Chapter 4

Bayesian modelling

This chapter introduces some topics that allow you to take the knowledge you have learned so far on the mechanics of fitting models, and apply them to genuine research problems, usually with multiple parameters and complex relationships between them. We shall look at hierarchical models, which allow multiple data sets to be combined in a single analysis, regression and other generalised linear models, which you should have seen already and which allow relationships between variables in datasets to be understood, model checking and comparison, which are necessary when multiple models could feasibly be selected for a single problem.

4.1 Hierarchical modelling

To introduce the concept of hierarchical models, let us consider three motivating case studies.

4.1.1 Case study 1: damping off of radish

Radish is a member of the economically important Brassicaceae family of crops, which are often pathogenised by *Rhizoctonia solani*, a fungus which in radish seedlings causes damping off. *R. solani* spreads from the soil to plants and its spread may be hastened by infecting neighbouring plants. Otten et al (2003; *Ecology* 84:3232–9) performed an experimental inoculation study to quantify the effect of increasing levels of inoculation of the soil in disease levels in radish seedlings, in controlled, laboratory conditions. In this study, they set up 26 isolated populations, or microcosms, of up to 414 plants (on an 18×23 grid, though some seeds did not emerge from the soil), and colonised the soil at the same time they sowed the seedlings. Then then scored each

Table 4.1: Inoculuation density (Inoc.), number infected (Inf.) and number at risk (N) in each of 26 radish microcosms studied by Otten et al (2003).

Inoc.	**Inf.**	**N**	**Inoc.**	**Inf.**	**N**
H	349	410	L	192	403
H	368	398	L	217	408
H	379	404	L	217	399
H	343	405	L	233	403
H	327	401	L	239	398
H	316	395	L	153	399
H	324	398	L	238	404
H	362	386	L	289	401
H	373	398	L	263	400
H	308	404	L	224	402
H	363	405	L	279	395
H	358	400	L	236	405
H	295	404	L	157	401

population daily for disease, though we shall focus our analysis on the data on day 21, when the experiment was completed, only. The data are tabulated in 4.1 and plotted in figure 4.1. For each microcosm, the posterior for the final attack rate—assuming independence between hosts in the same microcosm (i.e. that the disease is *not* hastened by previous infections), and taking a $U(0, 1)$ prior independently for each microcosm—is also plotted. Note the lack of overlap between several of these posteriors even within the same treatment arm. This indicates either

- the model is wrong and actually the process is the same in each microcosm in each arm; or

- there are inherent differences between the risk of infection within different microcosms within each treatment arm, i.e. variability is present between and within treatments.

In any case, the variability complicates the reporting of an overall treatment effect, as one could not justifiably assume the same infection risk within each microcosm.

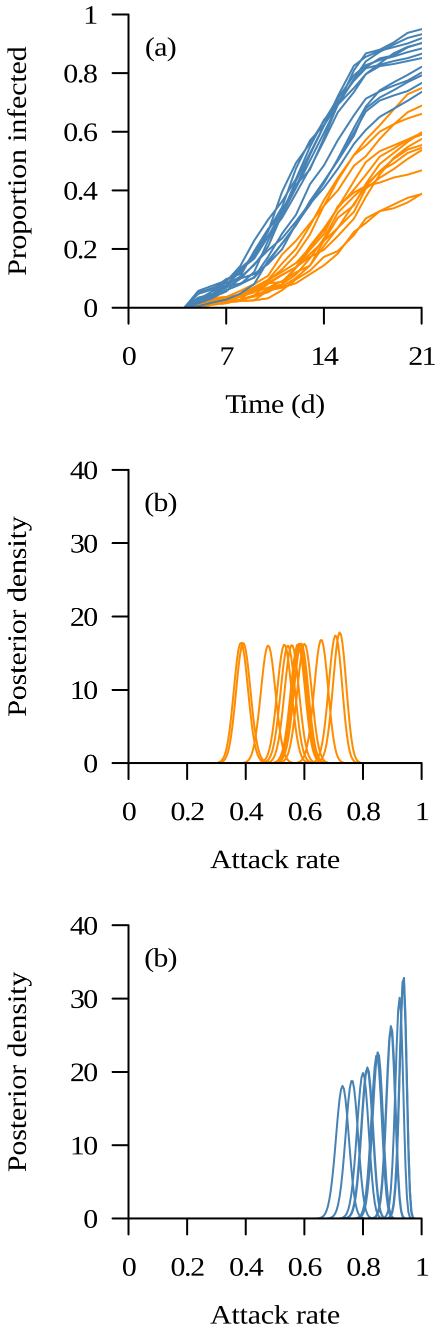

Figure 4.1: **Radish data.** (a) shows the full data, with the low inoculum treatment in orange and the high in blue. (b) shows posterior densities for the final attack rate in the high treatment and (c) in the low treatment. Note the lack of overlap for some densities in panels b and c.

Table 4.2: Statin Meta Analysis data. Study names, number of deaths in statin x_s and placebo x_p arms, and number at risk in each arm n. Data from Afilalo et al (2008).

study	x_s	n_s	x_p	n_p
4s	67	518	96	503
care	77	640	108	643
flare	2	179	6	187
hps	963	5366	1089	5331
lipid	287	1741	365	1773
lips	23	324	32	299
plac1	1	42	2	52
prosper	110	934	128	899
regress	1	75	1	63

4.1.2 Case study 2: meta-analysis of statins in the elderly

Statins are drugs that reduce cholesterol and may therefore reduce mortality among those at risk of heart attacks, particularly the elderly and those with coronary heart disease (CHD). Afilalo et al (2008; *J Am Coll Cardiol* 51:37–45) conducted a systematic review of the literature, searching for randomised controlled trials in which statins or a placebo were randomised to elderly patients ($\geq$ 65y) with CHD. Several outcomes were considered but we focus on all cause 5y mortality. The data from 9 studies are tabulated in table 4.2 The posterior distribution of relative risks, taking independent uniform priors on $[0, 1]$ for all variables and then generating Monte Carlo samples, are plotted in figure 4.2. Analysing the studies separately provides at best ambiguous evidence of a beneficial effect, but there are substantial differences in study design and patients (witness the different baseline risks) that make an assumption of common mortality rates hard to justify. It would therefore be of interest to “pool” the information from the 9 studies, while accounting for differences between them.

4.1.3 Case study 3: lip cancer in Scotland

This is a traditional example used to teach hierarchical Bayes, and one I learned as a study and wish to pass on, for involves my home country. Lip cancer is a rare disease associated (now) to exposure to sunlight. Clayton and Kaldor (1987; *Biometrics* 43:671–81) analyse data on lip cancer occurence

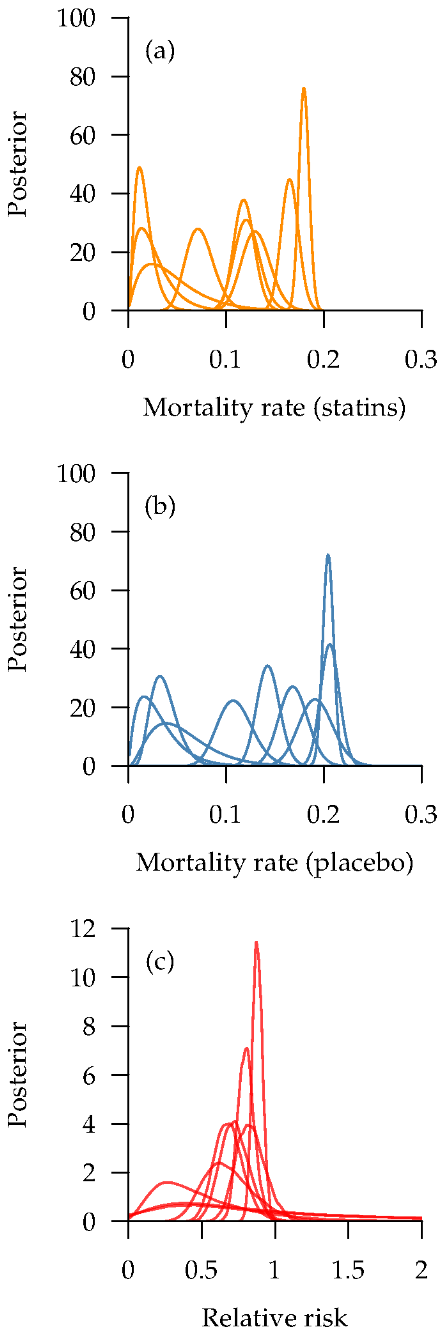

Figure 4.2: **Statin data.** Posteriors for 5y mortality in treatment (a) and placebo (b) arms, as well as relative risk (c) are plotted.

