family = binomial)
mod_lmm_cubic_PQL <- glmmPQL(fixed = r/m ~ poly(rain_s, 3), random = ~ 1 | city, weights = m,
                        data = toxo, family = binomial)
modelsummary(
    list("GH (25)" = mod_lmm_cubic_quad,
         "Laplace" = mod_lmm_cubic_LA,
         "PQL" = mod_lmm_cubic_PQL),
    output = "markdown",
    escape = FALSE,
    gof_map = "none",
    fmt = fmt_decimal(digits = 3),
    coef_omit = 6,
    coef_rename = c("$\\beta_1$", "$\\beta_2$", "$\\beta_3$", "$\\beta_4$", "$\\sigma_b$"))
```

Table 2.2: Estimates and associated standard errors for the parameters of the cubic individual-level model. 'GH (25)' is maximum likelihood with 25 Gauss-Hermite quadrature points, 'Laplace' is maximum approximate likelihood based on Laplace approximation, and 'PQL' is maximum PQL.

	GH (25)	Laplace	PQL
β_1	-0.106	-0.104	-0.110
	(0.127)	(0.126)	(0.127)
β_2	-0.106	-0.107	-0.098
	(0.687)	(0.682)	(0.718)
β_3	0.154	0.149	0.173
	(0.700)	(0.695)	(0.724)
β_4	1.628	1.626	1.607
	(0.654)	(0.649)	(0.682)
σ_b	0.423	0.417	0.431

There is a good agreement between the different estimation methods for the models considered in this example. The AIC and BIC (using 25 Gauss-Hermite quadrature points for the likelihood approximation) for the linear and the cubic individual-level models are

```
anova(mod_lmm_cubic_quad, mod_lmm_quad)
```

```
Data: toxo
Models:
mod_lmm_quad: r/m ~ rain_s + (1 | city)
mod_lmm_cubic_quad: r/m ~ poly(rain_s, 3) + (1 | city)
                   npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
mod_lmm_quad          3 65.754 70.333 -29.877   59.754
mod_lmm_cubic_quad    5 63.840 71.472 -26.920   53.840 5.9139  2    0.05198 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.5.3 Bayesian inference for GLMMs

As for LMMs, Bayesian inference in GLMMs is typically based on the Gibbs sampler. For the GLMM in (3.1), with prior densities $\pi(\beta)$ and $\pi(\Sigma_b)$ and known σ^2, we obtain the *conditional* posterior distributions

$$\begin{aligned} f(\beta \mid y, \text{rest}) &\propto \pi(\beta) \prod_i f(y_i \mid X, Z, \beta, b) \\ f(b \mid y, \text{rest}) &\propto \phi(b\,;\, 0, \Sigma_b) \prod_i f(y_i \mid X, Z, \beta, b) \\ f(\Sigma_b \mid y, \text{rest}) &\propto \phi(b\,;\, 0, \Sigma_b)\pi(\Sigma_b)\,, \end{aligned}$$

For a conditionally conjugate choice of $\pi(\Sigma_b)$, $f(\Sigma_b \mid y, \text{rest})$ is straightforward to sample from. The conditionals for β and b are not generally available for direct sampling. However, there are a number of ways of modifying the basic Gibbs sampling to go around this.

Chapter 3

Nonlinear models

> The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.
>
> — *John W. Tukey (1915–2000)*
>
> in Tukey (1986). Sunset Salvo. The American Statistician, 40 (1), p. 74.

3.1 Nonlinear models with fixed effects

So far we have only considered models where the link function of the mean response is equal to a linear predictor. For example, GLMMs, which are the most general models we have seen so far, have

$$\begin{aligned} Y_i \mid x_i, z_i, b &\overset{\text{ind}}{\sim} \text{EF}(\mu_i, \sigma^2)\,, \\ \begin{bmatrix} g(\mu_1) \\ \vdots \\ g(\mu_n) \end{bmatrix} &= \eta = X\beta + Zb\,, \\ b &\sim \text{N}(0, \Sigma_b)\,, \end{aligned} \tag{3.1}$$

where $\text{EF}(\mu_i, \sigma^2)$ is an exponential family distribution with mean μ_i and variance $\sigma^2 V(\mu_i)/m_i$ for known m_i. The key point is that the predictor η is a linear function of the parameters. Linear models, generalized linear models and linear mixed models are all special cases of the GLMM.

Models with linear predictors form the basis of most applied statistical analyses. It is often the case, though, that there is no scientific reason to believe these linear models are true for a given application.

We begin by considering nonlinear extensions of the normal linear model

$$Y_i = x_i^\top \beta + \epsilon_i\,, \tag{3.2}$$

where $\epsilon_1, \ldots, \epsilon_n$ are independent with $\epsilon_i \sim \text{N}(0, \sigma^2)$, and β are the p regression parameters.

Instead of the mean response being the linear predictor $x_i^\top \beta$, we may allow it to be a nonlinear function of parameters, that is

$$Y_i = \eta(x_i, \beta) + \epsilon_i\,, \tag{3.3}$$

where $\epsilon_1, \ldots, \epsilon_n$ are independent with $\epsilon_i \sim \text{N}(0, \sigma^2)$, and $\eta(x_i, \beta)$ is a nonlinear function of covariates and parameters β.

The linear model (3.2) is a special case of the model specified by (3.3) for $\eta(x, \beta) = x^\top \beta$.

Parameters in nonlinear models can be of two different types:

- *Physical parameters* that have particular meaning in the subject-area where the model comes from. Estimating the value of physical parameters, then, contributes to scientific understanding.

- *Tuning parameters* that do not necessarily have any physical meaning. Their presence is justified as a simplification of a more complex underlying system. The aim when estimating them is to make the model represent reality as well as possible.

The function $\eta(x, \beta)$ can be specified in two ways:

- *Mechanistically*: prior scientific knowledge is incorporated into building a mathematical model for the mean response. That model can often be complex and $\eta(x, \beta)$ may not be available in closed form.
- *Phenomenologically* (empirically): a function $\eta(x, \beta)$ may be posited that appears to capture the non-linear nature of the mean response.

Example 3.1 (Calcium uptake)**.** The `calcium` dataset in the `SMPracticals` R package provides data on the uptake of calcium (`cal`; in nmoles per mg) at set times (`time`; in minutes) by 27 cells in "hot" suspension. Figure 3.1 shows calcium uptake against time.

```
data("calcium", package = "SMPracticals")
plot(cal ~ time, data = calcium,
     xlab = "Time (minutes)",
     ylab = "Calcium uptake (nmoles/mg)",
     bg = "#ff7518", pch = 21)
```

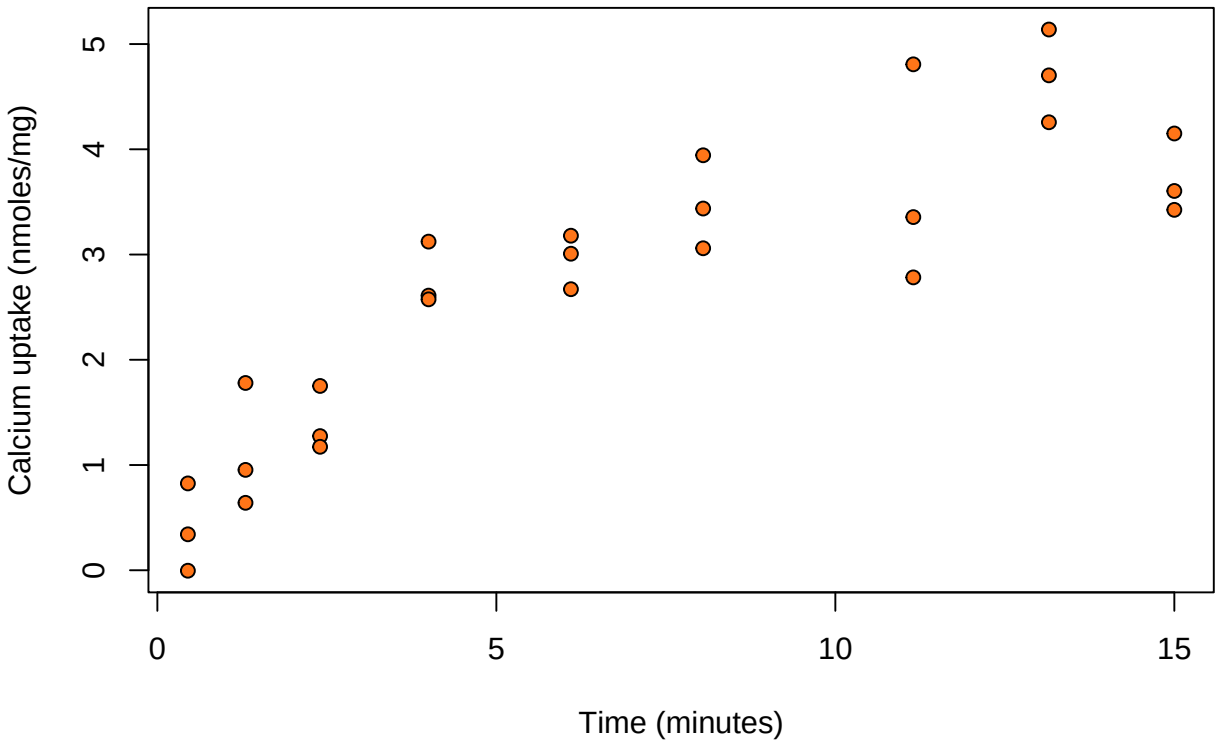


Figure 3.1: Calcium uptake against time.

We see that calcium uptake grows with time. There is a large class of phenomenological models for growth curves.

Consider the non-linear model with

$$\eta(x, \beta) = \beta_0 \left(1 - \exp\left(-x/\beta_1\right)\right) . \tag{3.4}$$

This is derived by assuming that the rate of growth is proportional to the calcium remaining, i.e.

$$\frac{d\eta}{dx} = (\beta_0 - \eta)/\beta_1 .$$

The solution to this differential equation, with initial condition $\eta(0, \beta) = 0$, is (3.4). Here, β_0 is the calcium uptake after infinite time, and β_1 controls its growth rate.

We use R to fit a model assumes a linear relationship of calcium uptake with time, a model that assumes a quadratic relationship, and the model specified by (3.4).

```
calc_lm1 <- lm(cal ~ time, data = calcium)
calc_lm2 <- lm(cal ~ time + I(time^2), data = calcium)
calc_nlm <- nls(cal~ beta0 * ( 1 - exp(-time/beta1)), data = calcium,
                start = list(beta0 = 5, beta1 = 5))
```

Figure 3.2 shows fitted curves for the three different models overlaid on the scatterplot of calcium uptake against time.

```
newdata <- data.frame(time = seq(min(calcium$time), max(calcium$time), length.out = 100))
pred_lm1 <- predict(calc_lm1, newdata = newdata)
pred_lm2 <- predict(calc_lm2, newdata = newdata)
pred_nlm <- predict(calc_nlm, newdata = newdata)
plot(cal ~ time, data = calcium,
     xlab = "Time (minutes)",
     ylab = "Calcium uptake (nmoles/mg)",
     bg = "#ff7518", pch = 21)
lines(newdata$time, pred_lm1, col = gray(0.8), lty = 1, lwd = 2)
lines(newdata$time, pred_lm2, col = gray(0.6), lty = 2, lwd = 2)
lines(newdata$time, pred_nlm, col = gray(0.4), lty = 3, lwd = 2)
legend("bottomright", legend = c("LM (linear)", "LM (quadratic)", "NLM"),
       col = gray(c(0.8, 0.6, 0.4)), lty = 1:3, lwd = 2)
```

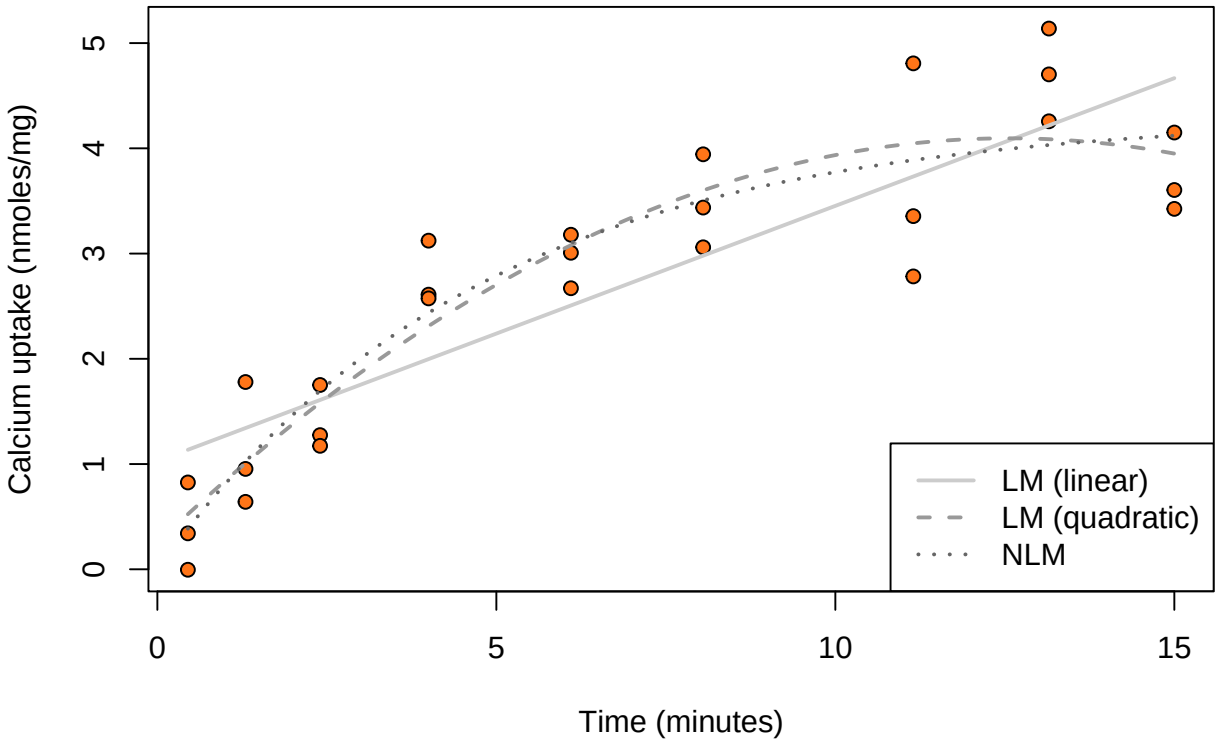


Figure 3.2: Calcium uptake against time, overlaid by estimated expected uptake from three models.

A comparison of the three models in terms of number of parameters, maximized log-likelihood value, and AIC and BIC returns

```
models <- list(`LM (linear)` = calc_lm1,
               `LM (quadratic)` = calc_lm2,
               `NLM` = calc_nlm)
out <- sapply(models, function(m) {
    c(p = length(coef(m)),
      loglik = logLik(m),
      AIC = AIC(m),
      BIC = BIC(m))
})
round(t(out), 3)
```

```
               p  loglik    AIC    BIC
LM (linear)    2 -28.701 63.403 67.290
LM (quadratic) 3 -20.955 49.910 55.093
NLM            2 -20.955 47.909 51.797
```

The maximized log-likelihoods from the quadratic and nonlinear model are identical up to 3 decimal places. The nonlinear model has is more parsimonious with fewer parameters. As a result it has lower AIC and BIC, and would be the preferred model.

3.2 Nonlinear mixed effects models

Example 3.2 (Theophylline data)**.** Theophylline is an anti-asthmatic drug. An experiment was performed on 12 individuals to investigate the way in which the drug leaves the body. The study of drug concentrations inside organisms is called *pharmacokinetics*.

An oral dose was given to each individual at time $t = 0$, and the concentration of theophylline in the blood was then measured at 11 time points in the next 25 hours.

Let Y_{ij} be the theophylline concentration (mg/L) for individual i at time t_{ij}, and D_i the dose that was administered.

Figure 3.3 shows the concentration of theophylline against time for each of the individuals. There is a sharp increase in concentration followed by a steady decrease.

```
data("Theoph", package = "datasets")
plot(conc ~ Time, data = Theoph, type = "n",
     ylab = "Concentration (mg/L)", xlab = "Time (hours)")
for (i in 1:30) {
    dat_i <- subset(Theoph, Subject == i)
    lines(conc ~ Time, data = dat_i, col = "grey")
}
points(conc ~ Time, data = Theoph,
       bg = "#ff7518", pch = 21, col = "grey")
```

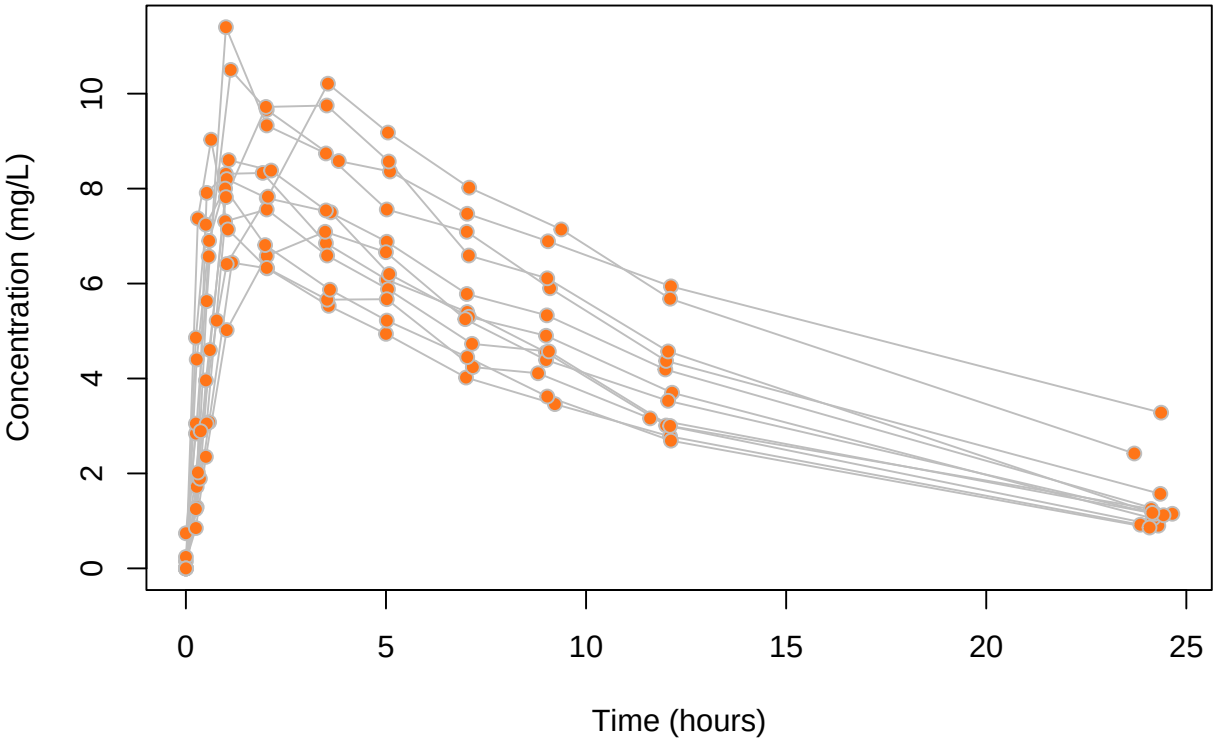


Figure 3.3: Concentration of theophylline against time for each of the individuals in the study.