(from 1975 to 1980) in different regions of Scotland—a small country of ca. 5 million souls in northern Europe—along with expected number of cases per region, based on age and population size. If the number of cases in area i is O_i, and we assume $O_i \sim Po(\lambda_i E_i)$ where E_i is the (known) expected number, which is scaled by λ_i to reflect the increase or decrease in risk in location i due to unexplained factors, then a posterior for each λ_i can be obtained by setting an appropriate prior (e.g. $\lambda_i \sim U(0, 100)$). However, because it is a rare disease (which makes the Poisson distribution appropriate in the first place), sparsely populated regions of Scotland would have very uncertain estimates. It would therefore be valuable to, somehow, give an informative prior to help obtain usable estimates in such "ulu" regions. The data are available in the geoBUGS examples in openBUGS (in which a more sophisticated model is fit than here) or from the paper by Clayton and Kaldor (1987).

4.1.4 Hierarchical models: why

Consider two extreme models for the radish data. In both we might assume, if x_i is the number of diseased plants in microcosm i, that $x_i \sim Bin(n_i, p_i)$. The models differ in the priors for the risk, p_i.

One extreme takes p_i to be *a priori* $Be(1, 1)$ *independently* of other replicates. In this scenario, the data from each microcosm are analysed independently, and the information from one replicate does not impact the understanding of another.

The other extreme takes $p_i = p_j = p$ for all i and j within a treatment arm and assumes $p \sim Be(1, 1)$, i.e. each infection in the neighbouring microcosm tells me just as much about one particular microcosm as each in that microcosm itself.

Both involve quite strong assumptions. The first approach implies the different replicates provide no mutual information, i.e. all are so different from each other that you can ignore all but one. The second implies that there are *no* differences between the underlying process.

Hierarchical models fall between these two extremes. The parameters are not assumed to be identical, nor independent, but rather to be "similar" to the parameters in other replicates. This is done by assuming each replicate's parameters come from a specific distribution with (hyper) parameters that are, themselves, estimated.

One example of an hierarchical model for the radish data is:

$$\begin{aligned} x_i &\sim Bin(n_i, p_i) \\ p_i &\sim Be(a, b) \\ a &\sim Exp(1/100) \\ b &\sim Exp(1/100). \end{aligned}$$

In this model, the likelihood for the datum x_i is unchanged, but now p_i and p_j are related to each other via a common distribution, $Be(a, b)$. So if I tell you the values of p_i for $i = 1, \ldots, 12$ you now have a good idea of the distribution of (a, b) and therefore have an informative prior for p_{13}. Here, a and b—which are sometimes called hyperparameters, as they belong one level up from the parameters p_i—are themselves given prior distributions (or hyperpriors).

Hierarchical models bring several benefits.

- They allow you to give an *informative prior* for parameters directly from data. The data from replicates 1–12 tell you about the kind of parameter values that are suitable for replicate 13, the data from replicates 1–11 and 13 tell you about replicate 12, and so on. By fitting a large model to all 13 datasets at one time, information is thus *pooled* between them.

- The effect of pooling is to shirnk individual estimates towards the grand mean. This is an effective way to reduce bias. It also effectively narrows the posteriors for parameters of data components with low information content.

- They allow you to quantify between dataset variability. For instance, for the radish data, the fact that the variance of the model for p_i given the data is so high gives us reason to doubt the underlying model (of independence between hosts within the same microcosm).

- They allow you to report overall effects that account for between group variability. For instance, in the statins study, we are interested in the overall reduction in mortality, not the reduction within particular studies, but the variability between studies needs to be accounted for in order to describe the overall effect correctly.

4.1.5 Hierarchical models: how

The first step is to identify where in the dataset there is replication with parameters that might feasibly be modelled by a distribution. The support of the parameters will determine the appropriate kind of distribution, which must account at a minimum for overall tendency and variability, by having a model with at least two parameters. If the parameters to be modelled hierarchically are scalars θ_i, then

- if they take support on the real line, an appropriate distribution might be normal;
- if they are probabilities, an appropriate distribution might be beta, or normal with a logit link;
- if they are real but positive, then models might include log-normal, gamma, or Weibull.

Let us consider the three case studies again to see how one might set up an hierarchical model for the data.

4.1.6 Model 1: damping off of radish

The data are tabulated in table 4.1.

A model might be:

$$\begin{aligned} x_{hi} &\sim Bin(n_{hi}, p_{hi}) \\ p_{hi} &\sim Be(a_h, b_h) \\ a_h &\sim Exp(1/100) \\ b_h &\sim Exp(1/100), \end{aligned}$$

where h is 1 for the high and 0 for the low inoculum density. This model assumes independence between the two densities (and hence they might be analysed separately) but takes a beta model for each probability. The $Exp(1/100)$ distributions give a marginal for p_{hi}, unconditional on the data, which is approximately uniform:

```
  a=rexp(10000,0.01)
  b=rexp(10000,0.01)
  p=rbeta(10000,a,b)
  hist(p)
```

This can be run with JAGS with the following model

```
model{
  for(rep in 1:13)
  {
    xH[rep]~dbin(pH[rep],nH[rep])
    pH[rep]~dbeta(aH,bH)
  }
  for(rep in 1:13)
  {
    xL[rep]~dbin(pL[rep],nL[rep])
    pL[rep]~dbeta(aL,bL)
  }
  aL~dexp(0.01)
  bL~dexp(0.01)
  aH~dexp(0.01)
  bH~dexp(0.01)
}
```

and following R code:

```
  library(rjags)
  dataset=list(xH=data$x[1:13],nH=data$n[1:13],
               xL=data$x[14:26],nL=data$n[14:26])
  initialisation=list(aL=1,aH=1,bL=1,bH=1,
                      pH=runif(13),pL=runif(13))
  jagmod=jags.model("model_radish.txt",data=dataset,
                     inits=initialisation,n.chains=1)
  update(jagmod, n.iter=1000, progress.bar="text")
  posterior = coda.samples(jagmod, c("pH","pL","aH","bH","aL","bL"),
                           n.iter=100000, progress.bar="text",thin=10)
```

Here, I have run 100 000 iterations as the mixing of the hierarchical components was not so swift. This took less than a minute to run and gave satisfactory convergence diagnostics. The a and b parameters can be converted to means and standard deviations thus:

```
 post=as.data.frame(as.matrix(posterior))
 a=post$alphaH;b=post$betaH
 post$muH=a/(a+b)
 post$sigmaH=sqrt(a*b/((a+b)^2*(a+b+1)))
 #etc
```

to give an overall estimate for the attack rate of 86% (95%I: 82–89%) in the high and 56% (51–61%) in the low inoculum arms, a 66% (59–72%) reduction. Variability between replicates in the high inoculum arm had posterior mean standard deviation 6% (4–9%), while that in the low inoculum arm was 9% (6–13%). The estimates are plotted in figure 4.3.