Compartmental models are a common class of model used in pharmacokinetics studies. If the initial dosage is D, then a pharmacokinetic model with a first-order compartment function is

$$\eta(\beta, D, t) = \frac{D\beta_1\beta_2}{\beta_3(\beta_2 - \beta_1)} \left(\exp\left(-\beta_1 t\right) - \exp\left(-\beta_2 t\right)\right) , \tag{3.5}$$

where the parameters $\beta_1, \beta_2, \beta_3$ are all positive and have natural interpretations as follows:

- β_1: the elimination rate which controls the rate at which the drug leaves the organism;
- β_2: the absorption rate which controls the rate at which the drug enters the blood;
- β_3: the clearance which controls the volume of blood for which a drug is completely removed per time unit.

Since all the parameters are positive, and their estimation will most probably require a gradient descent step (e.g. what some of the methods in `optim` do), it is best to rewrite expression (3.5) in terms of $\gamma_i = log(\beta_i)$, which can take values on the whole real line. We can write

$$\eta'(\gamma, D, t) = \eta(\beta, D, t) = D\frac{\exp(-\exp(\gamma_1)t) - \exp(-\exp(\gamma_2)t)}{\exp(\gamma_3 - \gamma_1) - \exp(\gamma_3 - \gamma_2)} . \tag{3.6}$$

Let's initially ignore the dependence induced from repeated measurements on individuals and fit the nonlinear model

$$Y_{ij} = \eta'(\gamma, D_i, t_{ij}) + \epsilon_{ij} , \tag{3.7}$$

where $\epsilon_{ij} \overset{\text{ind}}{\sim} \mathrm{N}(0, \sigma^2)$.

```
fm <- conc ~ Dose *
    (exp(-exp(gamma1) * Time) - exp(-exp(gamma2) * Time)) /
    (exp(gamma3 - gamma1) - exp(gamma3 - gamma2))
(pkm <- nls(fm, start = list(gamma1 = 0, gamma2 = -1, gamma3 = -1),
            data = Theoph))
```

```
Nonlinear regression model
  model: conc ~ Dose * (exp(-exp(gamma1) * Time) - exp(-exp(gamma2) *     Time))/(exp(gamma3 - gamma
   data: Theoph
 gamma1  gamma2  gamma3
 0.3992 -2.5242 -3.2483
 residual sum-of-squares: 274.4

Number of iterations to convergence: 8
Achieved convergence tolerance: 3.709e-06
```

The estimates for β_1, β_2 and β_3 are

```
setNames(exp(coef(pkm)), paste0("beta", 1:3))
```

```
     beta1      beta2      beta3
1.49066465 0.08011951 0.03884169
```

and, using the delta method, the corresponding estimated standard errors are

```
setNames(exp(coef(pkm)) * coef(summary(pkm))[, "Std. Error"], paste0("beta", 1:3))
```

```
      beta1       beta2       beta3
0.175208160 0.008840925 0.002889612
```

Figure 3.4 shows boxplots of the residuals from fitting the model in (3.7), separately for each subject. We see evidence of unexplained differences between individuals.

```
res <- residuals(pkm, type = "pearson")
ord <- order(ave(res, Theoph$Subject))
subj <- Theoph$Subject[ord]
subj <- factor(subj, levels = unique(subj), ordered = TRUE)
plot(res[ord] ~ subj,
     xlab = "Subject (ordered by mean residual)",
     ylab = "Pearson residual",
     col = "#ff7518", pch = 21)
abline(h = 0, lty = 2)
```

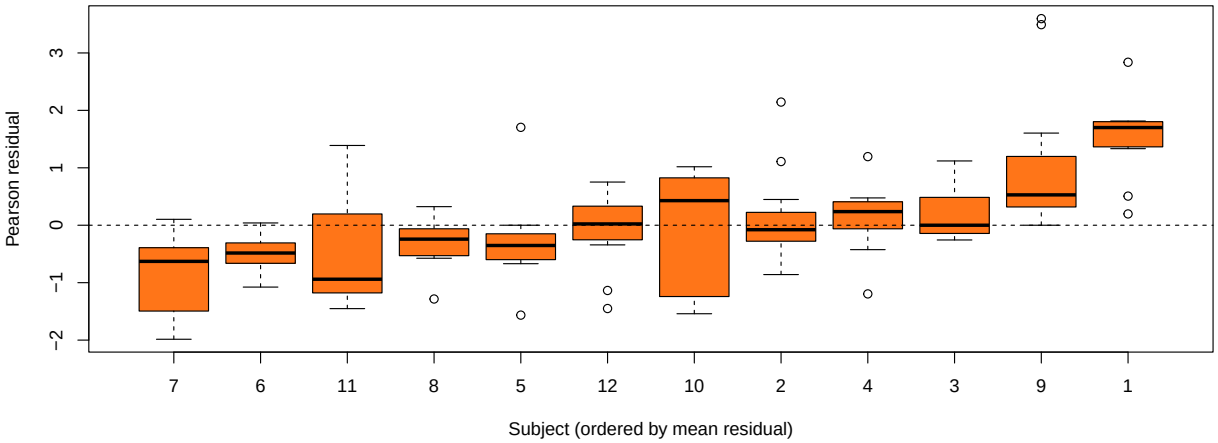


Figure 3.4: Residuals for each individual in the theopylline study from model (3.7).

Further evidence of unexplained differences is found in Figure 3.5, which shows the estimated curves from model (3.7) per individual. Accounting for heterogeneity between individuals seems worthwhile towards getting a better fit.

```
library("ggplot2")
st <- unique(Theoph[c("Subject", "Dose")])
pred_df <- as.list(rep(NA, nrow(st)))
for (i in seq.int(nrow(st))) {
    pred_df[[i]] <- data.frame(Time = seq(0, 25, by = 0.2),
                               Dose = st$Dose[i],
                               Subject = st$Subject[i])
}
pred_df <- do.call("rbind", pred_df)
pred_df$conc <- predict(pkm, newdata = pred_df)
## Order according to mean residual
theoph <- within(Theoph, Subject <- factor(Subject, levels = unique(subj), ordered = TRUE))
fig_theoph <- ggplot(theoph) +
    geom_point(aes(Time, conc), col = "#ff7518") +
    geom_hline(aes(yintercept = Dose), col = "grey", lty = 3) +
    facet_wrap(~ Subject, ncol = 3) +
    labs(y = "Concentration (mg/L)", x = "Time (hours)") +
    theme_bw()
fig_theoph +
    geom_line(data = pred_df, aes(Time, conc), col = "grey")
```

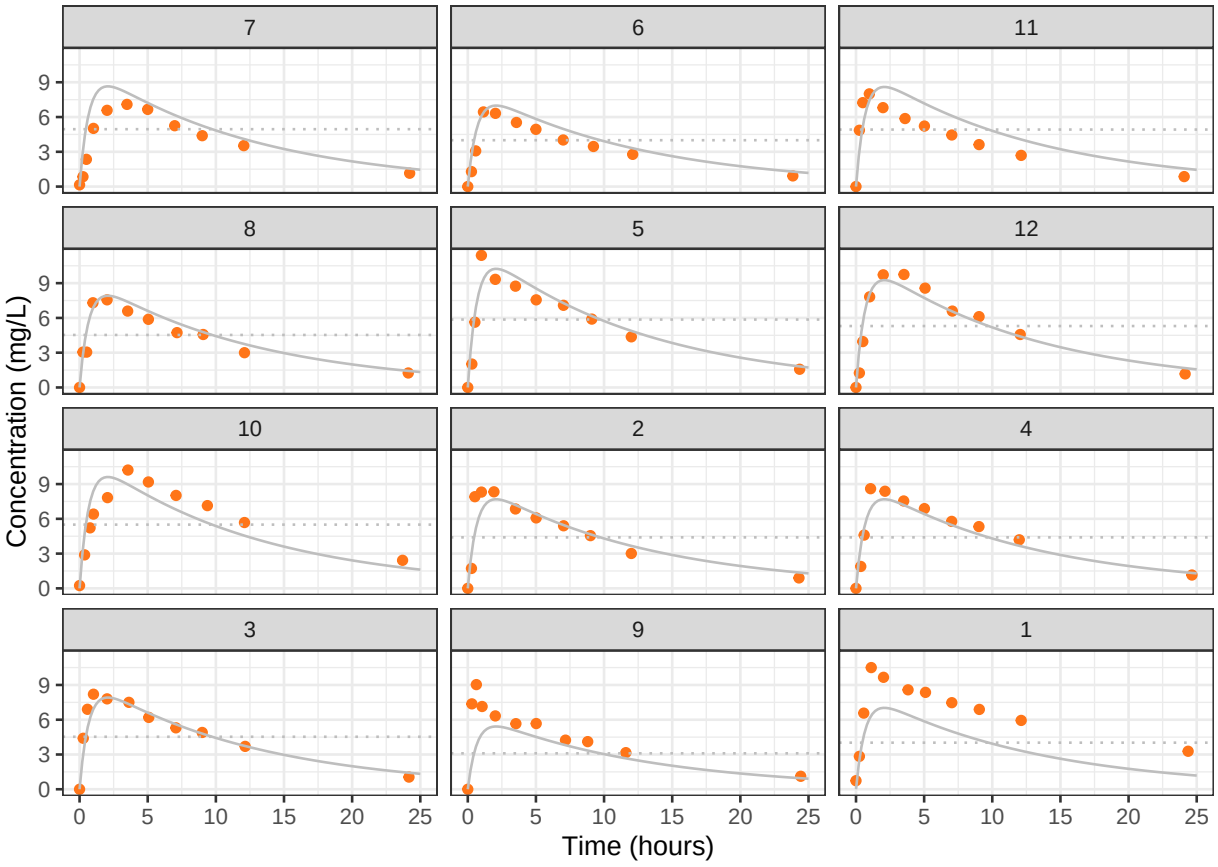


Figure 3.5: Estimated concentrations (grey) for each individual in the theopylline study from model (3.7). The dotted line is the administered dose.

A nonlinear mixed effects model that accounts for heterogeneity between clusters is

$$Y_{ij} = \eta(\beta + b_i, x_{ij}) + \epsilon_{ij},$$

where $\epsilon_{ij} \overset{\text{ind}}{\sim} \mathrm{N}(0, \sigma^2)$, $b_i \overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b)$, and Σ_b is a $q \times q$ covariance matrix.

The above model specifies that $\beta + b_i$ are parameters specific to the ith cluster with their relationship to the covariates being perhaps nonlinear.

For example, a mixed effects extension of model (3.7) for the Theophylline data, allows each individual to have distinct log-elimination rate, log-absorption rate and log-clearance, with joint distribution $\mathrm{N}(\gamma, \Sigma_b)$. The means γ of the cluster-specific parameters across all individuals are the population parameters. Note that adding normally distributed random variables to $\gamma_1, \gamma_2, \gamma_3$ (the logarithms of elimination rate, absorption rate and clearance) is less controversial than adding normally distributed random variables to $\beta_1, \beta_2, \beta_3$, which should be necessarily positive.

It is sometimes useful to specify the model in a way such that only a subset of the parameters can be different for each cluster, and the remainder fixed for all clusters. Suppose there are $q \leq p$ parameters that can be different for each cluster. Then, a more general way of writing the nonlinear mixed model is

$$Y_{ij} = \eta(\beta + Ab_i, x_{ij}) + \epsilon_{ij}\,, \tag{3.8}$$

where $\epsilon_{ij} \overset{\text{ind}}{\sim} \mathrm{N}(0, \sigma^2)$ and $b_i \overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b)$. Here Σ_b is a $q \times q$ covariance matrix and A is a $p \times q$ matrix of zeros and ones, which determines which parameters are fixed and which are varying.

The linear mixed model is a special case of the nonlinear mixed model with

$$\eta(\beta + Ab_i, x_{ij}) = x_{ij}^\top(\beta + Ab_i) = x_{ij}^\top\beta + x_{ij}^\top Ab_i = x_{ij}^\top\beta + z_{ij}^\top b_i\,.$$

If the first element of x_{ij} is 1 for all i and j, then a random intercept model results for $q = 1$ and $A = (1, 0, \ldots, 0)^\top$.

Example 3.3 (Theophylline data (revisited))**.** We use the `nlme` R package to fit a nonlinear mixed model that allows all the parameters to vary across individuals, i.e. $A = I_3$.

```
library("nlme")
pkmR <- nlme(fm,
             fixed = gamma1 + gamma2 + gamma3  ~ 1,
             random = gamma1 + gamma2 + gamma3 ~ 1,
             groups = ~ Subject,
             start = coef(pkm),
             control = lmeControl(msMaxIter = 500, maxIter = 500),
             data = Theoph)
pkmR
```

```
Nonlinear mixed-effects model fit by maximum likelihood
  Model: fm
  Data: Theoph
  Log-likelihood: -173.32
  Fixed: gamma1 + gamma2 + gamma3 ~ 1
    gamma1     gamma2     gamma3
 0.4514513 -2.4326850 -3.2144578

Random effects:
 Formula: list(gamma1 ~ 1, gamma2 ~ 1, gamma3 ~ 1)
 Level: Subject
 Structure: General positive-definite, Log-Cholesky parametrization
         StdDev    Corr
gamma1   0.6376932 gamma1 gamma2
gamma2   0.1310518  0.012
gamma3   0.2511873 -0.089  0.995
Residual 0.6818359

Number of Observations: 132
```

```
Number of Groups: 12
```

The estimates for the population parameters β are

```
setNames(exp(fixef(pkmR)), paste0("beta", 1:3))
```

```
     beta1      beta2      beta3
1.57058986 0.08780077 0.04017711
```

and, using the delta method, the corresponding estimated standard errors are

```
setNames(exp(fixef(pkmR)) * coef(summary(pkmR))[, "Std.Error"], paste0("beta", 1:3))
```

```
      beta1       beta2       beta3
0.308184354 0.005533699 0.003238213
```

Instead of reporting the estimate of Σ_b, the output provides estimates of the standard deviation of each random effect (`StdDev`), computed as the square roots of the diagonal elements of the estimate of Σ_b, and the correlation of the random effects (`Corr`). A first observation from the output is that the estimated variance of the random effect for the logarithm of the absorption rate (random effect with mean γ_2) is considerably smaller than those for the other two random effects, and that that random effect is is highly correlated to that with mean γ_3.

Let's examine the quality of the fit of a model that allows random effects with means γ_1 and γ_3, and just a population parameter for the logarithm of the absorption rate. Such a nonlinear mixed effects model has

$$Y_{ij} = \eta'\left(\begin{bmatrix}\gamma_1 + b_{i1}\\ \gamma_2\\ \gamma_3 + b_{i3}\end{bmatrix}, D_i, t_{ij}\right) + \epsilon_{ij}\,, \tag{3.9}$$

where $\epsilon_{ij} \overset{\text{ind}}{\sim} \mathrm{N}(0, \sigma^2)$, $(b_{i1}, b_{i3})^\top \overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b)$. This corresponds to (3.8) with

$$A = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix} \quad \text{and} \quad b_i = \begin{bmatrix}b_{i1}\\ b_{i3}\end{bmatrix}.$$

A comparison of model (3.9) to the model with all effects varying across individuals gives weak evidence against the former.

```
pkmR_2 <- update(pkmR, random = gamma1 + gamma3 ~ 1)
anova(pkmR, pkmR_2)
```

```
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
pkmR       1 10 366.6399 395.468 -173.3200
pkmR_2     2  7 368.0464 388.226 -177.0232 1 vs 2 7.406425    0.06
```

Indeed, the fit with γ_2 not varying has AIC 368.05 which is just higher than the AIC 366.64 of the full model, and BIC 388.23 which is smaller than the BIC 395.47 of the full model.

Figure 3.6 shows the estimated curves per subject from model (3.7) that ignores repeated measurements, and from the nonlinear mixed effects models with two and three effects varying across individuals. We can see that the nonlinear mixed effects models result in good fits. The estimated curved from the two mixed effects models are almost identical, apart from a slight deviation at the tail of the estimated curve from model (3.9) for individual 1.

```
conc_nlm <- pred_df$conc
conc_nlme_2 <- predict(pkmR_2, newdata = pred_df)
conc_nlme_3 <- predict(pkmR, newdata = pred_df)
pred_df_all <- pred_df[c("Subject", "Dose", "Time")]
pred_df_all <- rbind(
    data.frame(pred_df_all, conc = conc_nlm, model = "NLM"),
    data.frame(pred_df_all, conc = conc_nlme_2, model = "NLME(2)"),
```

```
    data.frame(pred_df_all, conc = conc_nlme_3, model = "NLME(3)"))
fig_theoph +
    geom_line(data = pred_df_all, aes(Time, conc, color = model))
```

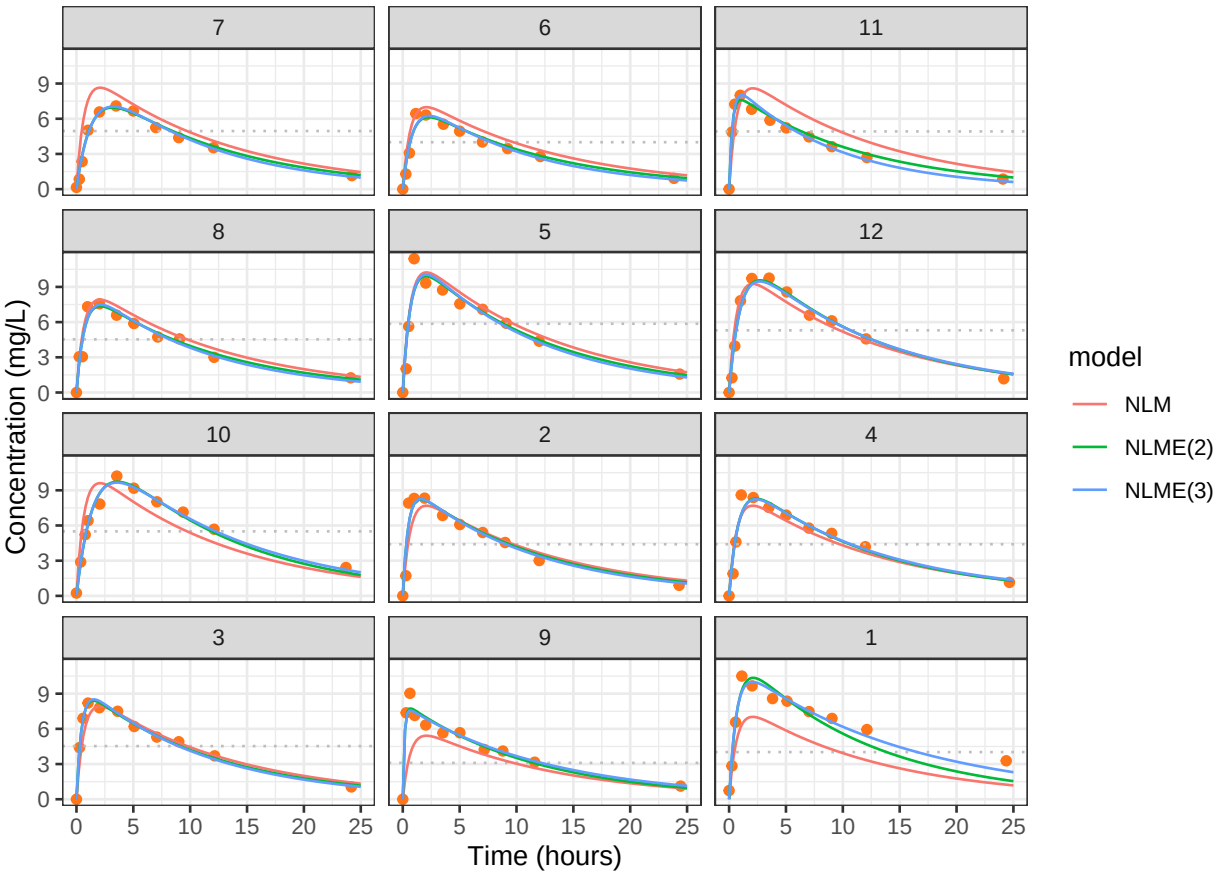


Figure 3.6: Estimated concentrations for each individual in the theopylline study from model (3.7; NLM), model (3.9; NLME(2)) and the model with all effects varying (NLM(3)). The dotted line is the administered dose.

3.3 Generalized nonlinear mixed effects models

Nonlinear models can be extended to non-normal responses in the same way as linear models. The generalized nonlinear mixed effects model (GNLMM) assumes

$$\begin{aligned} Y_i \mid x_i, b_i &\overset{\text{ind}}{\sim} \text{EF}(\mu_i, \sigma^2)\,, \\ \begin{bmatrix} g(\mu_1) \\ \vdots \\ g(\mu_n) \end{bmatrix} &= \eta(\beta + Ab_i, x_i)\,, \\ b_i &\overset{\text{ind}}{\sim} \text{N}(0, \Sigma_b)\,, \end{aligned} \tag{3.10}$$

where, again, $\text{EF}(\mu_i, \sigma^2)$ is an exponential family distribution with mean μ_i and variance $\sigma^2 V(\mu_i)/m_i$ for known m_i.

Model (3.10) has as special cases the linear model, the nonlinear model, the linear mixed effects model, the nonlinear mixed effects model, the generalized linear model, and the generalized nonlinear model.

There are various technical and practical issues related to fitting generalized nonlinear mixed effects models. For instance:

- as in GLMMs, the likelihood function is not available in closed form and needs to be approximated;
- oftentimes, general-purpose optimization routines do not converge to a global maximum of the likelihood;
- evaluating $\eta(\beta, x)$ can be computationally expensive in some applications, like, for example, when $\eta(\beta, x)$ is defined via a differential equation, which can only be solved numerically.

These are all areas of current research.

Chapter 4

Latent variables

4.1 Setting

Many statistical models simplify when written in terms of unobserved *latent variable* U in addition to the observed data Y. The latent variable

- may really exist: for example, when $Y = I(U > c)$ for some continuous U ("do you earn less than c per year?");
- may be human construct: for example, something called IQ is said to underlie scores on intelligence tests, but is IQ just a cultural construct? ("Mismeasure of man" debate, etc.);
- may just be a mathematical / computational device: for example, latent variables are used in the implementation of MCMC or EM algorithms.

Some prominent examples include models with random effects, hidden variables in probit regression, and mixture models.

Example 4.1 (Velocity of galaxies)**.** The `galaxies` data set of the `MASS` R packages provides the velocities, in km/sec, of 82 galaxies, moving away from our own galaxy, from 6 well-separated conic sections of an 'unfilled' survey of the Corona Borealis region. If galaxies are indeed super-clustered the distribution of their velocities estimated from the red-shift in their light-spectra would be multimodal, and unimodal otherwise.

Figure 4.1 shows two kernel density estimators with gaussian kernel but different bandwidth selection procedures. We can think of each observation having a latent, unobserved, variable indicating the super-cluster the galaxy belongs to. Clearly, depending on what density estimator is used, we may end up with different inferences.

```
cols <- hcl.colors(3)
data("galaxies", package = "MASS")
## Fix typo see `?galaxies`
galaxies[78] <- 26960
## Rescale to 1000km/s
galaxies <- galaxies / 1000
plot(x = c(0, 40), y = c(0, 0.25), type = "n", bty = "l",
     xlab = "velocity of galaxy (1000km/s)", ylab = "density")
rug(galaxies)
lines(density(galaxies, bw = "nrd0"), col = cols[2])
lines(density(galaxies, bw = "SJ"), col = cols[3])
legend("topleft", legend = c('bw = "nrd0"', 'bw = "SJ"'),
       col = cols[2:3], lty = 1)
```

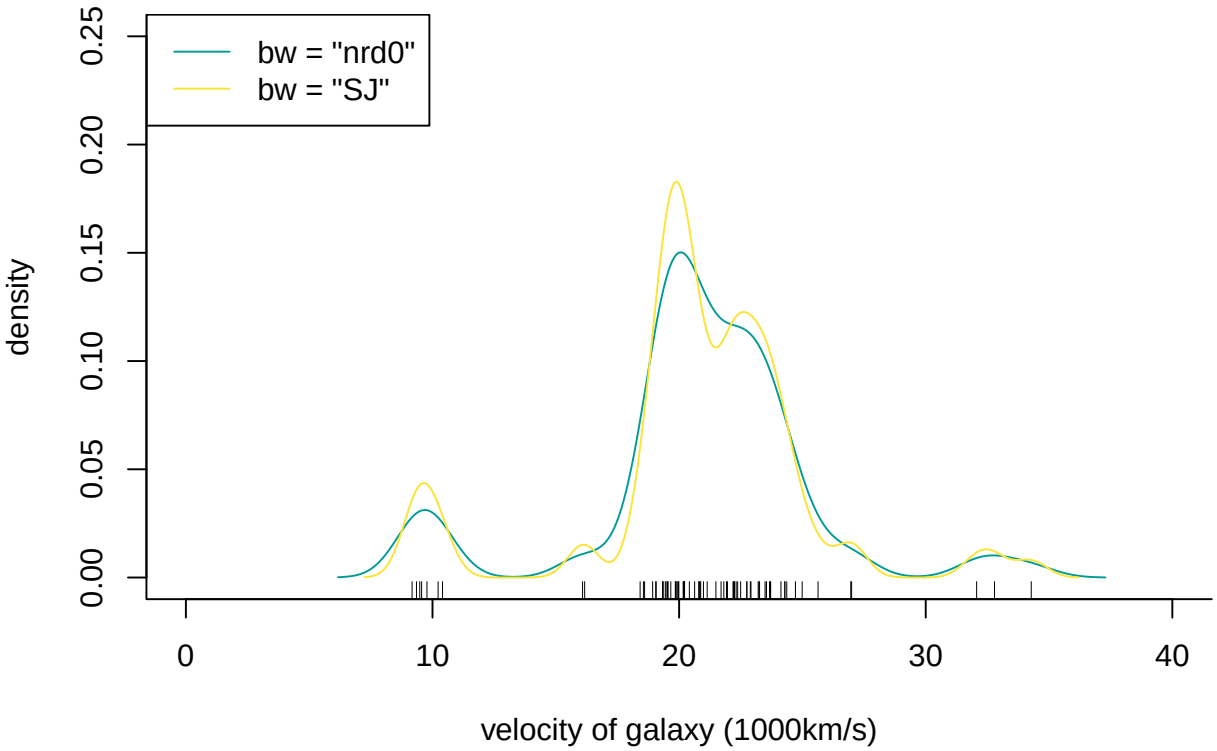


Figure 4.1: Density of galaxy velocities in 1000km/s, using two kernel density estimators with gaussian kernel but different bandwidth selection procedures.

4.2 Latent variable models

Let $[U], D$ denote discrete random variables, and $(U), X$ continuous ones.

Then in notation for graphical models:

- $[U] \to X$ or $[U] \to D$: finite mixture models, hidden Markov models, changepoint models, etc.
- $(U) \to D$: data coarsening (censoring, truncation, ...)
- $(U) \to X$ or $(U) \to D$: variance components and other hierarchical models (generalized nonlinear mixed models, exploratory / confirmatory factor analysis, etc.)

Example 4.2 (Probit regression)**.** For example, a generalized linear model for binary data with a probit link function can be written as

$$U_i \overset{\text{ind}}{\sim} \mathrm{N}(x_i^\top \beta, 1)$$
$$Y_i = I(U_i \geq 0)\,.$$

The log-likelihood of the probit regression model is then

$$\sum_{i=1}^{n} \{y_i \log \Phi(x_i^\top \beta) + (1 - y_i) \log\{1 - \Phi(x^\top \beta)\}\}\ .$$

Different assumptions for the distribution of the latent variable U give rise to different models (e.g. logistic distribution gives rise to logistic regression, extreme-value distribution to complementary log-log regression, etc.).

4.3 Finite mixture models

Settings like that of Example 4.1 are common in practice; observations are taken from a population composed of distinct sub-populations, but it is unknown from which of those sub-populations the observations

come from. A natural model for such settings is a finite mixture model. A p-component finite mixture model has density or probability mass function

$$f(y\,;\,\pi,\theta) = \sum_{r=1}^{p} \pi_r f_r(y\,;\,\theta) \quad (0 \le \pi_r \le 1; \sum_{r=1}^{p} \pi_r = 1)\,,$$

where π_r is the probability that y is from the rth component, and $f_r(y\,;\,\theta)$ is its density or probability mass function conditional on this event (*component density*).

We can represent the mixture model using indicator variables U, taking a value in $1, \dots, p$ with probabilities $\pi_1, \dots, \pi_p$, respectively, indicating from which component Y is drawn.

Mixture models is a widely used class of models for density estimation and clustering. It is often assumed that the number of components p is unknown.

Such models are non-regular for likelihood inference, for the following reasons:

- The ordering of components is non-identifiable. In other works, charging the order of the components does not change the model.
- Setting $\pi_r = 0$ eliminates the unknown parameters in $f_r(y\,;\,\pi,\theta)$
- Depending on the specification of the component distribution, the maximum of the log-likelihood can be $+\infty$, and can be achieved for various θ.

Typically, implementations use special constraints on the parameters overcome to the above issues.

4.4 Expectation-Maximization

4.4.1 Derivation of the EM algorithm

Suppose that the aim is to use the observed value y of Y for inference on θ when we cannot easily compute

$$f(y\,;\,\theta) = \int f(y \mid u\,;\,\theta) f(u\,;\,\theta) du\,.$$

Assuming that we have observations on U, the Bayes theorem gives that the *complete data log-likelihood* is

$$\log f(y, u\,;\,\theta) = \log f(y\,;\,\theta) + \log f(u \mid y\,;\,\theta)\,. \tag{4.1}$$

On the other hand, the *incomplete data log-likelihood* (sometimes also called the *observable data log-likelihood*) is simply

$$\ell(\theta) = \log f(y\,;\,\theta)\,.$$

Taking expectations in both sides of (4.1) with respect to $f(u \mid y\,;\,\theta')$ gives

$$\mathrm{E}\left(\log f(Y, U\,;\,\theta) \mid Y = y\,;\,\theta'\right) = \ell(\theta) + \mathrm{E}\left(\log f(U \mid Y\,;\,\theta) \mid Y = y\,;\,\theta'\right)\,, \tag{4.2}$$

which we write as

$$Q(\theta\,;\,\theta') = \ell(\theta) + C(\theta\,;\,\theta')\,.$$

Let's fix θ', and consider how $Q(\theta\,;\,\theta')$ and $C(\theta\,;\,\theta')$ depend on θ. Note that, by Jensen's inequality, $C(\theta'\,;\,\theta') \ge C(\theta\,;\,\theta')$, with equality only when $\theta = \theta'$. Hence,

$$Q(\theta\,;\,\theta') \ge Q(\theta'\,;\,\theta') \implies \ell(\theta) - \ell(\theta') \ge C(\theta'\,;\,\theta') - C(\theta\,;\,\theta') \ge 0\,. \tag{4.3}$$

Under mild smoothness conditions, $C(\theta\,;\,\theta')$ has a stationary point at $\theta := \theta'$, so if $Q(\theta\,;\,\theta')$ is stationary at $\theta := \theta'$, so too is $\ell(\theta)$.

Hence, we now formulate the *Expectation-Maximization* (EM) algorithm for maximizing $\ell(\theta)$. Starting from an initial value θ' of θ

1. Compute $Q(\theta\,;\,\theta') = \mathrm{E}\,(\log f(Y,U\,;\,\theta) \mid Y = y\,;\,\theta')$.
2. With θ' fixed, maximize $Q(\theta\,;\,\theta')$ with respect to θ, and let $\theta^\dagger$ be the maximizer.
3. Check if the algorithm has converged, based on $\ell(\theta^\dagger) - \ell(\theta')$ if available, or $\|\theta^\dagger - \theta'\|$, or both. If the algorithm has converged stop and return $\theta^\dagger$ as the value of the maximum likelihood estimator $\hat\theta$. Otherwise, set $\theta' := \theta^\dagger$ and go to 1.

Steps 1 and 2 are the Expectation (E) and maximization (M) steps, respectively.

The M-step ensures that $Q(\theta^\dagger\,;\,\theta') \geq Q(\theta'\,;\,\theta')$, so (4.3) implies that $\ell(\theta^\dagger) \geq \ell(\theta')$. The log likelihood never decreases between EM iterations!

4.4.2 Convergence

If $\ell(\theta)$ has only one stationary point, and if $Q(\theta\,;\,\theta')$ eventually reaches a stationary value at $\hat\theta$, then $\hat\theta$ must maximize $\ell(\theta)$. Otherwise, the algorithm may converge to a local maximum of the log likelihood or to a saddlepoint.

Note here, that the EM algorithm never decreases the log likelihood so it is, generally, more stable than Newton–Raphson-type algorithms. The rate of convergence depends on closeness of $Q(\theta\,;\,\theta')$ and $\ell(\theta)$.

Similarly to (4.2), we can write

$$-\frac{\partial^2 \ell(\theta)}{\partial\theta\partial\theta^\top} = \mathrm{E}\left(-\frac{\partial^2 \log f(Y,U\,;\,\theta)}{\partial\theta\partial\theta^\top}\middle| Y = y\,;\,\theta\right) - \mathrm{E}\left(-\frac{\partial^2 \log f(U \mid Y\,;\,\theta)}{\partial\theta\partial\theta^\top}\middle| Y = y\,;\,\theta\right),$$

or $J(\theta) = I_c(\theta\,;\,y) - I_m(\theta\,;\,y)$. The latter expression is often referred to as the *missing information principle*: the observed information equals the complete-data information minus the missing information.