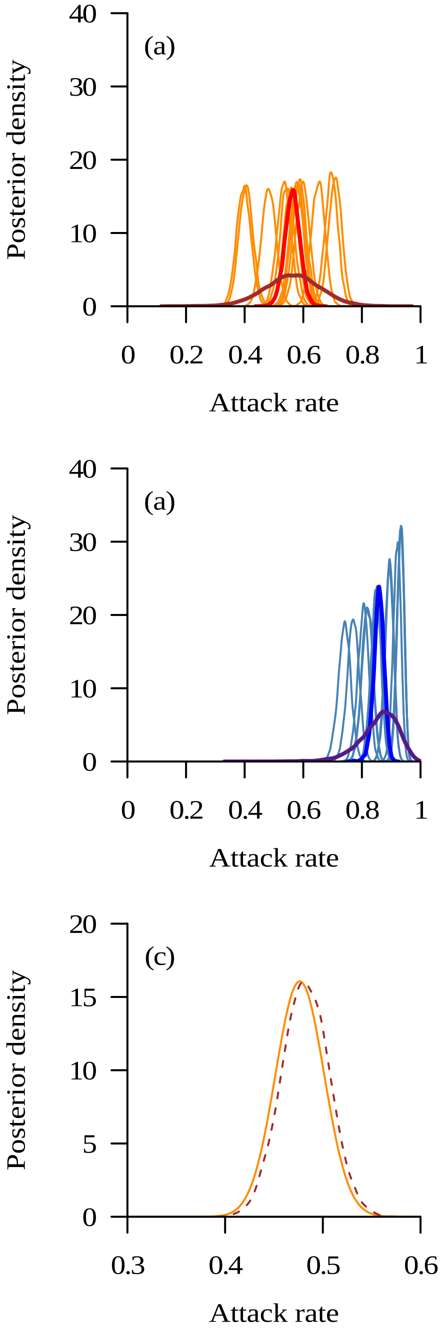

Figure 4.3: Posteriors for the hierarchical model for the low (a) and high (b) inoculum treatments. Solid orange and light blue lines indicate replicate estimates, the red and blue lines the overall mean, and the purple and brown lines the predictive distribution of a new replicate. Panel (c) shows shrinkage for one replicate towards the overall mean: orange is the non-hierarchical model, brown the hierarchical one.

4.1.7 Model 2: meta-analysis of statins in the elderly

In this example, the variability between studies should apply to *both* treatment arms, and represents differences in the patients (such as their age and other risk factors) in each study. This is hard to achieve using the beta model used for the radish data, and so we will use a normal model, transformed to yield a probability using the logit function.

The model file used in JAGS is

```
model{
  for(study in 1:9)
  {
    xs[study]~dbin(ps[study],ns[study])
    xp[study]~dbin(pp[study],np[study])
    ps[study] <- ilogit(beta[study])
    pp[study] <- ilogit(gamma+beta[study])
    beta[study] ~ dnorm(mu,tau)
  }

  tau <- 1/(sigma*sigma)
  sigma ~ dunif(0,100)
  mu ~ dnorm(0,0.0001)
  gamma ~ dnorm(0,0.0001)
}
```

which corresponds to the following

$$\begin{aligned} x_{ti} &\sim Bin(n_{ti}, p_{ti}) \\ \log\frac{p_{0i}}{1-p_{0i}} &= \beta_s \\ \log\frac{p_{1i}}{1-p_{1_i}} &= \beta_s + \gamma \\ \beta_i &\sim N(\mu, \sigma^2) \\ \mu &\sim N(0, 100^2) \\ \gamma &\sim N(0, 100^2) \\ \sigma &\sim U(0, 100). \end{aligned}$$

This says the baseline (placebo) mortality across all studies is normal (after transformation) with an overall constant impact due to the treatment governed by γ.

The model can be fitted using the following R code:

```
library(rjags)
dataset=list(xs=x$xs,ns=x$ns,xp=x$xp,np=x$np)
initialisation=list(thetap=0,thetas=0,beta=rnorm(9,0,0.1),
                    sigma=0.1)
jagmod=jags.model("model3_statins.txt",data=dataset,
                  inits=initialisation,n.chains=1)
update(jagmod, n.iter=1000, progress.bar="text")
posterior = coda.samples(jagmod, c("mu","gamma","beta","sigma"),
                    n.iter=10000, progress.bar="text",thin=1)
```

This passes convergence diagnostic checks. Output is presented in figure 4.4. The overall posterior statin effect is to reduce mortality rates by 18% (12–24%). These were obtained using the following code:

```
beta=post$mu
gamma=post$gamma
pp=inv.logit(beta)
ps=inv.logit(beta+gamma)
pr=ps/pp
de=density(pr)
```

Note that we have considered an hypothetical average population for this, which is why *beta* has been set to *mu*.

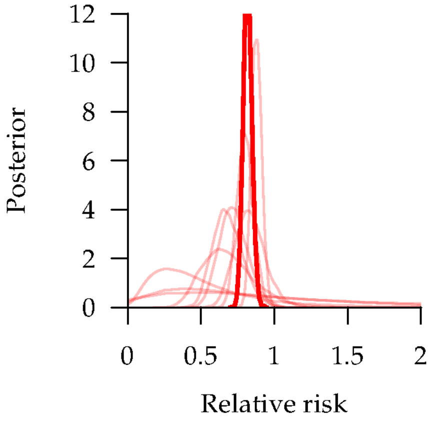

Figure 4.4: Posteriors for the hierarchical (overall effect, red) and non-hierarchical models (for each study individually, light red) for the statin data.

4.1.8 Model 3: lip cancer in Scotland

Here, we need a model that characterises differences in λ_i, a coefficient that must be positive, between areas i. One suitable option is a log normal (i.e. normal after taking logs of λ_i). The JAGS model file I used is as follows:

```
model
{
  for(i in 1:N)
  {
    O[i] ~ dpois(mu[i])
    mu[i] <- lambda[i]*E[i]
    lambda[i] ~ dlnorm(alpha,beta)
  }
  alpha ~ dunif(-100,100)
  beta ~ dunif(0,100)
}
```

The R code was

```
source('data_lips.r')
library(rjags)
dataset=list(O=observed,E=expected,N=N)
initialisation=list(alpha=0,beta=0.5,
                   lambda=rlnorm(dataset$N,0,0.5))
jagmod=jags.model("model_lips.txt",data=dataset,
                  inits=initialisation,n.chains=1)
update(jagmod, n.iter=1000, progress.bar="text")
posterior = coda.samples(jagmod, c("alpha","beta","lambda"),
                   n.iter=10000, progress.bar="text",thin=1)
```

Again, this passes convergence diagnostics.

4.1.9 When to use hierarchical models

It is usually clear from the dataset when an hierarchical model is called for. If there is, potentially, unexplained variability between some “units” in the data, then an hierarchical model allows that variability to be explained and shrunk. However, when the variability is potentially attributable to one or more factors in the data, then a regression model, relating factors or parameters for some units, might be more appropriate instead of, or in addition to, an hierarchical model.

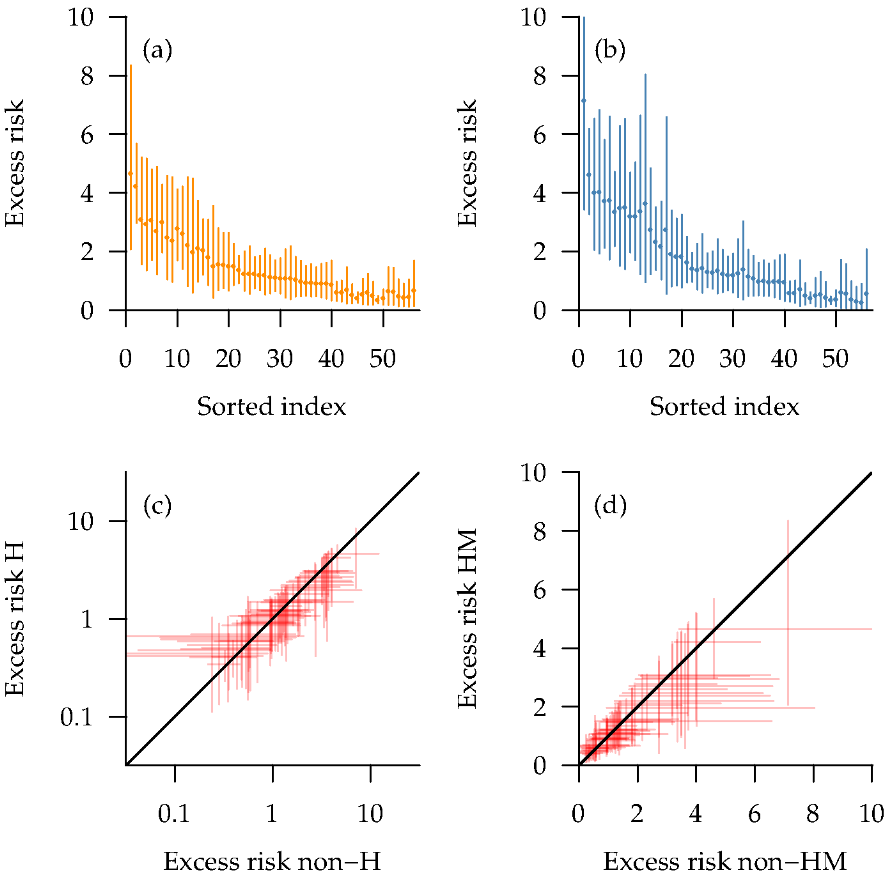

Figure 4.5: **Posteriors for lip cancer data.** Estimates of excess risk (posterior mean and 95% intervals) using the hierarchical model (a) and non-hierarchical model (b). The relationship between these estimates is plotted in panels c (log scale) and d.