The rate of convergence of the EM is slow if the largest eigenvalue of $I_c(\theta\,;\,y)^{-1} I_m(\theta\,;\,y) \approx 1$. Roughly, this occurs if the missing information is a high proportion of the total.

Example 4.3 (Negative binomial)**.** For a toy example, suppose that conditional on $U = u$, Y is a Poisson variable with mean u, and that U is gamma with mean θ and variance θ^2/ν. Inference is required for θ with the shape parameter $\nu > 0$ assumed known. The complete data log-likelihood is

$$y \log u - u - \log y! + \nu \log \nu - \nu \log \theta + (\nu - 1)\log u - \nu u/\theta - \log \Gamma(\nu)\,,$$

and hence (4.2) is

$$Q(\theta\,;\,\theta') = (y + \nu - 1)\,\mathrm{E}(\log U \mid Y = y\,;\,\theta') - (1 + \nu/\theta)\,\mathrm{E}(U \mid Y = y\,;\,\theta') - \nu \log \theta$$

plus terms that depend neither on U nor on θ.

The E-step, that is the computation of $Q(\theta\,;\,\theta')$, involves two expectations, but fortunately $\mathrm{E}(\log U \mid Y = y\,;\,\theta')$ does not appear in terms that involve θ and so is not required. To compute $\mathrm{E}(U \mid Y = y\,;\,\theta')$, note that Y and U have joint density

$$f(y \mid u) f(u\,;\,\theta) = \frac{u^y}{y!} e^{-u} \times \frac{\nu^\nu u^{\nu-1}}{\theta^\nu \Gamma(\nu)} e^{-\nu u/\theta}, \quad y = 0, 1, \ldots,\ u > 0, \quad \theta > 0\,,$$

so the marginal density of Y is

$$f(y\,;\,\theta) = \int_0^\infty f(y \mid u) f(u\,;\,\theta,\nu) du = \frac{\Gamma(y+\nu)\nu^\nu}{\Gamma(\nu) y!} \frac{\theta^y}{(\theta+\nu)^{y+\nu}} \quad (y = 0, 1, \ldots)\,.$$

Hence, the conditional density $f(u \mid y\,;\,\theta')$ is gamma with shape parameter $y + \nu$ and mean $\mathrm{E}(U \mid Y = y\,;\,\theta') = (y + \nu)/(1 + \nu/\theta')$. Ignoring terms that do not depend on θ, we can take

$$Q(\theta\,;\,\theta') = -(1 + \nu/\theta)(y + \nu)/(1 + \nu/\theta') - \nu \log \theta\,.$$

The M-step involves maximization of $Q(\theta\,;\,\theta')$ over θ for fixed θ'. If we differentiate with respect to θ we find that the maximizing value is

$$\theta^\dagger = \theta'(y+\nu)/(\theta'+\nu) \tag{4.4}$$

Hence, the EM algorithm boils down to choosing an initial θ', updating it to $\theta^\dagger$ using (4.4), setting $\theta' = \theta^\dagger$ and iterating to convergence.

4.5 EM for mixture models

Consider the p-component mixture density $f(y\,;\,\pi,\theta) = \sum_{r=1}^{p} \pi_r f_r(y\,;\,\theta)$. The contribution to the complete data log-likelihood (assuming that we know from what component y is from) has the form

$$\log f(y,u\,;\,\pi,\theta) = \sum_{r=1}^{p} I(u=r)\left\{\log \pi_r + \log f_r(y\,;\,\theta)\right\}\,.$$

Hence, for the E-step, we must compute the expectation of $\log f(y,u\,;\,\pi,\theta)$ over the conditional distribution

$$w_r(y\,;\,\pi',\theta') = P(U=r \mid Y=y\,;\,\pi',\theta') = \frac{\pi'_r f_r(y\,;\,\theta')}{\sum_{s=1}^{p} \pi'_s f_s(y\,;\,\theta')} \quad (r=1,\dots,p)\,. \tag{4.5}$$

This probability can be regarded as the weight attributable to component r if y has been observed.

The expected value of $I(U=r)$ with respect to (4.5) is $w_r(y\,;\,\pi',\theta')$, so the expected value of the complete-data log likelihood based on a random sample $(y_1,u_1),\dots,(y_n,u_n)$ is

$$\begin{aligned} Q(\pi,\theta\,;\,\pi',\theta') &= \sum_{j=1}^{n}\sum_{r=1}^{p} w_r(y_j\,;\,\pi',\theta')\left\{\log\pi_r + \log f_r(y_j\,;\,\theta)\right\} \\ &= \sum_{r=1}^{p}\left\{\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')\right\}\log\pi_r + \sum_{r=1}^{p}\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')\log f_r(y_j\,;\,\theta)\,. \end{aligned}$$

The M-step of the algorithm entails maximizing $Q(\pi,\theta\,;\,\pi',\theta')$ over π and θ for fixed π' and θ'. As π_r do not usually appear in the component density f_r, the maximizing values $\pi_r^\dagger$ are obtained from the first term of Q, which corresponds to a multinomial log likelihood. Thus

$$\pi_r^\dagger = \frac{1}{n}\sum_j w_r(y_j\,;\,\theta')$$

which is the average weight for component r.

Estimates of the parameters of the component distributions are obtained from the weighted log likelihoods that form the second term of $Q(\pi,\theta\,;\,\pi',\theta')$. For example, if f_r is the normal density with mean μ_r and variance σ_r^2, simple calculations give the weighted estimates

$$\mu_r^\dagger = \frac{\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')y_j}{\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')}\,, \quad \sigma_r^{2\dagger} = \frac{\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')(y_j-\mu_r^\dagger)^2}{\sum_{j=1}^{n} w_r(y_j\,;\,\pi',\theta')} \quad (r=1,\dots,p)\,.$$

Given initial values for the parameters, the EM algorithm simply involves computing the weights $w_r(y_j\,;\,\pi',\theta')$ at these initial values, updating to obtain $\pi^\dagger,\theta^\dagger$, and checking convergence using the log likelihood, $\|\theta^\dagger-\theta'\|+\|\pi^\dagger-\pi'\|$, or both. If convergence is not yet attained, π',θ' are replaced by $\pi^\dagger,\theta^\dagger$ and the cycle repeated.

Example 4.4 (Velocity of galaxies (revisited))**.** We now revisit Example 4.1, and fit a mixture model with normal component densities, where each component has its own mean and variance. This can be done using the `mclust` R package, which provides the EM algorithm for gaussian mixture models. The code chunk below fits all mixture models with 1 up to 10 components, and plots the values of BIC for the 10 models (note that mclust reports the negative of the BIC as we defined it). The model with 4 components has the best BIC value.

```
library("mclust")
gal_mix <- Mclust(galaxies, G = 1:10, modelNames = "V")
plot(gal_mix, what = "BIC")
```

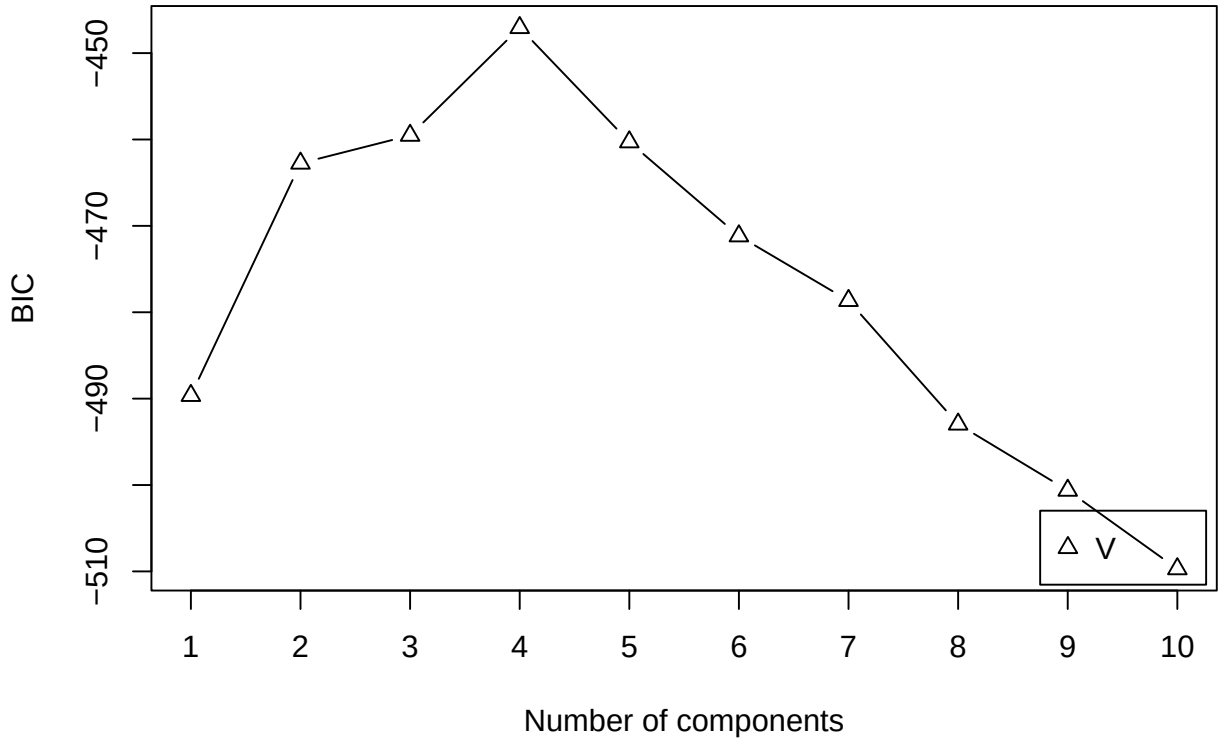


Figure 4.2 shows the estimated mixture density.

```
plot(x = c(0, 40), y = c(0, 0.25), type = "n", bty = "l",
     xlab = "velocity of galaxy (1000km/s)", ylab = "density")
rug(galaxies)
lines(density(galaxies, bw = "nrd0"), col = cols[2])
lines(density(galaxies, bw = "SJ"), col = cols[3])
legend("topleft", legend = c('GMM', 'bw = "nrd0"', 'bw = "SJ"'),
       col = cols, lty = 1)
ra <- seq(0, 40, length.out = 1000)
lines(ra, dens(ra, modelName = "V", parameters = gal_mix$parameters),
      col = cols[1])
```

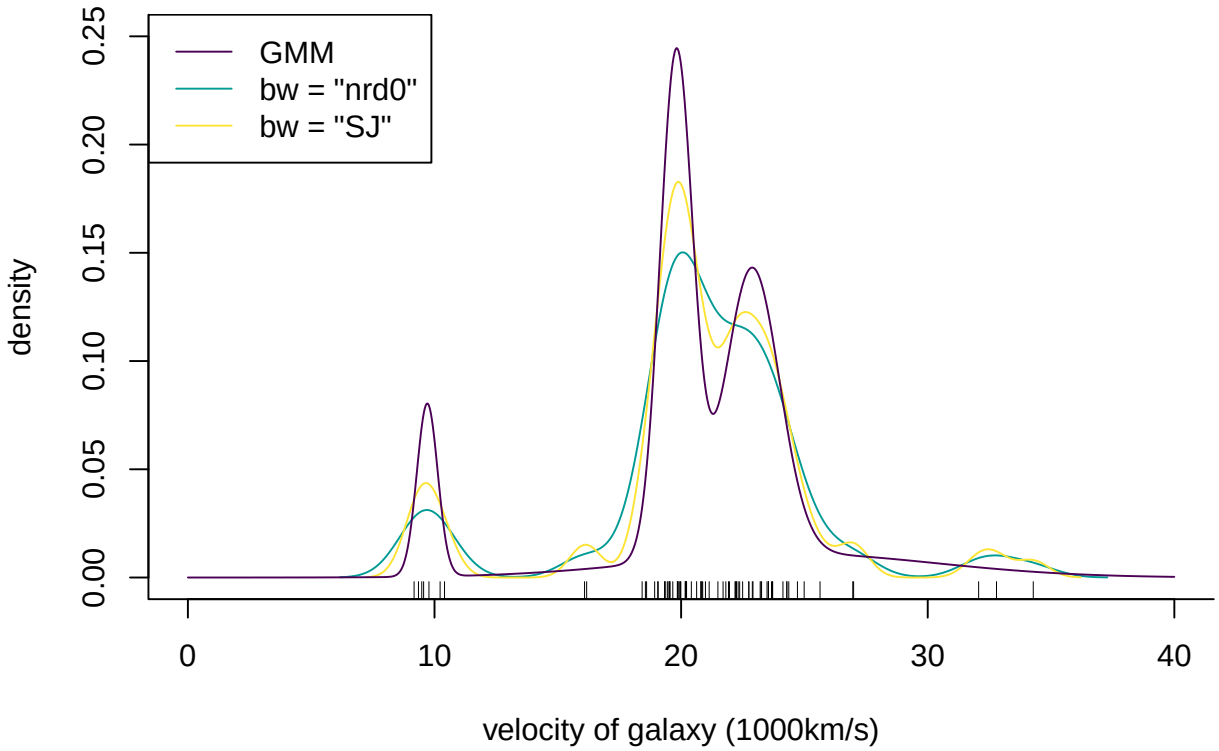
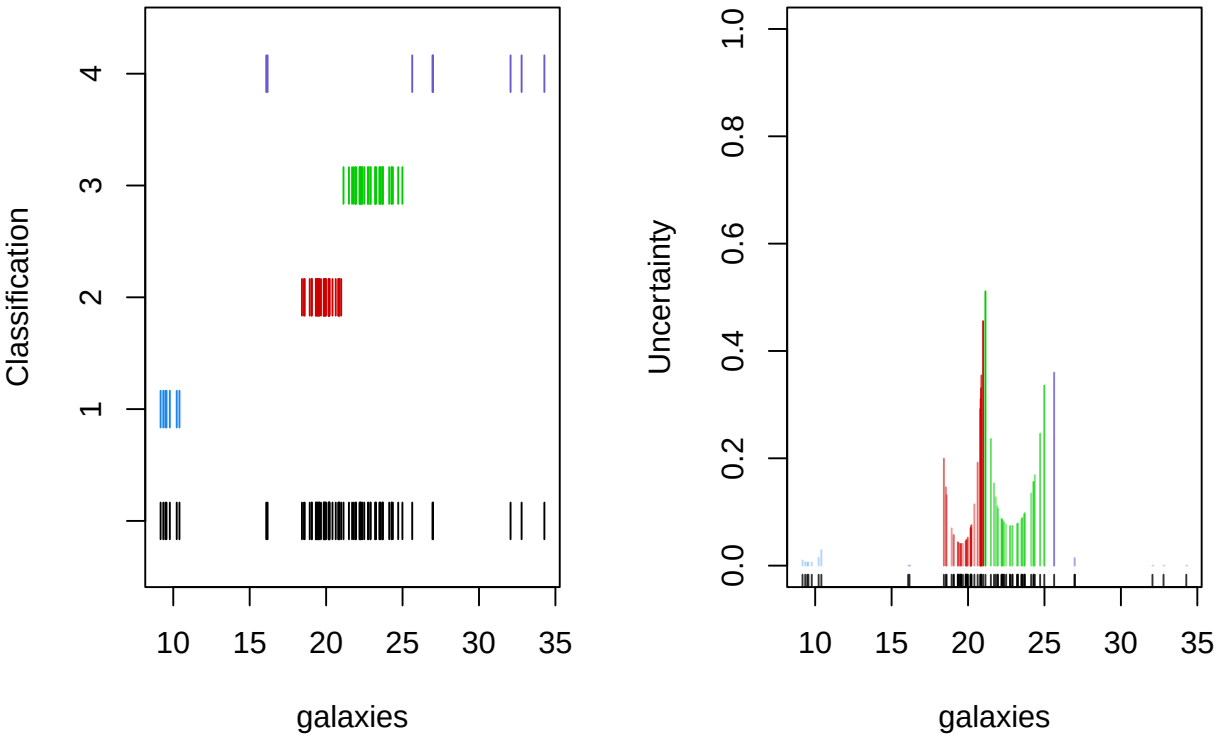


Figure 4.2: Density of galaxy velocities in 1000km/s, using two kernel density estimators with gaussian kernel but different bandwidth selection procedures, and a mixture of 4 normal densities.

One of the advantages of using the EM to fit mixture models is that the weights (4.5) from the E-step can be used to determine both at which component is observation has been assigned to and the uncertainty around that assignment. For example,

```
par(mfrow = c(1, 2))
plot(gal_mix, what = "classification")
plot(gal_mix, what = "uncertainty")
```



4.6 Exponential families

Suppose that the complete-data log likelihood is the logarithm of the density or probability mass function of an exponential family distribution, that is

$$\log f(y, u\,;\,\theta) = s(y, u)^\top \theta - \kappa(\theta) + c(y, u)\,. \tag{4.6}$$

In order to implement the EM algorithm, we need the expected value of $\log f(y, u\,;\,\theta)$ with respect to $f(u \mid y\,;\,\theta')$. The final term in (4.6) can be ignored. Hence, the M-step involves maximizing

$$Q(\theta\,;\,\theta') = \mathrm{E}\left(s(y, U)^\top \theta \mid Y = y\,;\,\theta'\right) - \kappa(\theta)\,,$$

or, equivalently, solving for θ the equation

$$\mathrm{E}\left(s(y, U) \mid Y = y\,;\,\theta'\right) = \frac{\partial \kappa(\theta)}{\partial \theta}\,.$$

The likelihood equation for θ based on the complete data is $s(y, u) = \partial\kappa(\theta)/\partial\theta$. So, the EM algorithm simply replaces $s(y, u)$ by its conditional expectation $\mathrm{E}\left(s(y, U) \mid Y = y\,;\,\theta'\right)$ and solves the likelihood equation. Thus, a routine to fit the complete-data model can readily be adapted for missing data if the conditional expectations are available.

Bibliography

This section provides a curated list of a few landmark papers and books on statistical modelling, covering topics in the current set of notes, and beyond (extra topics include missing data, EM algorithms, mixture models, learning under sparsity, causal inference, graphical models). Of course, given the breadth of the field and the speed at which it it developing, any list like the one below can hardly provide fair coverage, and will be missing many old and recent core developments. Its purpose it to be a helpful starting point for engaging more with key ideas and developments in statistical modelling.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Czáki (Eds.), *Second international symposium on information theory* (pp. 267–281). Akademiai Kiadó. https://doi.org/10.1007/978-1-4612-0919-5_38

Albert, J. H. (2007). *Bayesian computation with r*. Springer-Verlag. https://doi.org/10.1007/978-0-387-92298-0

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, *88*, 669–679. https://doi.org/10.2307/2290350

Best, N., & Thomas, A. (2000). Bayesian graphical models and software for GLMs. In D. K. Dey, S. K. Ghosh, & B. K. Mallick (Eds.), *Generalized linear models: A Bayesian perspective* (pp. 387–406). Marcel Dekker. https://doi.org/10.1201/9781482293456

Breslow, N. E., & Clayton, D. G. (1993). Appproximate inference in generalised linear mixed models. *Journal of the American Statistical Association*, *88*, 9–25. https://doi.org/10.2307/2290687

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multi-model inference: A practical information theoretic approach* (Second). Springer. https://doi.org/10.1007/978-1-4757-2917-7

Candès, E., & Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n (with discussion). *Annals of Statistics*, *35*, 2313–2404. https://doi.org/10.1214/009053606000001523

Claeskens, G., & Hjort, N. L. (2008). *Model selection and model averaging*. Cambridge University Press. https://doi.org/10.1017/CBO9780511790485

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhelter, D. J. (1999). *Probabilistic networks and expert systems*. Springer-Verlag. https://doi.org/10.1007/b97670

Crowder, M. J., & Hand, D. J. (1990). *Analysis of repeated measures*. Chapman; Hall/CRC. https://doi.org/10.1201/9781315137421

Davison, A. C. (2003). *Statistical models*. Cambridge University Press. https://doi.org/10.1017/CBO9780511815850

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society Series B*, *39*, 1–38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Diggle, P. J., Heagerty, P., Liang, K.-Y., & Zeger, S. (2002). *Analysis of longitudinal data* (2nd ed.). Oxford University Press. https://global.oup.com/academic/product/analysis-of-longitudinal-data-9780199676750

Draper, D. (1995). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society Series B*, *57*, 45–97. https://doi.org/10.1111/j.2517-6161.1995.tb02015.x

Efron, B. (1975). Defining the curvature of a statistical problem (with applications to second order efficiency). *The Annals of Statistics*, *3*(6), 1189–1242. https://doi.org/10.1214/aos/1176343282

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (Eds.). (2013). *Regression: Models, methods and applications*. Springer. https://doi.org/10.1007/978-3-642-34333-9

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties.

Journal of the American Statistical Association, *96*, 1348–1360.
Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space (with Discussion). *Journal of the Royal Statistical Society Series B*, *70*, 849–911. https://doi.org/10.1198/016214501753382273
Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2004). *Bayesian data analysis* (3rd ed.). Chapman; Hall/CRC. https://doi.org/10.1201/b16018
Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge University Press. https://doi.org/10.1017/9781139161879
Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain monte carlo in practice*. Chapman & Hall.
Green, P. J., Hjørt, N. L., & Richardson, S. (Eds.). (2003). *Highly structured stochastic systems*. Chapman & Hall/CRC. 10.1201/b14835
Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: The Lasso and generalizations*. Chapman; Hall/CRC. https://doi.org/10.1201/b18401
Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian model averaging: A tutorial (with discussion). *Statistical Science*, *14*, 382–417. https://doi.org/10.1214/ss/1009212519
Jamshidian, M., & Jennrich, R. I. (1997). Acceleration of the EM algorithm by using quasi-Newton methods. *Journal of the Royal Statistical Society Series B*, *59*, 569–587. https://doi.org/10.1111/1467-9868.00083
Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*, 773–795. https://doi.org/10.1080/01621459.1995.10476572
Liang, K., & Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, *73*(1), 13–22. https://doi.org/10.1093/biomet/73.1.13
Linhart, H., & Zucchini, W. (1986). *Model selection*. Wiley. https://doi.org/10.1007/978-3-642-04898-2_373
Little, R. J. A., & Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Wiley. https://doi.org/10.1002/9781119013563
Marin, J.-M., & Robert, C. P. (2007). *Bayesian core: A practical approach to computational bayesian statistics*. Springer-Verlag. https://doi.org/10.1007/978-0-387-38983-7
McCullagh, P. (2002). What is a statistical model? *The Annals of Statistics*, *30*(5), 1225–1310. https://doi.org/10.1214/aos/1035844977
McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). Chapman & Hall. https://doi.org/10.1201/9780203753736
McCulloch, C. E., Searle, S. R., & Neuhaus, J. M. (2008). *Generalized, linear, and mixed models* (2nd ed.). Wiley. https://doi.org/10.1002/0471722073
McLachlan, G. J., & Krishnan, T. (2008). *The EM algorithm and extensions* (2nd ed.). Wiley. https://doi.org/10.1002/9780470191613
McQuarrie, A. D. R., & Tsai, C.-L. (1998). *Regression and time series model selection*. World Scientific. https://doi.org/10.1142/3573
Meng, X.-L., & van Dyk, D. (1997). The EM algorithm — an old folk-song sung to a fast new tune (with discussion). *Journal of the Royal Statistical Society Series B*, *59*, 511–567. https://doi.org/10.1111/1467-9868.00082
Nelder, J. A., Lee, Y., & Pawitan, Y. (2017). *Generalized linear models with random effects: A unified approach via h-likelihood* (2nd ed.). Chapman; Hall/CRC. https://doi.org/10.1201/9781315119953
Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, *135*(3), 370–384. https://doi.org/10.2307/2344614
O'Hagan, A., & Forster, J. J. (2004). *Kendall's advanced theory of statistics. Volume 2B: Bayesian inference* (Second). Hodder Arnold. https://www.wiley.com/en-us/Kendall%27s+Advanced+Theory+of+Statistic+2B-p-9780470685693
Oakes, D. (1999). Direct calculation of the information matrix via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *61*, 479–482. https://doi.org/10.1111/1467-9868.00188
Ogden, H. (2021). On the error in laplace approximations of high-dimensional integrals. *Stat*, *10*(1), e380. https://doi.org/10.1002/sta4.380
Ogden, H. E. (2017). On asymptotic validity of naive inference with an approximate likelihood. *Biometrika*, *104*(1), 153–164. https://doi.org/10.1093/biomet/asx002
Peters, J., Janzing, D., & Schlkopf, B. (2017). *Elements of causal inference: Foundations and learning*

algorithms. The MIT Press. https://mitpress.mit.edu/books/elements-causal-inference

Pinheiro, J., & Bates, D. M. (2002). *Mixed effects models in S and S-PLUS*. New York:Springer-Verlag. https://doi.org/10.1007/b98882

Raftery, A. E., Madigan, D., & Hoeting, J. A. (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, *92*, 179–191. https://doi.org/10.1080/01621459.1997.10473615

Richardson, S., & Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society Series B*, *59*, 731–792. https://doi.org/10.1111/1467-9868.00095

Rissanen, J. (1987). Stochastic complexity (with discussion). *Journal of the Royal Statistical Society, Series B*, *49*, 223–239. https://doi.org/10.1111/j.2517-6161.1987.tb01694.x

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464. https://doi.org/10.1214/aos/1176344136

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. van der. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, *64*, 583–639. https://doi.org/10.1111/1467-9868.00353

Tanner, M. A. (1996). *Tools for statistical inference: Methods for the exploration of posterior distributions and likelihood functions* (Third). Springer. https://doi.org/10.1007/978-1-4612-4024-2

Venables, W., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed.). Springer-Verlag. https://doi.org/10.1007/978-0-387-21706-2

Wedderburn, R. W. M. (1974). Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. *Biometrika*, *61*(3), 439. https://doi.org/10.2307/2334725

Wood, S. N. (2017). *Generalized additive models: An introduction with r* (2nd ed.). Chapman; Hall/CRC. https://doi.org/10.1201/9781315370279

Chapter 5

Lab 1 (with solution)

5.1 Exercise

Suppose

$$Y_{im} \overset{\text{ind}}{\sim} \text{Poisson}\left(\mu^*(x_{im})\right) \quad (i = 1, \dots, n; m = 1, \dots, M),$$

where

$$\begin{aligned} \mu^*(x_{im}) &= 8 \exp\left(w(x_{im})\right), \\ x_{im} = x_i &= -10 + 20\frac{i-1}{n-1}, \\ w(x) &= 0.001\left(100 + x + x^2 + x^3\right). \end{aligned}$$

Consider the following simulation study. For $b = 1, \dots, B$:

- Generate
$$Y_{im} \overset{\text{ind}}{\sim} \text{Poisson}(\mu(x_{im})) \quad (i = 1, \dots, n; m = 1, \dots, M)$$
- Compute the AIC and BIC of the candidate models
$$Y_{im} \overset{\text{ind}}{\sim} \text{Poisson}(\mu(x_{im})), \quad \mu(x_{im}) = \exp\left(\sum_{j=1}^{p} \beta_j x_{im}^{j-1}\right),$$
for $p = 1, \dots, p_{\max}$.

You can carry out the simulation study for $n = 200$, $M = 3$, $p_{\max} = 20$, and $B = 100$ with the following code:

```
B <- 100
n <- 200
M <- 3
pmax <- 20


w <- function(x) {
    0.001 * (100 + x + x^2 + x^3)
}

mu <- function(x) {
    8 * exp(w(x))
}

## Covariates
```

```
x <- rep(seq(from = -10, to = 10, length = n), each = M)

## Objects to hold AICs and BICs
aics <- bics <- matrix(NA, nrow = B, ncol = pmax)

set.seed(20240416)
for (b in 1:B) {
    ## Simulate responses
    y <- rpois(n = M * n, lambda = mu(x))
    ## Fit intercept-only model and compute AIC, BIC
    mod <- glm(y ~ 1, family = poisson)
    aics[b, 1] <- AIC(mod)
    bics[b, 1] <- BIC(mod)
    ## Fit remaining models and compute AIC, BIC
    for(p in 2:pmax) {
        modp <- glm(y ~ poly(x, p - 1), family = poisson)
        aics[b, p] <- AIC(modp)
        bics[b, p] <- BIC(modp)
    }
}
```

The number of times each p has been selected by AIC and BIC is

```
AICorder <- apply(aics, 1, which.min)
BICorder <- apply(bics, 1, which.min)
table(AICorder)
```

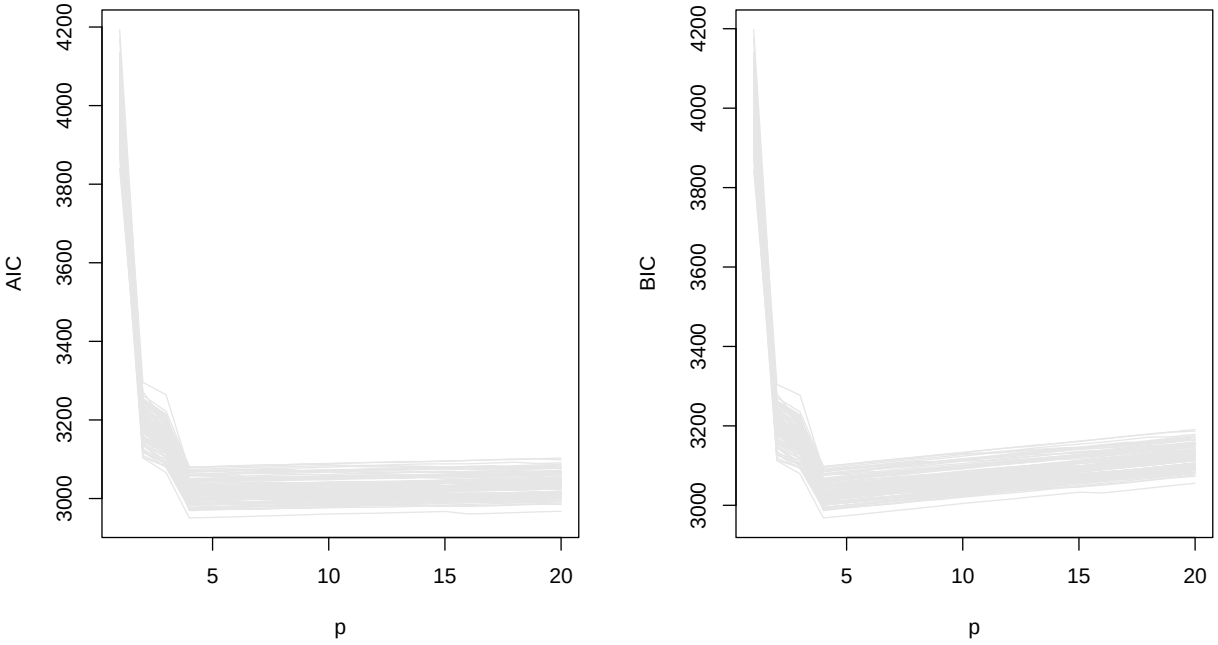
```
AICorder
 4  5  6  7  8  9 10 13
78 10  3  5  1  1  1  1
```

```
table(BICorder)
```

```
BICorder
 4  5
98  2
```

The AIC and BIC values for each sample as a function of p are

```
par(mfrow = c(1, 2))
matplot(x = 1:pmax, t(aics), xlab = "p", ylab = "AIC", type = "l", lty = 1, col = gray(0.9))
matplot(x = 1:pmax, t(bics), xlab = "p", ylab = "BIC", type = "l", lty = 1, col = gray(0.9))
```



1. Modify the code above to investigate the performance of AIC and BIC as model selection criteria for $n \in \{25, 50, 100, 1000\}$.

   [Suggestion: write a function that carries out the simulation for user-supplied n, M, $p_{\max}$, B, and the function $w(\cdot)$.]

2. Use
$$w(x) = \frac{1.2}{1 + \exp(-x)},$$
in the model we simulate from. How do AIC and BIC perform when the candidate models do not include the simulation model?

3. What information criterion would you use to estimate Δ in (1.2) and why? Use the simulation results to obtain a simulation-based estimate of Δ as a function of p, for $n \in \{25, 50, 100, 1000\}$, and each choice of $w(\cdot)$.

   Then, estimate Δ directly by its definition using out-of-sample log-likelihood.

   [Hint: for the out-of-sample log-likelihood you can do `sum(dpois(y_plus, mu_hat, log = TRUE))`, where `mu_hat` are the fitted means from a training sample and `y_plus` is an independent sample of the same size and at exactly the same x values as the training sample.]