4.2 Bayesian regression

The concept of ordinary linear regression is one of the most basic—and useful—statistical techniques. Unsurprisingly, regression can also be performed within the Bayesian framework. The most basic model formulation (for the data) is

$$\begin{aligned} y_i &= b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + \epsilon_i \\ \epsilon_i &\sim N(0, \sigma^2). \end{aligned}$$

(This can be written much more neatly using matrix and vector notation, but I prefer the conceptual simplicity of scalars.) Here, y_i is the outcome, or response, or dependent, variable for individual i, x_{ki} is a predictor, or independent, variable for i, b_0 is an intercept—corresponding to the expected value of y for an individual with a 0 for all predictors—b_k the co-efficient for predictor k, ϵ_i is the "error" for individual i, encapsulating the difference between what was observed and expected, and σ is the standard deviation for the error terms. If one introduces an artificial covariate $x_{0i} = 1$ for all i, then the intercept can be treated as if it were a coefficient parameter.

In this formulation, we assume:

- normal distributions for errors—often sensible, though this does not follow from any particular theory;
- that errors are homoskedastic, i.e. their variance does not depend on the values of x_{ki}; and
- that the effect on y_i of increasing x_{ki} by one unit is the same for all starting values of x_{ki}, i.e. there is a linear relationship between predictors and response.

We need to take prior distributions for each co-efficient, plus one for the intercept, plus another for the variance (or precision, or standard deviation). If you take the following improper prior:

$$\mathrm{p}(b_0, b_1, b_2, \ldots, \sigma) \propto \sigma^{-2}$$

then the posterior has a particularly nice form: the marginal posterior for σ^2 is inverse chi-squared, with parameters based on the sample size and sample standard deviation, and the posterior for the b vector conditional on σ^2 is multivariate normal with parameters based on the sample mean and variance. It is therefore possible to simulate σ and then b to get a sample from the joint posterior. See Gelman et al (2004; Bayesian Data Analysis, chapter 14) for more.

4.2.1 Transformations

There is no reason why the error term (for the outcome variable) should be normally distributed. It can be useful for some applications to transform the outcome before analysis, for instance, by taking its logarithm. However, this is not always necessary even when it is known that y is skewed or restricted to some parts of the parameter space. If for example y is skewed, it might be that the errors are still normal, depending on the distribution of the predictors. If y must be positive, or a proportion, say, it might still be well described by normal errors for the purposes of inference (think back to the SAF influenza data in chapter 3), and so transformation is only required if you wish to ensure realistic *predictions*. We will see an example later in which, ostensibly, one would think a transformation was required.

It can be useful to transform x values in the following situations:

- Firstly, if x is a categorical variable (for instance, Chinese, Malay and Indian race), then it must be transformed to something numeric before proceeding to insert it in a regression model. For a binary, or dichotomous variable, one level may be set to 0 and the other 1 (or, -1 and 1, or $-1/2$ and $1/2$). For an unordered categorical variable with $k > 2$ levels, you must create $k - 1$ dummy variables, with one level set as the baseline. Ordered categorical variables can be dealt with by treating them as continuous and using polynomials.

- There is no need to transform x variables to get approximate normality. However, you may wish to standardise them by subtracting the mean and dividing through by the standard deviation. This means that the coefficients you estimate represent the overall strength of the relationship between that x and y. However, to interpret the biological/clinical/political/etc significance of the effect requires working with the original scale.

4.2.2 Predictive distribution

If you wish to derive predictions of the possible outcome variable for future observations, with predictors known, this can trivially be done via the *posterior predictive distribution*:

$$\begin{aligned} \mathrm{p}(y_i^\star | x_i^\star, \{x_j, y_j\}) &= \int \mathrm{p}(y_i^\star | x_i^\star, \{x_j, y_j\}, b, \sigma) \mathrm{p}(b, \sigma | \{x_j, y_j\}) db\, d\sigma \\ &= \int \mathrm{p}(y_i^\star | x_i^\star, b, \sigma) \mathrm{p}(b, \sigma | \{x_j, y_j\}) db\, d\sigma. \end{aligned}$$

In other words, integrate over the posterior for the parameters, and for each, calculate the density of the new observation. This is easily done by sampling: taking the sample from the posterior and for each, simulating one or more values of $y_i^\star$.

The posterior predictive distribution (which, by the way, can be derived similarly for other problems, not just for regression) allows you to:

- do forecasting, e.g. in an auto-regressive model;
- create graphs for particular combinations of covariates;
- evaluate the fit of the model (see later);
- make decisions about optimal behaviour (see next chapter).

Note though that if you perform predictions for covariates that are wildly atypical of the data you fitted the model to, your predictions may not be reliable.

4.2.3 Example: deforestation of Pacific islands

When Jacob Roggeveen and his ship encountered Easter Island on Easter day, 1722CE, he was the first European to reach what was then the world's most isolated human habitation. There he discovered Polynesian islanders with no boats larger than a small canoe, who had constructed the innumerous, and iconic, massive stone heads (moai), the largest over 20m tall and weighing over 200 tonnes. When he landed, there were no trees at all on Easter Island, and the construction of the moai was a mystery. We now know that Easter Island had plentiful forests when it was first settled in around 900CE, and that by 1600CE all the trees had been felled, leaving the islanders unable to build boats or find wild food, probably leading to the population shrinking to about a quarter of its previous size. (Read more in Diamond (2005) Collapse, an excellent if thoroughly depressing story of the stupidity of man). Easter Island was not alone in suffering deforestation, and Rollett and Diamond (2004; *Nature* 431:443–6) have assessed the amount of deforestation across 69 Pacific islands, from Yap to Easter, and Hawai'i to New Zealand by carefully researching the first descriptions of the islands by the first European sailors, which should reflect the condition of the islands after the first Polynesians settled there but before European interference.

They scored the amount of deforestation on a five point scale, with 5 the worst, and also measured the island's own area, the area of all nearby islands, the distance to the nearest small, or large, island, the elevation, latitude, rainfall, age, the presence of makatea, the amount of dust from continental Asia, and the amount of volcanic ash (tephra)—for each island, or part of island for larger islands. The data are available in the supplementary information for Rollett and Diamond and are plotted in figure 4.6.

I fit a model with uniform priors on the co-efficients and standard deviation of the error terms. The R code is as follows:

```
source(’dataset.r’)
library(rjags)
initialisation=list(b=c(1,rep(0,10)),sigma=1)
jagmod=jags.model(’model.txt’,data=dataset,
                  inits=initialisation,n.chains=4)
update(jagmod,n.iter=1000,progress.bar=’text’)
posterior = coda.samples(jagmod, c("b","sigma"),
n.iter=25000, progress.bar="text",thin=1)
```

and the JAGS model file contains

```
model
{
  for(i in 1:79)
  {
    deforestation[i] ~ dnorm(mu[i],tau)
    mu[i] <- b[1] + b[2]*area_self[i] + b[2]*area_near[i]
           + b[3]*distance_nearest_small[i]
           + b[4]*distance_nearest_large[i] + b[5]*elevation[i]
           + b[6]*latitude[i] + b[7]*rainfall[i] + b[8]*age[i]
           + b[9]*makatea[i] + b[10]*dust[i] + b[11]*tephra[i]
  }
  for(i in 1:11){b[i]~dunif(-1000,1000)}
  tau <- 1/(sigma*sigma)
  sigma ~ dunif(0,1000)
}
```

The results are tabulated in table 4.3. Larger islands are at higher risk of deforestation (at about one deforestation unit (on the five point scale) per 40 000km^2), although having other islands nearby reduces the risk. Being further from the equator also increases the risk, at about one unit per 15 degrees latitude. Each meter of rainfall per year reduces the risk of deforestation, while older islands are at increased risk. We are not sure about the direction of effect of the other covariates.