5.2 Solution

Since we are planning to explore various experimental conditions for the simulation study, it is a good idea to write a general function that allows us to vary n, M, $p_{\max}$, B, and the function $w(.)$.

```
run_simulation <- function(n, M = 3, pmax = 20, B = 100,
                           w =  function(x) 0.001 * (100 + x + x^2 + x^3)) {
    mu <- function(x) {
        8 * exp(w(x))
    }
    x <- rep(seq(from = -10, to = 10, length = n), each = M)
    ## Objects to hold AICs and BICs
    aics <- bics <- matrix(NA, nrow = B, ncol = pmax)
    for (b in 1:B) {
        ## Simulate responses
        y <- rpois(n = M * n, lambda = mu(x))
```

```
        ## Fit intercept-only model and compute AIC, BIC
        mod <- glm(y ~ 1, family = poisson)
        aics[b, 1] <- AIC(mod)
        bics[b, 1] <- BIC(mod)
        ## Fit remaining models and compute AIC, BIC
        for(p in 2:pmax) {
            modp <- glm(y ~ poly(x, p - 1), family = poisson)
            aics[b, p] <- AIC(modp)
            bics[b, p] <- BIC(modp)
        }
    }
    list(AIC = aics, BIC = bics)
}
```

We can also write functions that take as input the output of `run_simulation()` and summarize the results, as is done in the code provided.

```
## Returns the number of times each $p$ has been selected by the
## criterion used to compute ics
get_selection_counts <- function(ics) {
    ic_order <- apply(ics, 1, which.min)
    table(ic_order)
}
## Plots the information criterion values in ics for each sample as a
## function of $p$ are
plot_ic <- function(ics, ylab = "IC", main = NULL) {
    matplot(x = 1:pmax, t(ics), xlab = "p", ylab = ylab, main = main, type = "l", lty = 1, col = gray
}
```

So, for the simulation study with for $n = 200$, $M = 3$, $p_{\max} = 20$, and $B = 100$, we can now simply do (using the same seed)

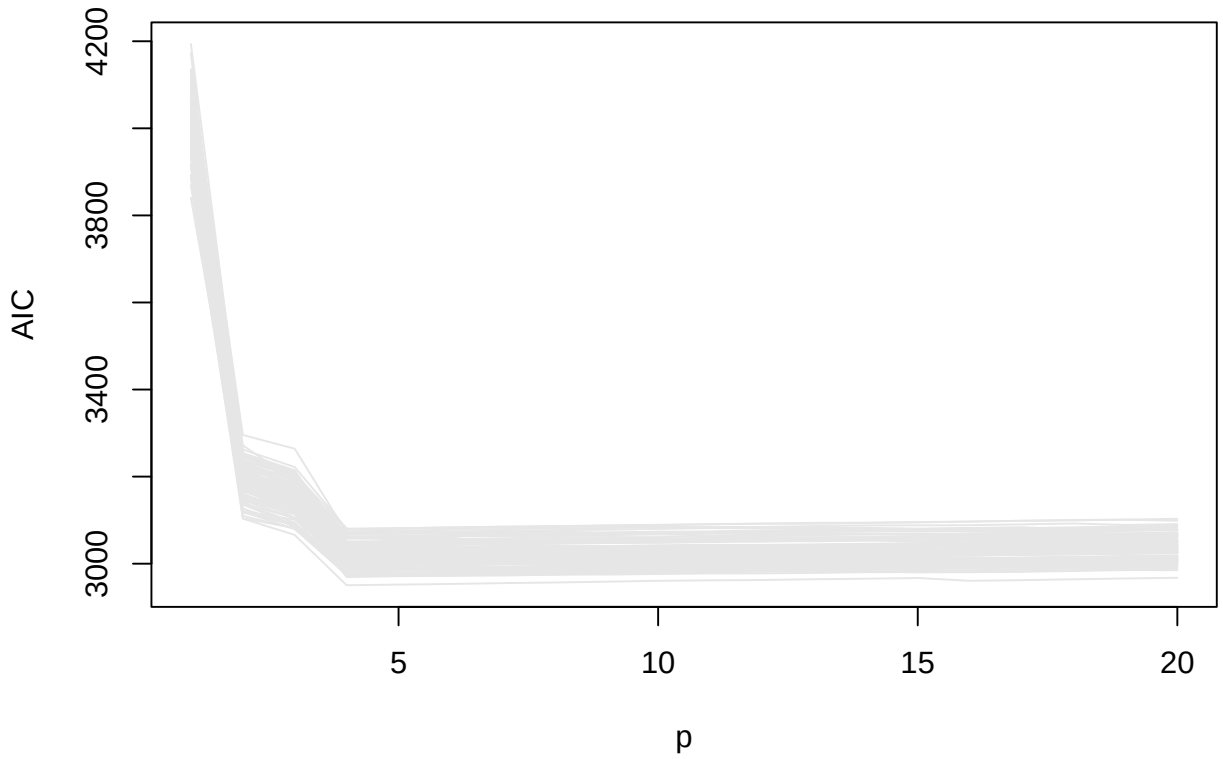
```
set.seed(20240416)

res <- run_simulation(n = 200, M = 3, pmax = 20, B = 100, w = function(x) 0.001 * (100 + x + x^2 + x
```

Then, for example, the results for AIC are

```
get_selection_counts(res$AIC)
```

```
ic_order
 4  5  6  7  8  9 10 13
78 10  3  5  1  1  1  1
```

```
plot_ic(res$AIC, ylab = "AIC")
```



1.

The code chunk below will run the required simulations and store the results in the list `res_true`.

```
set.seed(1)

ns <- c(25, 50, 100, 1000)
res_true <- as.list(numeric(length(ns)))
for (j in 1:length(ns)) {
    res_true[[j]] <- run_simulation(n = ns[j], M = 3, pmax = 20, B = 100,
                                    w = function(x) 0.001 * (100 + x + x^2 + x^3))
}
names(res_true) <- paste("n =", ns)
```

Let's investigate the behaviour of AIC.

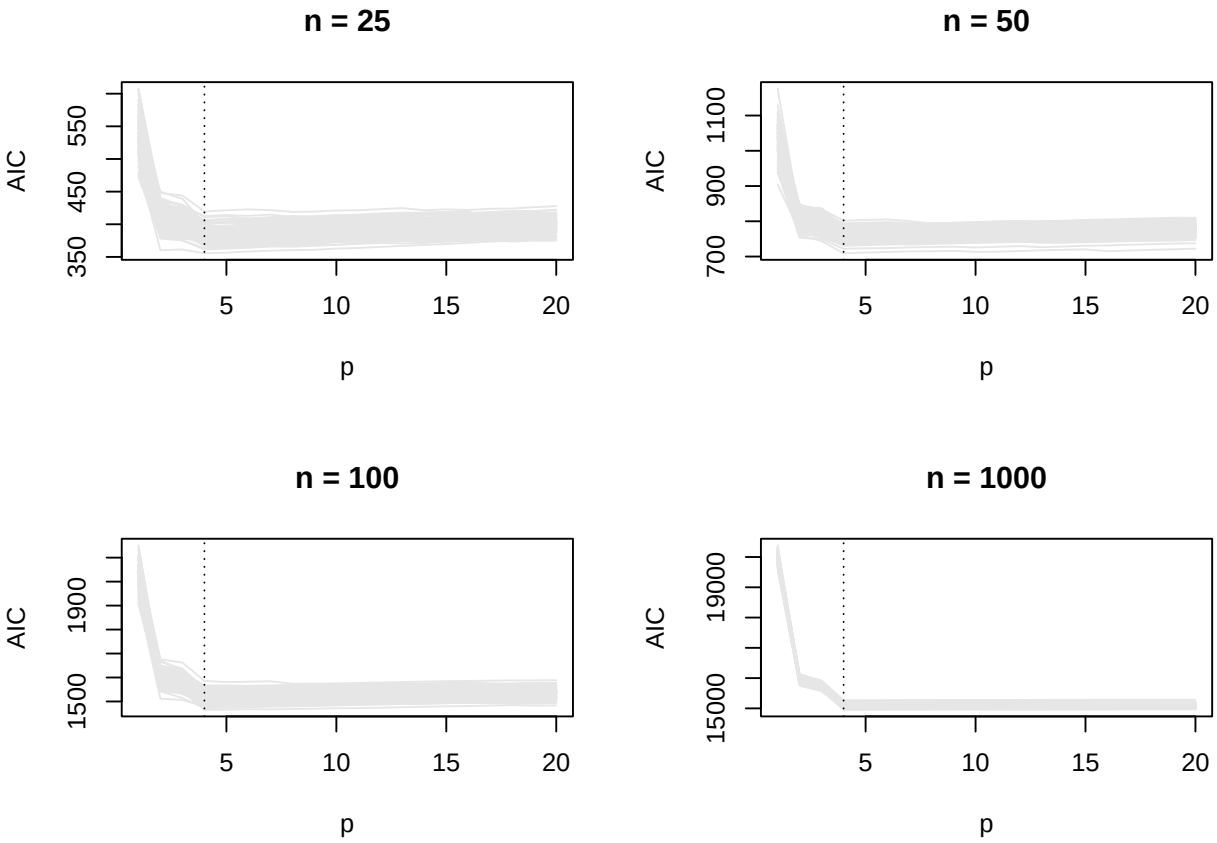
```
sapply(res_true, function(x) get_selection_counts(x$AIC))
```

```
$`n = 25`
ic_order
 4  5  6  7  8  9 10 18
67 11 11  2  5  1  1  2

$`n = 50`
ic_order
 4  5  6  7  8  9 11 13
70  7 10  5  3  1  3  1

$`n = 100`
ic_order
 4  5  6  7  8  9 18
72 11  8  5  2  1  1

$`n = 1000`
```

```
ic_order
 4  5  6  7  8  9 10 11
69 11 10  4  2  2  1  1
```

AIC behaves similarly for all n. In all cases, the correct (cubic) model is preferred most of the time, but the probability of it being selected does not tend to one as n grows. This is also apparent in the the AIC vs p plots

```
par(mfrow = c(2, 2))
for (j in 1:length(ns)) {
    plot_ic(res_true[[j]]$AIC, ylab = "AIC", main = paste("n =", ns[j]))
    abline(v = 4, lty = 3)
}
```



Let's explore the behaviour of BIC.

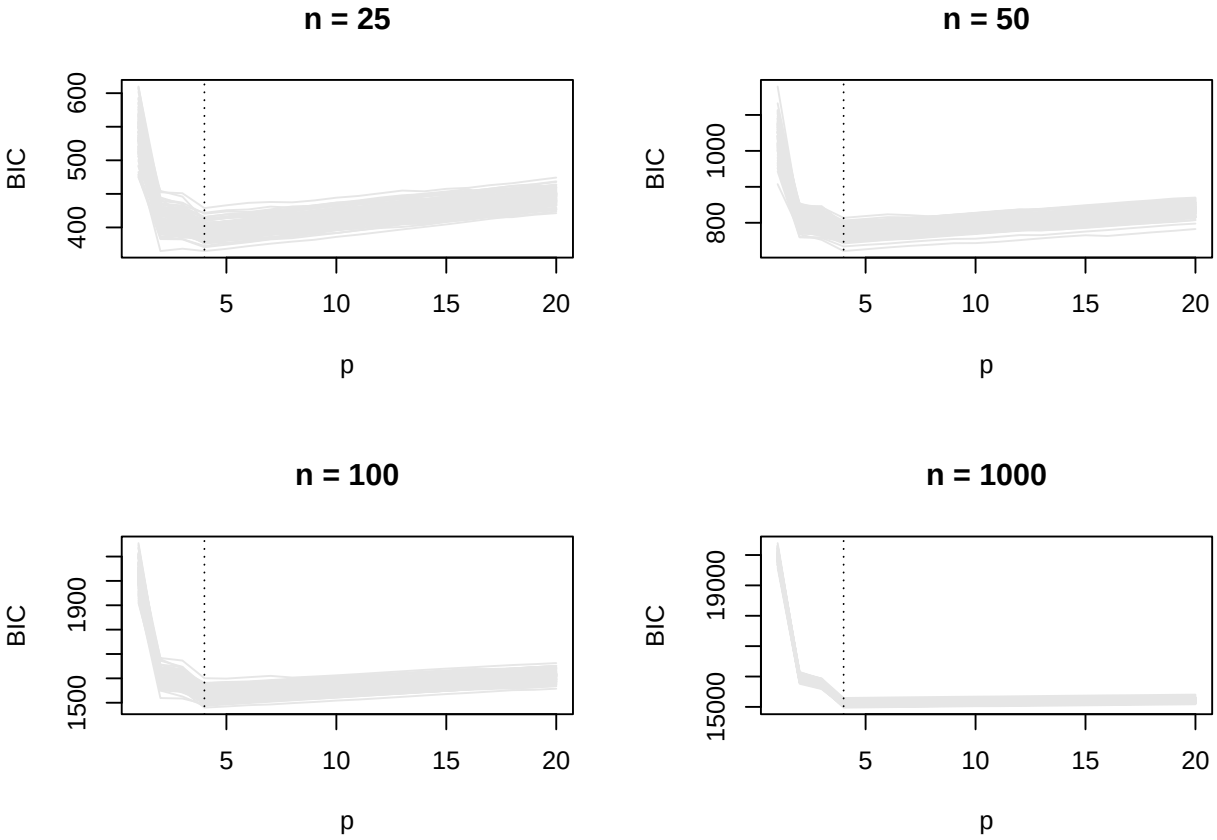
```
sapply(res_true, function(x) get_selection_counts(x$BIC))
```

```
$`n = 25`
ic_order
 2  3  4  5  6
 1  1 93  4  1

$`n = 50`
ic_order
 4  5
99  1

$`n = 100`
ic_order
 4  5
```

```
97  3

$`n = 1000`
ic_order
  4
100
```

In this example, the set of candidate models includes the model we simulate from, and BIC selects the correct model more and more often as n increases. This is also apparent in the BIC vs p plots.

```
par(mfrow = c(2, 2))
for (j in 1:length(ns)) {
    plot_ic(res_true[[j]]$BIC, ylab = "BIC", main = paste("n =", ns[j]))
    abline(v = 4, lty = 3)
}
```



2. The code chunk below will run the required simulations and store the results in the list `res_other`.

```
set.seed(1)

ns <- c(25, 50, 100, 1000)
res_other <- as.list(numeric(length(ns)))
for (j in 1:length(ns)) {
    res_other[[j]] <- run_simulation(n = ns[j], M = 3, pmax = 20, B = 100,
                                     w = function(x) 1.2 / (1 + exp( - x)))
}
names(res_other) <- paste("n =", ns)
```

Let's investigate the behaviour of AIC.

```
sapply(res_other, function(x) get_selection_counts(x$AIC))
```

```
$`n = 25`
ic_order
 4  5  6  7  8  9 10 11 12 13 14 15 16 20
 9  2 33  8 19  5  7  3  4  2  2  4  1  1

$`n = 50`
ic_order
 4  6  7  8  9 10 11 12 13 14 15 16 17 18 20
 1 31  5 22 10  8  5  4  1  3  3  3  1  2  1

$`n = 100`
ic_order
 6  7  8  9 10 11 12 13 16 17 18
15  3 33 10 28  2  3  1  1  2  2

$`n = 1000`
ic_order
10 11 12 13 14 15 16 17 18 20
13  5 33  6 21  5  8  1  7  1
```

As n increases, AIC tends to select increasingly complex models, which provide a better approximation to the true distribution which generated the data, which is not a polynomial model.

On the other hand, BIC prefers simpler models to AIC, although it still tends to prefer more complex models as n increases in this case.

```
sapply(res_other, function(x) get_selection_counts(x$BIC))
```

```
$`n = 25`
ic_order
 4  5  6  7  8 10 13
41  4 43  5  5  1  1

$`n = 50`
ic_order
 4  5  6  7  8  9 10
18  2 59  9 10  1  1

$`n = 100`
ic_order
 6  7  8 10
55  6 35  4

$`n = 1000`
ic_order
 8 10 11 12 13
13 76  3  7  1
```

3. We have shown that

$$\bar{\ell}(\hat{\theta}) - p/(nM) = -\frac{1}{2nM}\{2(p - \ell(\hat{\theta}))\} = -\frac{1}{2nM}AIC$$

is a bias-corrected estimator of Δ.

So,

$$\tilde{\Delta} = -\frac{1}{2nM}\,\mathrm{E}(AIC)\,,$$

should be approaching Δ as n increases. We can estimate $\tilde{\Delta}$ empirically by transforming the AIC values we obtained in the simulations, and taking averages for every p.

A direct, simulation-based estimator of Δ can be obtained by its definition, using independent samples from the ones that are used for estimation. The code chunk below provides a function that returns a simulation-base estimator of Δ as a function of p.

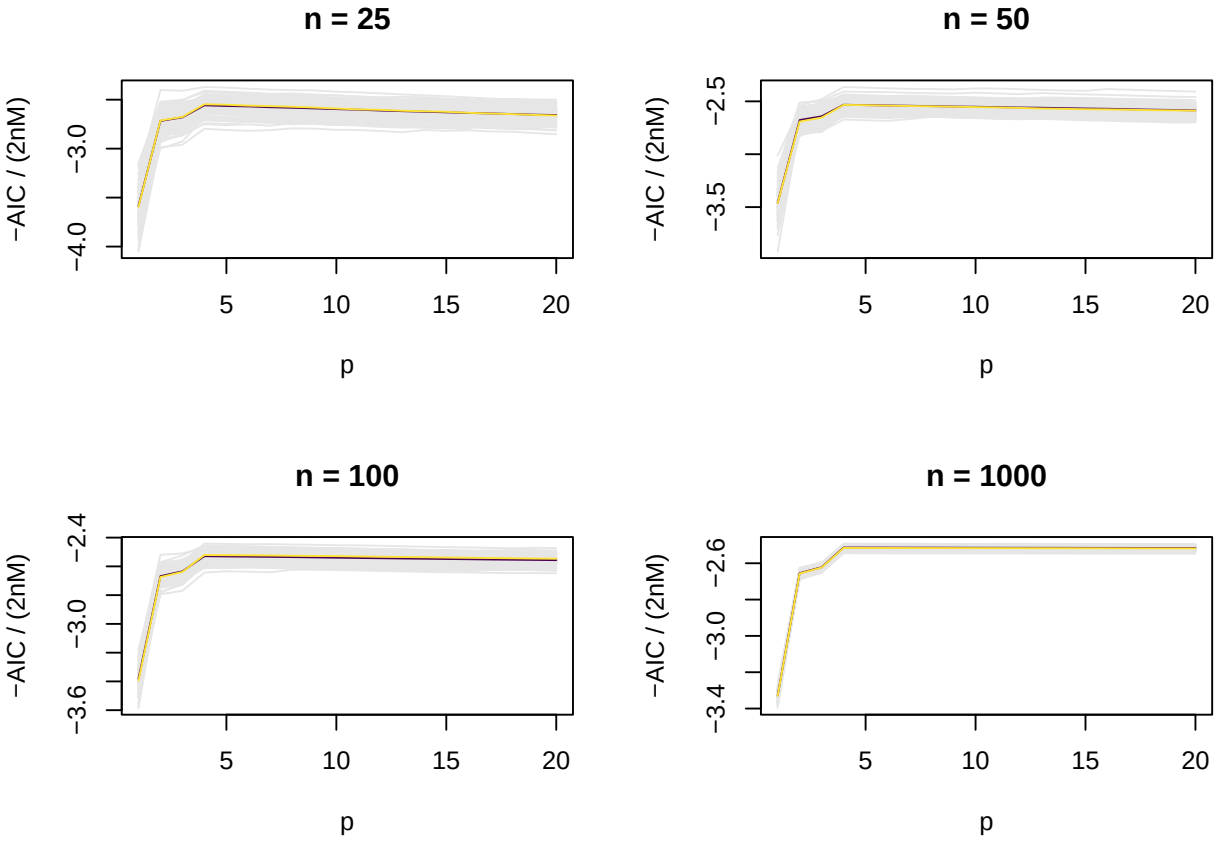
```
estimate_Delta <- function(n, M = 3, pmax = 20, B = 100,
                           w =  function(x) 0.001 * (100 + x + x^2 + x^3)) {
    mu <- function(x) {
        8 * exp(w(x))
    }
    x <- rep(seq(from = -10, to = 10, length = n), each = M)
    ## Object to hold out-of-sample log-likelihood
    oo_ll <- matrix(NA, nrow = B, ncol = pmax)
    for (b in 1:B) {
        ## Simulate responses
        y <- rpois(n = M * n, lambda = mu(x))
        y_plus <- rpois(n = M * n, lambda = mu(x))
        ## Fit intercept-only model and compute out-of-sample log-likelihood
        mod <- glm(y ~ 1, family = poisson)
        oo_ll[b, 1] <- sum(dpois(y_plus, fitted(mod), log = TRUE))
        ## Fit remaining models and compute out-of-sample log-likelihood
        for(p in 2:pmax) {
            modp <- glm(y ~ poly(x, p - 1), family = poisson)
            oo_ll[b, p] <- sum(dpois(y_plus, fitted(modp), log = TRUE))
        }
    }
    colMeans(oo_ll / (n * M))
}
```