Note that although the outcome variable takes values on $\{1, \dots, 5\}$ only, this is still a valid analysis to understand strength of effect, though the

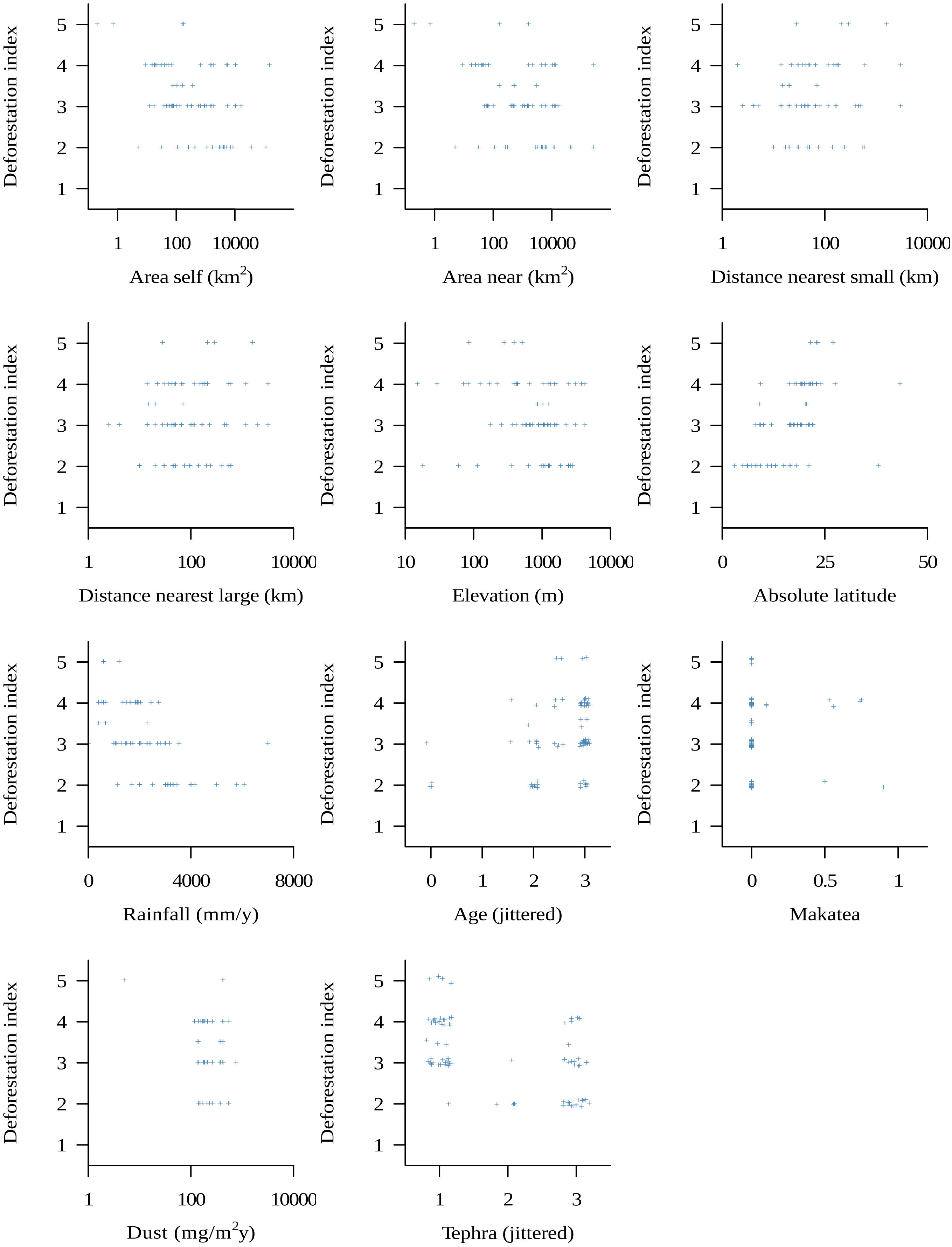

Figure 4.6: **Deforestation data.**

predictive distribution definitely does not match the actual distribution. A better mode of analysis would be a probit model, considered later in the course.

Table 4.3: **Results from island deforestation study.** Posterior means and equal tailed 95% intervals are tabulated (Mean, lower, higher) for the covariates considered by Rollett and Diamond—own area is the area of the island in question; nearby areas the area of islands within 50km; distance to the nearest small island is to one at least 25% the size of the island in question, to the nearest large island, it is to one at least 75% of that size (the latter has been set to 400km for New Caledonia) Latitude is the absolute latitude from the equator, ignoring direction. Island age uses an arbitrary scale. Makatea is the proportion of the island's surface with makatea, a raised coral that easily lacerates skin. Dust and tephra are the amount of fallout of dust from Asia and volcanic ash, respectively.

Variable	Mean	lower	upper	$p(>0)$
Intercept	1.2	0.25	2	–
Own area (1000km^2)	0.026	0.00037	0.053	0.98
Nearby areas (1000km^2)	-0.02	-0.033	-0.0066	0.0015
Distance nearest				
... small island (1000km)	0.24	-0.021	0.5	0.96
Distance nearest				
... large island (1000km)	-0.0011	-0.0022	-0.00013	0.014
Elevation (km)	-0.000035	-0.00022	0.00015	0.35
Latitude	0.073	0.051	0.096	1
Rainfall (m/y)	-0.23	-0.32	-0.13	0.00001
Island age				
... (arbitrary scale)	0.48	0.3	0.65	1
Makatea	0.24	-0.46	0.94	0.76
Dust (g/m^{2}y)	0.56	-0.32	1.4	0.9
Tephra	-0.054	-0.22	0.11	0.26
σ	0.48	0.4	0.57	–

4.3 Generalised linear models

Generalised linear models are, as their name suggests, generalisations of the linear regression model described in the previous section. We could write the previous model as

$$\begin{aligned} \mu_i &= b_0 + \sum_{k=1}^{K} b_k x_{ki} \\ y_i &\sim N(\mu_i, \sigma), \end{aligned}$$

that is, the mean for the ith outcome is a linear combination of predictors and the outcome has a distribution which is normal with additional spread around that mean. Generalised linear models replace μ_i by some other parameter governing centrality in the outcome distribution—usually involving a transformation to guarantee correct support. Examples include the Poisson model,

$$\begin{aligned} \log(\mu_i) &= b_0 + \sum_{k=1}^{K} b_k x_{ki} \\ y_i &\sim Po(\mu_i), \end{aligned}$$

and logistic regression model,

$$\begin{aligned} \log\left(\frac{p_i}{1-p_i}\right) &= b_0 + \sum_{k=1}^{K} b_k x_{ki} \\ y_i &\sim Bin(1, p_i) \end{aligned}$$

The advantages of these models is that the outcome variable is freed from the need to be Gaussian, and by ensuring a suitable transformation of the $b_0 + \sum_{k=1}^{K} b_k x_{ki}$ term, we can guarantee a legal configuration for all values of b. Both the Poisson and logistic models have an additional benefit: in the absence of interactions or non-linearities, they allow effects of covariates to be summarised in a single value that applies to all values of x, i.e. a relative risk for the Poisson model—from $\exp(b)$—and an odds ratio for the logistic—also from $\exp(b)$. Note that there is absolutely no reason why either of these models should be appropriate for any particular data set. The logistic regression model makes a useful but strong assumption that the odds ratio is constant for all starting values x—this may not be true. The Poisson model makes the strong assumption that the variance is equal to the mean, which is unlikely to be true (it will be true if the events are independent, but not if there are heterogeneous event rates).