The estimates of Δ for $w(x) = 0.001\,(100 + x + x^2 + x^3)$ and $w(x) = 1.2/\{1 + \exp(-x)\}$ are

```
set.seed(1)

ns <- c(25, 50, 100, 1000)
Delta_true <- Delta_other <- as.list(numeric(length(ns)))
for (j in 1:length(ns)) {
    Delta_true[[j]] <- estimate_Delta(n = ns[j], M = 3, pmax = 20, B = 100,
                                      w = function(x) 0.001 * (100 + x + x^2 + x^3))
    Delta_other[[j]] <- estimate_Delta(n = ns[j], M = 3, pmax = 20, B = 100,
                                       w = function(x) 1.2 / (1 + exp( - x)))

}
names(Delta_true) <- names(Delta_other) <- paste("n =", ns)
```

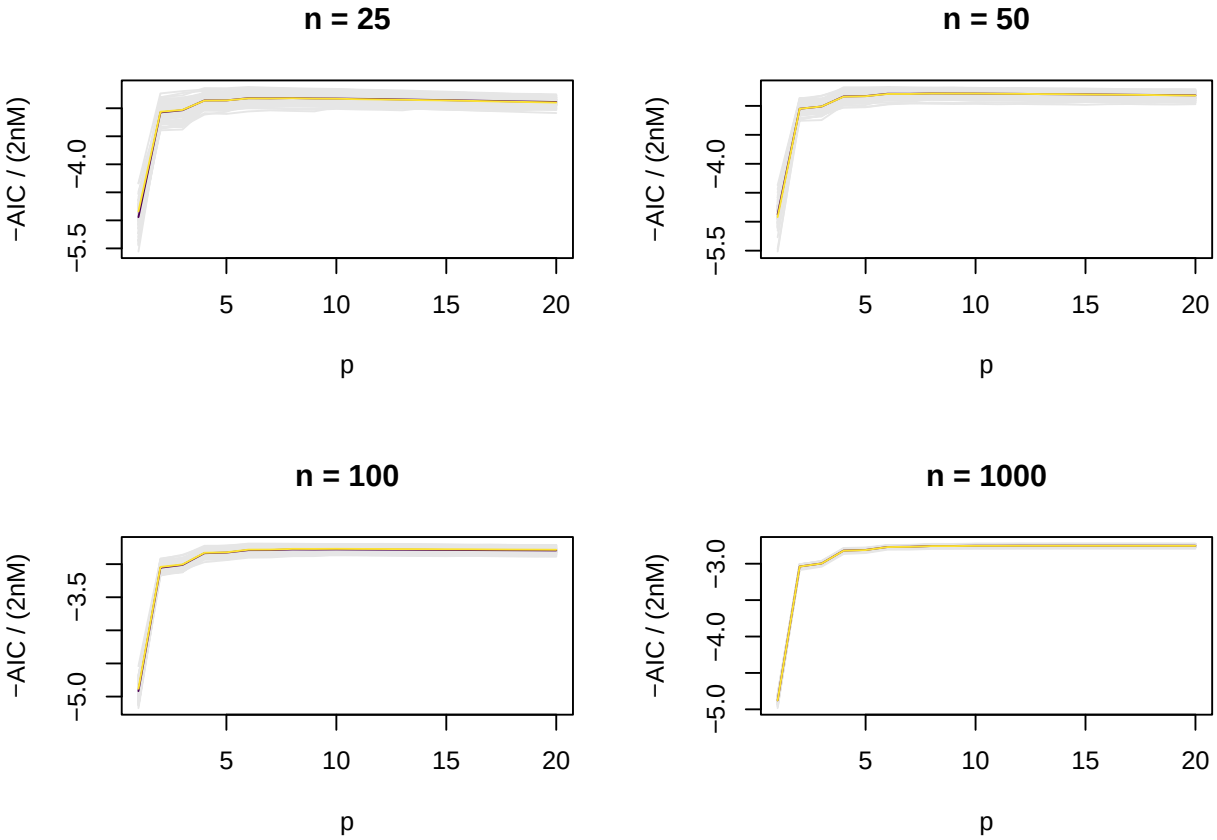
For $w(x) = 0.001\,(100 + x + x^2 + x^3)$, we get

```
cols <- hcl.colors(2)
par(mfrow = c(2, 2))
for (j in 1:length(ns)) {
    Delta_t <- - res_true[[j]]$AIC / (2 * ns[j] * M)
    plot_ic(Delta_t, ylab = "-AIC / (2nM)", main = paste("n =", ns[j]))
    lines(1:pmax, colMeans(Delta_t), col = cols[1])
    lines(1:pmax, Delta_true[[j]], col = cols[2])
}
```



and for $w(x) = 1.2/\{1 + \exp(-x)\}$, we get

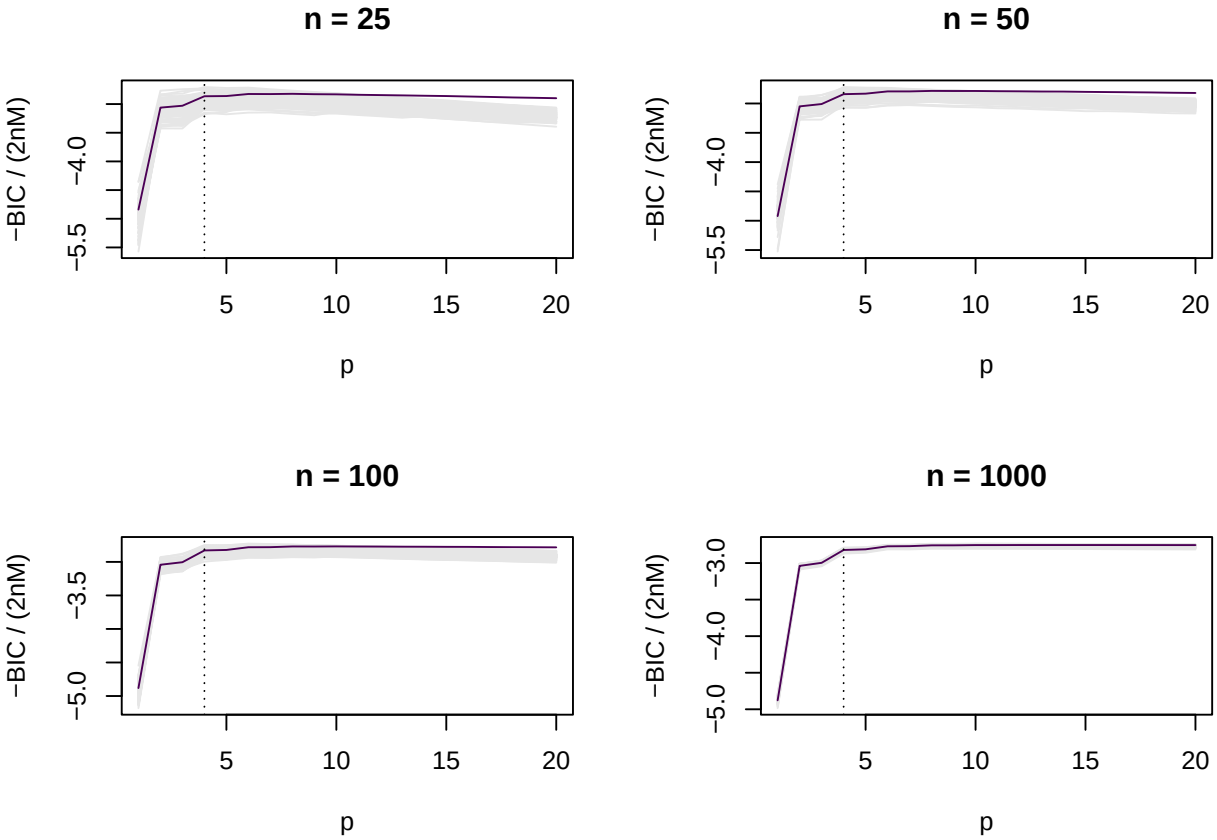
```
par(mfrow = c(2, 2))
for (j in 1:length(ns)) {
    Delta_t <- - res_other[[j]]$AIC / (2 * ns[j] * M)
    plot_ic(Delta_t, ylab = "-AIC / (2nM)", main = paste("n =", ns[j]))
    lines(1:pmax, colMeans(Delta_t), col = cols[1])
    lines(1:pmax, Delta_other[[j]], col = cols[2])
}
```



The purple piecewise linear functions are the simulation-based estimates of Δ based on AIC, and the yellow piecewise linear functions are the simulation-based estimates of Δ based on its definition. In the current setting, we can see that the two simulation-based estimates are almost identical in value.

The fact that BIC does not estimate Δ can now be made directly apparent. For example, for $w(x) = 1.2/\{1 + \exp(-x)\}$,

```
par(mfrow = c(2, 2))
for (j in 1:length(ns)) {
    plot_ic(- res_other[[j]]$BIC / (2 * ns[j] * M), ylab = "-BIC / (2nM)", main = paste("n =", ns[j])
    lines(1:pmax, Delta_other[[j]], col = cols[1])
    abline(v = 4, lty = 3)
}
```


n = 25
n = 50
n = 100
n = 1000
−BIC / (2nM)
p
5
10
15
20
−5.5
−4.0
−5.0
−3.5
−3.0

Chapter 6

Lab 2 (with solution)

6.1 Exercise

The data in hip.txt are from Crowder & Hand (1990), and we can put them in an R `data.frame` by

```
hip_url <- url("https://ikosmidis.com/files/APTS-SM-Notes/resources/hip.txt")
hip <- read.table(hip_url,
                  col.names = c("y", "age", "sex", "subj", "time"),
                  colClasses = c(rep("numeric", 4), "factor"))
str(hip)
```

```
'data.frame':   88 obs. of  5 variables:
 $ y   : num  47.1 31.1 32.8 44.1 31.5 ...
 $ age : num  66 66 66 70 70 70 44 44 44 70 ...
 $ sex : num  0 0 0 0 0 0 0 0 0 0 ...
 $ subj: num  1 1 1 2 2 2 3 3 3 4 ...
 $ time: Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
```

The variable `y` contains measurements of haematocrit on 30 patients (`subj`) on up to three occasions (`time`), one before a hip-replacement operation (coded as 1), and two afterwards (coded as 2 and 3). `time` is treated here as a categorical variable. The `age` and `sex` (0 for male, 1 for female) of the patients are also recorded.

The primary interest in this study is in possible differences in the evolution of *haematocrit* between males and females and whether there is an age effect.

1. For each value of `sex`, plot the time profiles of the response variable for each subject.

   Do you think you think it is necessary to include a random intercept for the subject? What about a random slope for time?

2. We will analyse these data using linear mixed models (LMMs) of the form

$$\begin{aligned} Y_{ij} \mid x_{ij}, z_{ij} &\overset{\text{ind}}{\sim} \mathrm{N}(\mu_{ij}, \sigma^2)\,, \\ \mu_{ij} &= x_{ij}^\top \beta + z_{ij}^\top b_i\,, \\ b_i &\overset{\text{ind}}{\sim} \mathrm{N}(0, \Sigma_b^*)\,, \end{aligned} \tag{6.1}$$

   where Y_{ij} is the random variable corresponding to the haematocrit measurement for subject i at time j, and x_{ij} and z_{ij} are fixed-effects and mixed-effects covariates, respectively.

   Consider building a set of candidate LMMs for explaining haematocrit, using the fixed-effects of `age`, `sex` and `time` (and possibly interactions of `sex` to `age` and `time`) and random-effects of `subj` and `time`. If you want to have models with interaction effects in your candidate set, make sure that

you respect the "marginality constraints", that is the model should have main effects for all terms from which interactions are formed.

What is your chosen model?

3. For the model you chose, plot the predicted haematocrit for each patient against time.

   Use your chosen model to predict the full haematocrit profiles of any patients that do not have haematocrit measurements at all three time points.

Hints for plotting

For plotting, you may adapt the code used for producing Figure 2.2 for the rat growth data, or you may use the `ggplot2` R package.

Hints for modelling

LMMs for clustered data can be fitted in R using the `lmer()` function from the `lme4` package. For example,

```
library(lme4)
hip_lmm1 <- lmer(y ~ age + sex + time + (1 | subj), data = hip)
```

fits the model with $x_{ij} = (1, \text{age}_i, \text{sex}_i, I(\text{time}_{ij} = 2), I(\text{time}_{ij} = 3))^\top$, and $z_{ij} = 1$.

The default estimation method in `lmer()` is REML. If you want to obtain maximum likelihood estimates (for example, for use in model comparison), they can be obtained using the additional argument `REML = FALSE`.

You might find useful some of the following methods for the object that `lmer()` returns: `summary`, `fitted`, `residuals`, `fixef` (fixed effects estimates), `ranef` (random effects estimates), `VarCorr` (variance estimates) `coef` (coefficient estimates at cluster level, incorporating fixed and random effects), `AIC`, `BIC` and `predict`.

For more information, see `?lmer`.

Quantities in the general LMM definiton in (2.5)

In terms of the general definition of LMMs in (2.5), in order to get a better understanding of what the formula interface of `lmer()` does, for `hip_lmm1`, Y is

```
mf1 <- model.frame(hip_lmm1)
model.response(mf1)
```

```
    1     2     3     4     5     6     7     8     9    10    11    12    13
47.10 31.05 32.80 44.10 31.50 37.00 39.70 33.70 24.50 43.30 18.35 36.60 37.40
   14    15    16    17    18    19    20    21    22    23    24    25    26
32.25 29.05 45.70 35.50 39.80 44.90 34.10 32.05 42.90 32.05 46.05 28.80 37.80
   27    28    29    30    31    32    33    34    35    36    37    38    39
42.10 34.40 36.05 38.25 29.40 30.50 43.00 33.70 36.65 37.80 26.60 30.60 37.25
   40    41    42    43    44    45    46    47    48    49    50    51    52
26.50 38.45 27.95 33.95 27.00 32.50 31.95 38.35 32.30 37.90 38.80 32.55 26.85
   53    54    55    56    57    58    59    60    61    62    63    64    65
44.65 32.25 34.20 38.00 27.10 37.85 34.00 23.20 25.95 44.80 37.20 29.70 45.95
   66    67    68    69    70    71    72    73    74    75    76    77    78
29.10 26.70 41.85 31.95 37.60 38.00 31.65 35.70 42.20 34.00 33.25 39.70 33.45
   79    80    81    82    83    84    85    86    87    88
32.65 37.50 28.20 30.30 34.55 30.95 28.75 35.50 24.70 29.75
```

X is

```
model.matrix(hip_lmm1, type = "fixed")
```

```
  (Intercept) age sex time2 time3
1           1  66   0     0     0
```