For an unordered categorical variable with $M+1$ classes, a multinomial logistic model can be used. This has form:

$$\begin{aligned}
\log\left(\frac{p_{im}}{p_{i0}}\right) &= b_{0m} + \sum_{k=1}^{K} b_{km} x_{ki}, m \neq 0 \\
p_{im} = \mathrm{p}(y_i = m) &= \frac{\exp\left(b_{0m} + \sum_{k=1}^{K} b_{km} x_{ki}\right)}{1 + \sum_{j=1}^{M}\left[\exp\left(b_{0j} + \sum_{k=1}^{K} b_{kj} x_{ki}\right)\right]}, m \neq 0 \\
p_{i0} = \mathrm{p}(y_i = 0) &= \frac{1}{1 + \sum_{j=1}^{M}\left[\exp\left(b_{0j} + \sum_{k=1}^{K} b_{kj} x_{ki}\right)\right]}.
\end{aligned}$$

For an ordered categorical variable with $M+1$ levels, one might use an ordered probit model (it is actually possible to use an ordered logistic model but the probit is a bit easier). This involves introducing a latent, i.e. unobserved, variable z_i for each individual with

$$\begin{aligned}
\mu_i &= \sum_{k=1}^{K} b_k x_{ki} \\
z_i &\sim N(\mu_i, 1)
\end{aligned}$$

(the standard deviation being set to 1, and the intercept to 0, for identifiability) and some threshold variables θ_j with $\theta_j < \theta_k$ for all $j < k$. Then, the outcome variable $y_i = j$ if $z_i \in (\theta_{j-1}, \theta_j]$, with $\theta_0 = -\infty$ and $\theta_{M+1} = \infty$. We shall see an example of this kind of model a little later.

4.3.1 Separation

Imagine you observe the following data on the relationship between a symptom being used to try to diagnose a disease:

Disease	No symptom	Symptom
Absent	100	80
Present	0	20

On the face of it, this symptom is a fairly good indicator of disease that you would want to include in any clinical algorithm to diagnose the disease: of those without the symptom, none have the disease, while some of those with the symptom, do.

Now imagine doing a logistic regression (classically) on these data. In R, this can be done as follows:

```
 x=c(rep(0,100),rep(1,100))
 y=c(rep(0,180),rep(1,20))
 fit=glm(y~x,family='binomial')
 summary(fit)
```

The output says

```
Call:
glm(formula = y ~ x, family = "binomial")

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.66805  -0.66805  -0.00005  -0.00005   1.79412

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -20.57    1773.04  -0.012    0.991
x              19.18    1773.04   0.011    0.991

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 130.03  on 199  degrees of freedom
Residual deviance: 100.08  on 198  degrees of freedom
AIC: 104.08

Number of Fisher Scoring iterations: 19
```

In other words, the odds ratio for disease for those with the symptom relative to those without is exp(19.18) = 213 682 110 with 95% (Wald) confidence interval $\exp(19.18 \pm 1773.04) = (0, \infty)$ (according to my computer). Which, by the way, is statistically not 'significant' ($p = 0.99$).

This is the simplest example of a pathology called *separation*, which can also happen with continuous variables and with linear combinations of more than one variable. In this example it happens because the likelihood of the logistic model can be increased by increasing the coefficient *ad nauseum.* R stops in this case after 19 iterations of increasing the coefficient but other statistical packages may stop at different points, leading to different 'estimates'. All three of the maximum likelihood estimate, the standard error, and the Wald p-value are inappropriate.

What often happens in practice is that when faced with such a silly odds ratio, the user decides to discard the variable at fault, even though it may be an excellent predictor. More attractive alternatives usually involve penalising the likelihood, e.g. by maximising a function such as

$$g(\theta) = \mathrm{p}(\mathrm{data}|\theta) + \lambda \sum_i |\theta_i|$$

where λ penalises over-parameterised models and is often chosen via cross-validation (though that would not work here). See Tibshirani (1996 *J Roy Stat Soc B* 58:267–88) and subsequent papers. Other forms of penalties include Firth's method (Heinze & Schemper, 2002, *Stat Med* 21:2409–19).

Penalised likelihood approaches actually have a lot in common with Bayesian analyses, with the penalty term acting like a prior distribution centred on zero that 'pulls' the estimates down. For the hypothetical data above, one could fit a model with a normal prior for each parameter, e.g. with the following JAGS code:

```
model
{
  for(i in 1:200)
  {
    y[i]~dbin(p[i],1)
    logit(p[i])<-a+b*x[i]
  }
  a~dnorm(0,0.01)
  b~dnorm(0,0.01)
}
```

and

```
dataset=list(x=x,y=y)
initialisation=list(a=1,b=1)
jagmod=jags.model('model.txt',data=dataset,
                  inits=initialisation,n.chains=4)
update(jagmod,n.iter=1000,progress.bar='text')
posterior = coda.samples(jagmod, c("a","b"),
    n.iter=25000, progress.bar="text",thin=1)
```

This assigns a normal prior to b, the troublesome parameter, with mean 0 and variance 10^2. The resulting posterior mean for b is 8.6 (with 95%I 5.4–10.7), i.e. the still rather large odds ratio is 3 197 871 411. A $N(0, 1^2)$ prior gives a posterior mean OR of 5.4 with 95% uncertainty interval 2.0–12.1, though the justification for taking a prior with that variance as opposed to any other is hard to make.

4.3.2 Example 1: mining accidents

The first real example we shall consider relates to historic data on coal mining accidents in the UK (see Carlin et al, 1992, *Appl Stat* 41:389–405; and Jarrett, 1979, *Biometrika* 66:191–3). The data are the number of such accidents each year from 1851 to 1962. On inspecting these data (figure 4.7), it appears (i) there is no obvious relation between the number from one year to the next and (ii) there was a change in the number of accidents somewhere around the year 1900. It is difficult to set up a model using the standard glm function in R which allows a change in this rate, but easy Bayesianly.

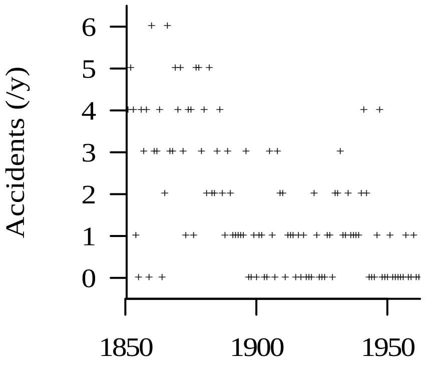

Figure 4.7: **Coal mining accidents in the UK from 1851 to 1962.**

The model I would like to fit to the data is as follows:

$$\begin{aligned} D_t &\sim Po(\lambda_t) \\ \lambda_t &= \mu_0 \mathbf{1}\{t \leq \tau\} + \mu_1 \mathbf{1}\{t > \tau\} \end{aligned}$$

where μ_0, μ_1 and τ are unknowns to be estimated. I will put non-informative priors on each ($\mu_0 \sim U(0, 100)$, $\mu_1 \sim U(0, 100)$ and $\tau \sim U(1851, 1962)$).

Here is some R code to fit the model:

```
data=list(
 N=112,
 D=c(4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,
     3,4,2,5,2,2,3,4,2,1,3,2,1,1,1,1,1,3,0,0,1,0,1,1,0,0,3,1,
     0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,2,3,1,1,
     2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,0),
 y=1851:1962)
```

```
logposterior=function(theta,data)
{
  mu=theta$mu0*(data$y<=theta$tau)+theta$mu1*(data$y>theta$tau)
  theta$logpost = sum(dpois(data$D,mu,log=TRUE))+
                  dunif(theta$tau,1851,1962,log=TRUE)+
                  dunif(theta$mu0,0,100,log=TRUE)+
                  dunif(theta$mu1,0,100,log=TRUE)
  theta
}
mh=function(theta,oldtheta,data)
{
  reject=FALSE
  if(theta$mu0<0)reject=TRUE
  if(theta$mu1<0)reject=TRUE
  if(theta$tau<=1851)reject=TRUE
  if(theta$tau>=1962)reject=TRUE
  if(!reject)
  {
    theta=logposterior(theta,data)
    logaccprob=theta$logpost - oldtheta$logpost
    lu=-rexp(1)
    if(lu>logaccprob)reject=TRUE
  }
  if(reject){theta=oldtheta}
  theta
}

set.seed(666)
library(mvtnorm)
theta=list(mu0=3.3,mu1=0.9,tau=1890)
SIGMA=matrix(c(0.0809, 0.0029,-0.1155,
               0.0029, 0.0129,-0.0530,
              -0.1155,-0.0530, 4.1241),3,3)
theta=logposterior(theta,data)
MCMCits=10000
storage=data.frame(mu0=rep(0,MCMCits),
                   mu1=rep(0,MCMCits),
                   tau=rep(0,MCMCits))
for(iteration in 1:MCMCits)
{
  if(iteration%%100==0)print(iteration)
```

```
    oldtheta=theta
    epsilon=rmvnorm(1,rep(0,3),SIGMA*0.5)
    theta$mu0=theta$mu0+epsilon[1]
    theta$mu1=theta$mu1+epsilon[2]
    theta$tau=theta$tau+epsilon[3]
    theta=mh(theta,oldtheta,data)
  storage$mu0[iteration]=theta$mu0
  storage$mu1[iteration]=theta$mu1
  storage$tau[iteration]=theta$tau
}
```

Posterior estimates are presented in figure 4.8. The posterior for τ, the changepoint, is bimodal. There is clear evidence for a change around 1890 (95%I 1886–94), from a mean annual rate of 3.2 (2.6–3.8) to 0.9 (0.7–1.1).