2	1	66	0	1	0
3	1	66	0	0	1
4	1	70	0	0	0
5	1	70	0	1	0
6	1	70	0	0	1
7	1	44	0	0	0
8	1	44	0	1	0
9	1	44	0	0	1
10	1	70	0	0	0
11	1	70	0	1	0
12	1	70	0	0	1
13	1	74	0	0	0
14	1	74	0	1	0
15	1	74	0	0	1
16	1	65	0	0	0
17	1	65	0	1	0
18	1	65	0	0	1
19	1	54	0	0	0
20	1	54	0	1	0
21	1	54	0	0	1
22	1	63	0	0	0
23	1	63	0	1	0
24	1	71	0	0	0
25	1	71	0	1	0
26	1	71	0	0	1
27	1	68	0	0	0
28	1	68	0	1	0
29	1	68	0	0	1
30	1	69	0	0	0
31	1	69	0	1	0
32	1	69	0	0	1
33	1	64	0	0	0
34	1	64	0	1	0
35	1	64	0	0	1
36	1	70	0	0	0
37	1	70	0	1	0
38	1	70	0	0	1
39	1	60	1	0	0
40	1	60	1	1	0
41	1	60	1	0	1
42	1	52	1	1	0
43	1	52	1	0	1
44	1	52	1	0	0
45	1	52	1	1	0
46	1	52	1	0	1
47	1	75	1	0	0
48	1	75	1	1	0
49	1	75	1	0	1
50	1	72	1	0	0
51	1	72	1	1	0
52	1	72	1	0	1
53	1	54	1	0	0
54	1	54	1	1	0
55	1	54	1	0	1
56	1	71	1	0	0
57	1	71	1	1	0

```
58           1  71   1     0     1
59           1  58   1     0     0
60           1  58   1     1     0
61           1  58   1     0     1
62           1  77   1     0     0
63           1  77   1     1     0
64           1  77   1     0     1
65           1  66   1     0     0
66           1  66   1     1     0
67           1  66   1     0     1
68           1  53   1     0     0
69           1  53   1     1     0
70           1  53   1     0     1
71           1  74   1     0     0
72           1  74   1     1     0
73           1  74   1     0     1
74           1  78   1     0     0
75           1  78   1     1     0
76           1  78   1     0     1
77           1  74   1     0     0
78           1  74   1     1     0
79           1  74   1     0     1
80           1  79   1     0     0
81           1  79   1     1     0
82           1  79   1     0     1
83           1  71   1     0     0
84           1  71   1     1     0
85           1  71   1     0     1
86           1  68   1     0     0
87           1  68   1     1     0
88           1  68   1     0     1
attr(,"assign")
[1] 0 1 2 3 3
attr(,"contrasts")
attr(,"contrasts")$time
[1] "contr.treatment"

attr(,"msgScaleX")
character(0)
```

and Z is

```
model.matrix(hip_lmm1, type = "random")
```

```
88 x 30 sparse Matrix of class "dgCMatrix"
  [[ suppressing 30 column names '1', '2', '3' ... ]]


1  1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2  1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
3  1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4  . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
5  . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
6  . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
7  . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
8  . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
9  . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

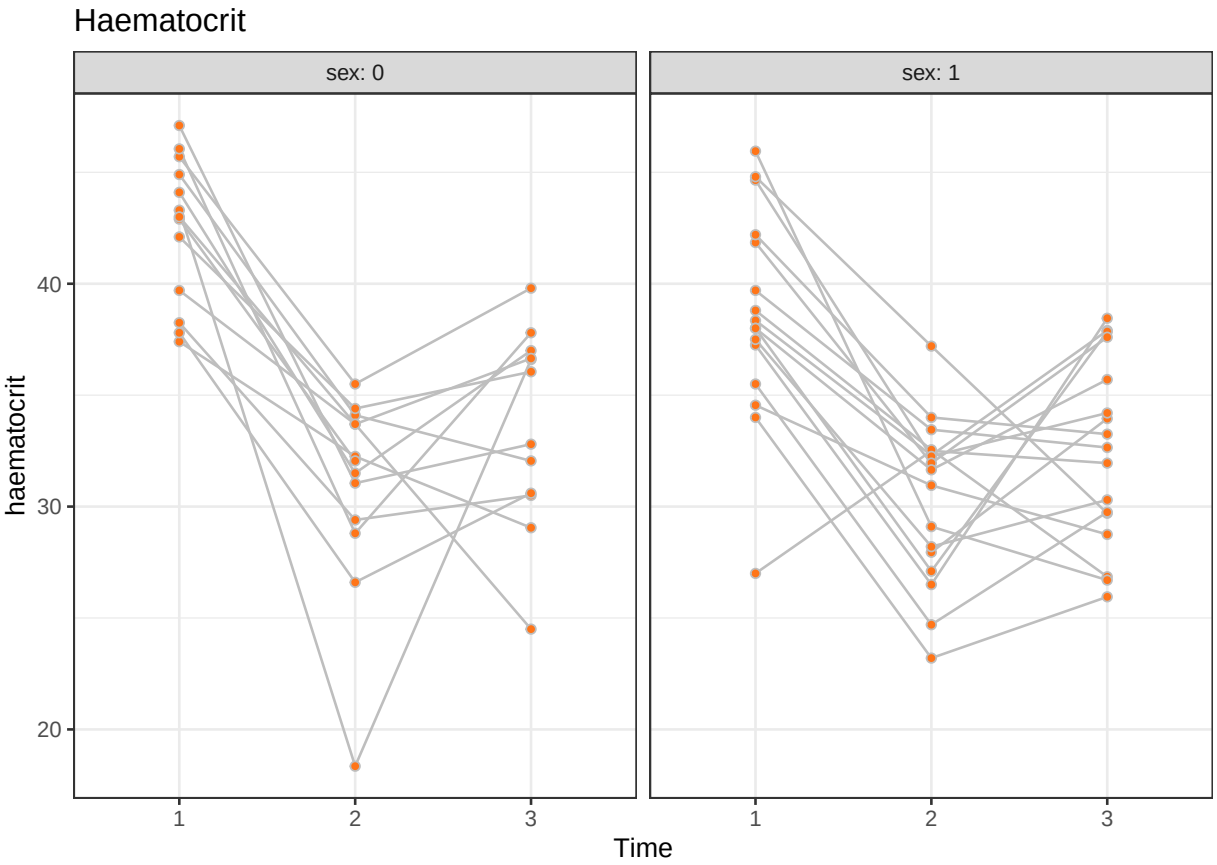
```
10 . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . .
11 . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . .
12 . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . . .
13 . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . .
14 . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . .
15 . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . . .
16 . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . .
17 . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . .
18 . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . . .
19 . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . .
20 . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . .
21 . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . . .
22 . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . .
23 . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . . .
24 . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . .
25 . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . .
26 . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . . .
27 . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . .
28 . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . .
29 . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . . .
30 . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . .
31 . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . .
32 . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . . .
33 . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . .
34 . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . .
35 . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . . .
36 . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . .
37 . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . .
38 . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . . .
39 . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . .
40 . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . .
41 . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . . .
42 . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . .
43 . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . . .
44 . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . .
45 . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . .
46 . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . . .
47 . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . .
48 . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . .
49 . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . . .
50 . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . .
51 . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . .
52 . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . . .
53 . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . .
54 . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . .
55 . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . . .
56 . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . .
57 . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . .
58 . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . . .
59 . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . .
60 . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . .
61 . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . . .
62 . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . .
63 . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . .
64 . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . . .
65 . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . . .
```

```
66 . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . .
67 . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . .
68 . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . .
69 . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . .
70 . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . . .
71 . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . .
72 . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . .
73 . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . . .
74 . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . .
75 . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . .
76 . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . . .
77 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . .
78 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . .
79 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . . .
80 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . .
81 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . .
82 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 . .
83 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 .
84 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 .
85 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1 .
86 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
87 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
88 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
```

6.2 Solution

1.

```
library("ggplot2")
hip_pl <- ggplot(hip) +
    geom_line(aes(time, y, group = subj), col = "grey") +
    geom_point(aes(time, y), fill = "#ff7518", pch = 21, col = "grey") +
    facet_grid(~ sex, labeller = label_both) +
    labs(y = "haematocrit", x = "Time") +
    theme_bw() +
    labs(title = "Haematocrit")
hip_pl
```



There appears to be heterogeneity between patients for both males and females. We observe no profound differences in profiles between males and females, perhaps apart from males having slightly elevated haematocrit levels. Also, with a few exceptions, the profiles appear to be roughly parallel across patients, or, equivalently, there appears to be no substantial heterogeneity in time.

2.

Let's fit all possible nested models of the model that includes an interaction of `sex` with `age` and `time` for the fixed effects, and random effects for patient, time or both patient and time, and compute the AIC and BIC for each model. In doing so, we need to respect marginality constraints. In other words, the list of candidate models should include all possible models with main effects ($2\hat{}3 = 8$ in that case), and from the models with interactions we should include only those that include their respective main effects. We can easily list the resulting set of candidate models for the fixed-effects in that case:

Main effects only	Interactions and respective main effects
`y ~ 1`	
`y ~ age`	
`y ~ sex`	
`y ~ time`	
`y ~ age + sex`	`y ~ age + sex + age:sex`
`y ~ age + time`	
`y ~ sex + time`	`y ~ sex + time + sex:time`
`y ~ age + sex + time`	`y ~ age + sex + time + age:time`
	`y ~ age + sex + time + sex:time`
	`y ~ age + sex + time + sex:time + age:time`

We can now include the above model formulas in R in a list, after adding a random intercept for patient, and use a `for` loop to fit all models using `lmer()` with `REML = FALSE`, and compute AIC, BIC and AICc for each model.

Instead, we can simply fit the model with all interactions with `REML = FALSE`, and use the `dredge()` function of the `MuMIn` R package to compute information criteria for all other models, as we did in Example 1.2 with linear models for the nodal involvement data. `dredge()` is convenient because marginality constraints are respected; see `?dredge`.

The code chunk below does that for AIC, BIC, and AICc.

```
library("MuMIn")
hip_full_subj <- lmer(y ~ sex * (age + time) + (1 | subj), data = hip,
                      REML = FALSE,
                      na.action = "na.fail")
ms <- dredge(hip_full_subj, rank = "AIC", extra = c("AIC", "AICc", "BIC"))
ms
```

```
Global model call: lmer(formula = y ~ sex * (age + time) + (1 | subj), data = hip,
    REML = FALSE, na.action = "na.fail")
---
Model selection table
   (Int)       age    sex tim age:sex sex:tim   AIC  AICc   BIC df   logLik
7  41.38           -1.860   +                 507.1 508.2 522.0  6 -247.574
23 42.48           -3.861   +               + 507.9 509.7 527.7  8 -245.952
5  40.35                    +                 508.4 509.2 520.8  5 -249.212
8  39.08 0.0351800 -1.918   +                 508.8 510.2 526.1  7 -247.380
24 40.04 0.0375300 -3.941   +               + 509.5 511.8 531.8  9 -245.731
6  38.68 0.0250900          +                 510.2 511.3 525.1  6 -249.122
16 41.25 0.0018280 -5.281   + 0.05108         510.6 512.4 530.4  8 -247.286
32 42.34 0.0022280 -7.506   + 0.05404       + 511.3 514.1 536.0 10 -245.626
3  35.71           -2.010                   567.7 568.2 577.7  4 -279.873
1  34.57                                    568.3 568.6 575.7  3 -281.142
4  32.60 0.0477400 -2.092                   569.3 570.0 581.7  5 -279.647
2  32.16 0.0363400                          570.0 570.5 579.9  4 -281.014
12 35.68 0.0004708 -6.880     0.07267        571.1 572.1 585.9  6 -279.527
     AIC delta weight
7  507.1  0.00  0.286
23 507.9  0.76  0.196
5  508.4  1.28  0.151
8  508.8  1.61  0.128
24 509.5  2.31  0.090
6  510.2  3.10  0.061
16 510.6  3.43  0.052
32 511.3  4.10  0.037
3  567.7 60.60  0.000
1  568.3 61.14  0.000
4  569.3 62.15  0.000
2  570.0 62.88  0.000
12 571.1 63.91  0.000
Models ranked by AIC(x)
Random terms (all models):
  1 | subj

Fixed term is "(Intercept)"

boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
boundary (singular) fit: see help('isSingular')
```

We see that AIC and AICc agree that the best model is the model with main effects for `sex` and `time`

and a patient-specific random intercept. That model is only second-best in terms of BIC (BIC of 522.01) after the model with with a main effect of only `time` (BIC of 520.81).

So, from the models with patient-specific intercepts, there is evidence for the model `y ~ sex + time + (1 | subj)`.

Let's try to do the same including a patient-specific random slope for `time` in the model:

```
hip_full_subj_time <- lmer(y ~ sex * (age + time) + (time | subj), data = hip,
                           REML = FALSE,
                           na.action = "na.fail")
```

```
Error: number of observations (=88) <= number of random effects (=90) for term (time | subj); the ra
```

We get an error message, because there are now too many different random effect terms in the model to be able to estimate them all from the data available.

3.

Let's fit the chosen model using REML

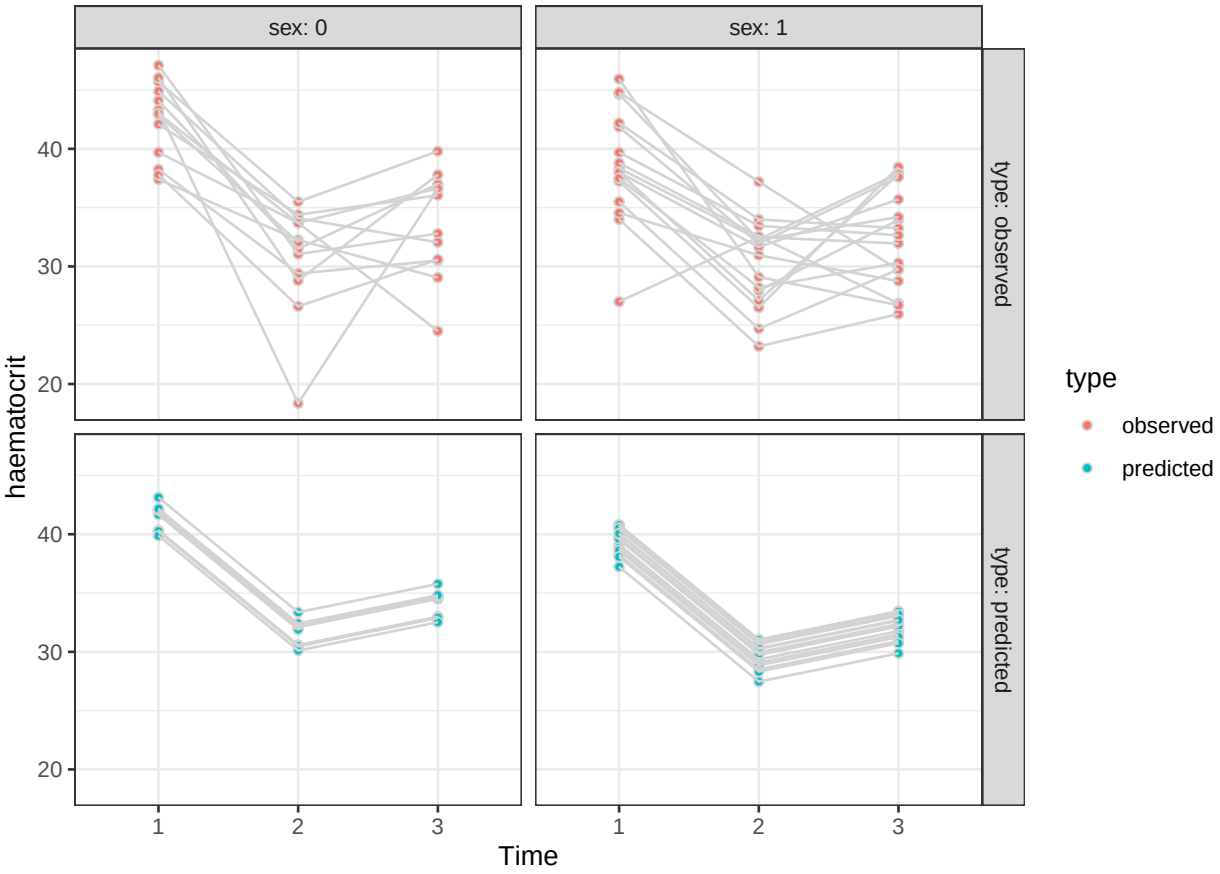
```
hip_mod <- lmer(y ~ sex + time + (1 | subj), data = hip)
summary(hip_mod)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ sex + time + (1 | subj)
   Data: hip

REML criterion at convergence: 489.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.1899 -0.5637  0.0305  0.6154  1.6744

Random effects:
 Groups   Name        Variance Std.Dev.
 subj     (Intercept)  2.92    1.709
 Residual             14.55    3.815
Number of obs: 88, groups:  subj, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept)  41.3847     0.9661  42.838
sex          -1.8600     1.0360  -1.795
time2        -9.7657     0.9947  -9.817
time3        -7.3572     1.0047  -7.323

Correlation of Fixed Effects:
      (Intr) sex    time2
sex   -0.596
time2 -0.518 -0.011
time3 -0.504 -0.026  0.505
```

We can use the `predict()` method to get predictions for the haematocrit profiles of the patients in the sample, and compare them to the observed profiles.

```
hip_pred <- within(hip, y <- predict(hip_mod))
hip$type <- "observed"
hip_pred$type <- "predicted"
ggplot(rbind(hip, hip_pred)) +
    geom_point(aes(time, y, fill = type), pch = 21, col = "lightgray") +
```

```
    geom_line(aes(time, y, group = subj), col = "lightgray") +
    facet_grid(type ~ sex, labeller = label_both) +
    labs(y = "haematocrit", x = "Time") +
    theme_bw()
```



In order to identify the patients with missing haematocrit measurements, we check which patients do not have all 3 times in the data.

```
id_na <- which(tapply(hip, hip$subj, function(x) !all(1:3 %in% x$time)))
hip_na <- subset(hip, subj %in% id_na)
hip_na
```

```
       y age sex subj time     type
22 42.90  63   0    8    1 observed
23 32.05  63   0    8    2 observed
42 27.95  52   1   15    2 observed
43 33.95  52   1   15    3 observed
```

Since the chosen model only involves sex, time and subject, we want to predict the haematocrit levels for every row of the data frame

```
## Get unique sex/subject combinations for the two patients with missing
## data
new_data <- unique(hip_na[c("sex", "subj")])
## Add time
new_data <- rbind(data.frame(time = factor(1:3), new_data[1, ]),
                  data.frame(time = factor(1:3), new_data[2, ]))
new_data
```

```
  time sex subj
```

```
1    1   0    8
2    2   0    8
3    3   0    8
4    1   1   15
5    2   1   15
6    3   1   15
```

The predictions are

```
new_data$y <- predict(hip_mod, newdata = new_data)
new_data
```

```
  time sex subj        y
1    1   0    8 41.66335
2    2   0    8 31.89765
3    3   0    8 34.30619
4    1   1   15 39.52090
5    2   1   15 29.75520
6    3   1   15 32.16374
```

We can plot the observed and the predicted levels for the two patients

```
new_data$type <- "predicted"
hip_na$type <- "observed"
ggplot(rbind(new_data, hip_na[names(new_data)])) +
    geom_point(aes(time, y, fill = type), pch = 21, col = "lightgray") +
    geom_line(aes(time, y, group = interaction(subj, type)), col = "lightgray") +
    facet_grid( ~ sex, labeller = label_both) +
    labs(y = "haematocrit", x = "Time") +
    theme_bw()
```