We can derive estimates of the yearly means, accounting for uncertainty in whether each year was in the first or second era, using the following code to derive posterior mean and 95% intervals (pointwise):

```
CI=array(0,c(data$N,3))
for(i in 1:data$N)
{
  mu=storage$mu0*(data$y[i]<=storage$tau)+
     storage$mu1*(data$y[i]>storage$tau)
  CI[i,c(1,3)]=quantile(mu,c(.025,.975))
  CI[i,2]=mean(mu)
}
```

These are plotted in figure 4.9.

4.3.3 Example 2: English association football

The next example also illustrates an advantage the Bayesian approach has over classical approaches, the ability readily to incorporate latent variables. We shall analyse results from the 2008–9 season in the English second division (a memorable season for Luton Town fans, who saw their team, the oldest professional football team in southern England, receive a fixed points penalty of 30 (equivalent to 10 wins) for entering and then 'improperly' leaving administration and were relegated as a result—see the unusually insightful analysis of Cunniffe and Cook (2009, *J Quant Anal Sports* 5:a5)). Games of association football are played between two teams, one playing at their 'home' stadium, the other 'away', with the team playing at 'home' thought to receive an advantage due to familiarity with the pitch and the greater

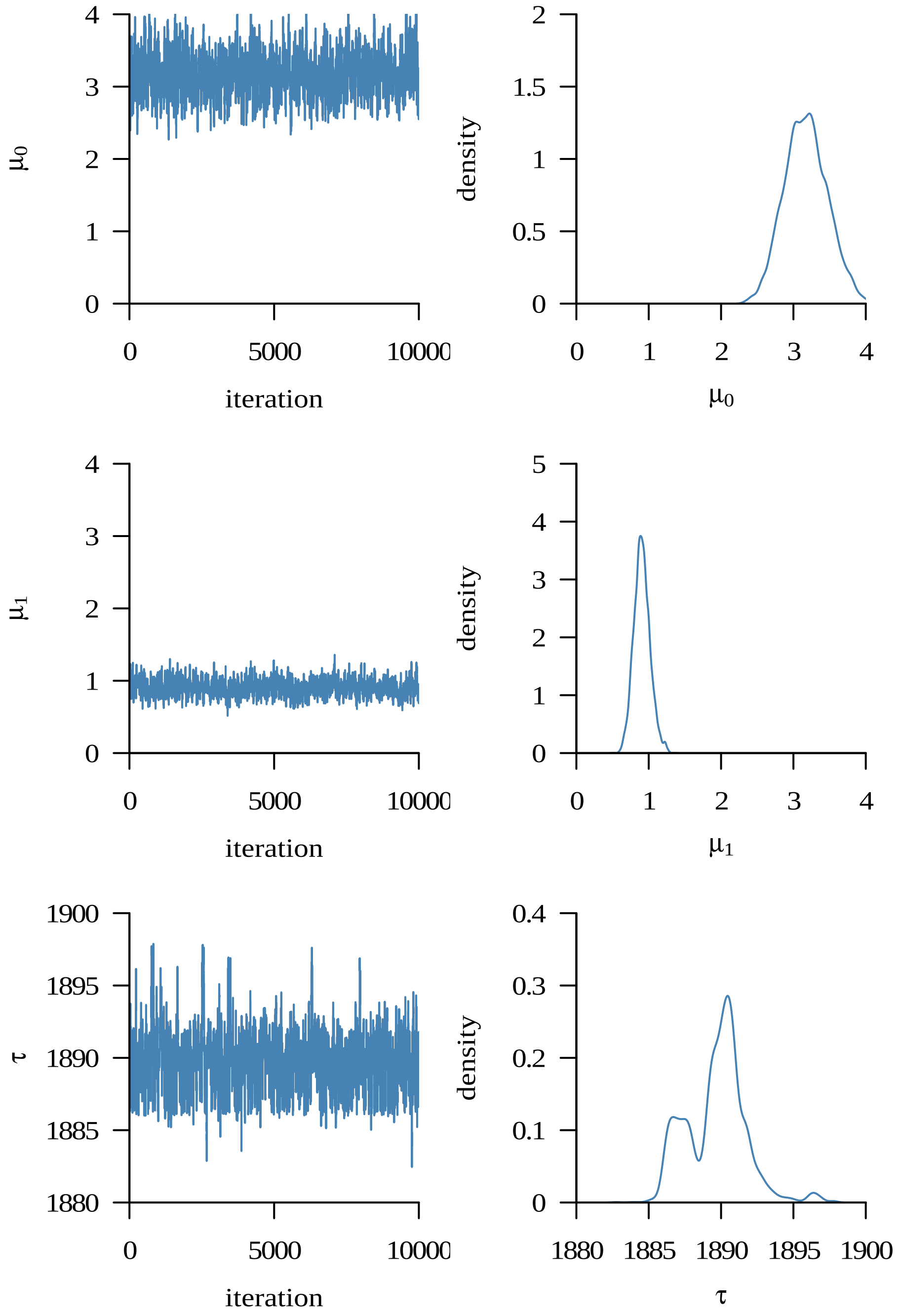

Figure 4.8: **Mining accident parameter estimates.** Traceplots of the mean annual number before the changepoint, after the changepoint, and of the changepoint itself, are left. Kernel density estimates of the three parameters are right. Note the bimodal distribution of τ.

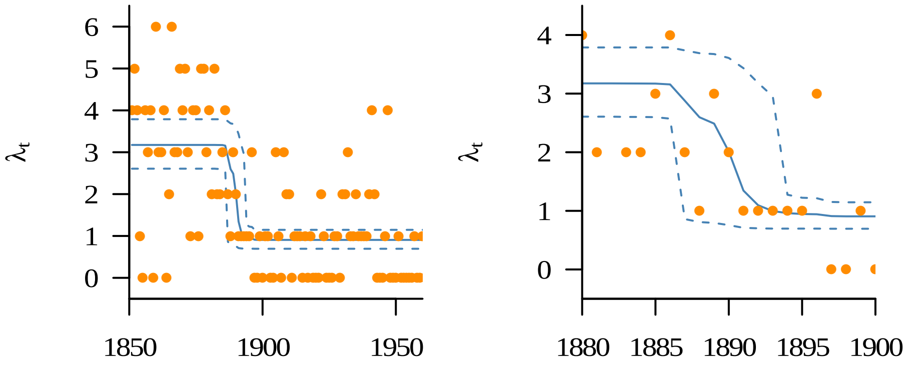

Figure 4.9: **Posterior distribution for annual risk of mining accident.** Posterior mean and 95% interval are plotted (blue) alongside the data (orange).

support from allied spectators. In the league format played in English association football, each pair of teams in a division play each other twice, once home and once away. At the end of each game, if one team has scored more goals than the other, it has won, and a draw occurs if they score the same number of goals.

In this analysis, we shall model the final outcome of games, with the notation $y_{ij} \in \{W, D, L\}$ denoting that the game between teams i and j, with i at home, resulted in a win (W), draw (D) or loss (L) for the home team, i (and hence a loss, draw or win for j, respectively). The analysis will account for two things:

- the asymmetry between playing at home and away;
- the perceived differences in quality between teams.

We will use a latent variable x_{ij} on the real line to represent how much better i played than j (with a high x_{ij} acting in i's favour), with μ_{ij} representing the overall propensity for i to beat j and $x_{ij} \sim N(\mu_{ij}, 1)$. Since there are three outcomes, with a win better than a draw better than a loss, we shall use two thresholds, θ_1 and θ_2, with i beating j if $x_{ij} > \theta_2$, drawing with j if $x_{ij} \in (\theta_1, \theta_2)$, and losing if $x_{ij} < \theta_1$.

We will take $\mu_{ij} = \beta_i - \beta_j$, thus allowing teams to differ in strength (but not in terms of the home advantage), with β_i assumed to be normal with mean 0 and standard deviation σ. Non-informative priors are taken for σ and θ.

The R code used follows:

```
source("/code/read.r")
source("/code/loglikelihood.r")
source("/code/initialise.r")

nits=10000;burnin=1000;thinprint=100;thin=10
parameters=logposterior(parameters,result)

stored.parameters=list(beta=array(0,c(nits,nteams)),
            betasd=array(0,c(nits)),theta=array(0,c(nits,2)))

for(nit in (-burnin+1):nits)
{
  if(nit%%thinprint==0)print(paste("Iteration",nit,"of",nits,
     "; logpost =",round(parameters$logpost,2),'; theta =',
     round(parameters$theta[1],2),round(parameters$theta[2],2)))
  for(thinning in 1:thin)
  {
    ##change to betas
    for(i in 1:nteams)
    {
      parametersold=parameters
      parameters$beta[i]=rnorm(1,parameters$beta[i],0.2)
      parameters=mh(parameters,parametersold,result)
    }
    ##change to betasd
      parametersold=parameters
      parameters$betasd=rnorm(1,parameters$betasd,0.02)
      parameters=mh(parameters,parametersold,result)
    ##change to thetas
    for(i in 1:2)
    {
      parametersold=parameters
      parameters$theta[i]=rnorm(1,parameters$theta[i],0.02)
      parameters=mh(parameters,parametersold,result)
    }
  }
  #dump to store
  if(nit>0)
  {
    stored.parameters$beta[nit,]=parameters$beta
    stored.parameters$betasd[nit]=parameters$betasd
    stored.parameters$theta[nit,]=parameters$theta
  }
}
source("/code/output.r")
source("/code/plots.r")
```

This calls additional functions, some of which are:

```
 #in loglikelihood.r
logposterior=function(p,result)#i home, j away
{
  mu=p$beta[result$homeid]-p$beta[result$awayid]
  cprobs=cbind(rep(0,length(result$awayid)),pnorm(p$theta[1],mu,1),
               pnorm(p$theta[2],mu,1),rep(1,length(result$awayid)))
  probs=cbind(cprobs[,2]-cprobs[,1],cprobs[,3]-cprobs[,2],
              cprobs[,4]-cprobs[,3])
  f=probs[,1]*(result$result==-1)+probs[,2]*(result$result==0)+
                                  probs[,3]*(result$result==1)
  p$logpost=sum(log(f))+sum(dnorm(p$beta,0,p$betasd,log=TRUE))+
            sum(dunif(p$theta,-100,100))
  p
}

mh=function(current,old,result)
{
  reject=FALSE
  if(current$theta[2]<current$theta[1])reject=TRUE
  if(current$betasd<0)reject=TRUE
  if(!reject)
  {
    current=logposterior(current,result)
    logaccprob=current$logpost-old$logpost
    lu=-rexp(1)
    if(lu>logaccprob)reject=TRUE
  }
  if(reject)current=old
  current
}

# in initialise.r
```

```
parameters=list(
  beta=rnorm(nteams,0,0.1),
  betasd=0.1,
  theta=c(-0.5,0.5)
)
```

The estimates of the β parameters for each team are presented in figure 4.10. The teams are effectively indistinguishable from each other—something that may appear surprising, and that may raise suspicions of a coding error, or a statistical artifact of the model used. We would therefore like to explore whether this model provides a good fit to the data before we trusted this finding. We shall consider model checking in the next section.

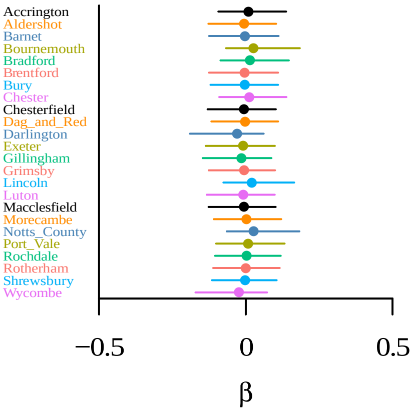

Figure 4.10: **Football team abilities for English second division in 2008–9**. Posterior mean and equal tailed 95% intervals plotted. 'Dag and Red' is short for Dagenham and Redbridge FC.

4.4 Model checking

Model checking is an integral component of statistical analysis, despite it often being treated as trivial compared to the more 'serious' work of estimating parameters given a model handed down from above. Model checking is covered in considerable detail in chapter 6 of Gelman et al (2004, Bayesian Data Analysis), from which I have drawn most of the ideas in this section.

Assessing convergence is unrelated to model checkiing. Convergence is a property of a numerical method, not of a model and data set. If, however, there is a bug in one's code, it may *appear* that the fitted model does not agree with the data, when in fact properly written code for the same model might yield a good fit. This scenario can be assessed with a simple trick: simulate a data set from the model being fitted, using realistic parameter values and sample sizes, then running the routine and checking the actual parameter values used are supported in the posterior. (This trick works also for frequentist routines.)

4.4.1 Priors

A model has two components: a model for the data given the parameters, and a model for the parameters. Checking the latter, the priors, is therefore one part of model checking. This can entail:

- Checking if the prior is consistent with the posterior. If it is not, it is worth reassessing the information that went into the priors since it clearly clashes with that of the data. Note that the prior being consistent with the posterior means they overlap, not that they are equal (which one would not expect, and which may indicate a problem in the coding).

- If a non-informative prior were used, a sensitivity analysis to the exact form selected could be undertaken. What happens if instead of the $N(0, 100^2)$ distribution you originally used, you tried a $U(-100, 100)$ instead? Probably very little change, in which case you can comfort yourself in the knowledge that your results are robust to the exact formulation of the prior.

- If an informative prior were used (using external data, please!), a sensitivity analysis to the exact formulation might be undertaken. What if instead of assuming a normal distribution with mean μ and variance σ^2, a log-normal with the same moments were used? Again, probably there would be little change.

It is quite rare that this prior check will reveal a problem since the final posterior is usually robust to small changes in the prior if it is non-informative, and if you have properly justified the choice of external data then the only question is how, precisely, they form the prior.

4.4.2 Likelihoods

Model checking is more vital in the model for the data, since it is here that inferences are most susceptible to the assumptions used. *All* analyses make assumptions about the model for the data, even non-parametric, frequentist methods such as Kruskal–Wallis ANOVA or Kaplan–Meier estimates of survival (which depend—strongly—on independence between sampling units). The question thus should be posed: how much do our conclusions about this data depend on the assumptions/model used to reach those conclusions?

The strongest way to validate a model is via external data. If you develop a model, and fit it to past data, and make some sort of falsifiable prediction based on that (and not too many—like the magician who put in his pocket a note each morning saying “I will die today” and the date), and then go out and perform the additional experiments or data collection needed to test the prediction, and the future data agree with what the model predicted—that is quite convincing evidence in favour of the model used.

For example, Ong et al (2010, *PLOS One* 5:e10036), during the 2009 influenza A-H1N1 pandemic, set up a network of general practitioners in Singapore who provided daily data on the progress of the epidemic. We fitted a complex stochastic process model to the data in real time and used the model to predict the future time course, including the time of the peak. These predictions were placed daily on the web for the world to see (and they did attract hits from all over the world) and we compared past predictions against eventual data. The model and routine (we used sequential importance sampling) predicted the peak to within one week, thereby providing strong evidence to support the model assumptions and validating its use in planning by various organisations.

Usually, however, we have a set of historic data and no future data will become available, in which case, external validation is not possible and a weaker form of validation—internal—must be done instead. To do this, we typically derive the posterior predictive distribution of new “data” and compare characteristics of these to those of the real data.

To do this, one must define one or more test quantities (similar to test statistics in classical null hypothesis significance testing). We then assess the posterior distribution of the test quantities given the data—which, you’ll note, is different from the distribution of test statistics given a null hypothesis. The choice of test quantities depends on the model and data and your cunning, as can be best illustrated with examples.

4.4.3 Example 1: Coin tosses and fake coin tosses

Here is a somewhat fun parlour trick. First, create 100 Bernoulli random variables with a probability 50% of being heads, and otherwise tails (or, better still, have an audience toss 100 coins—perhaps 10 people each doing 10—and record them. If you are interested in tossing coins, you might read Diaconis et al, 2007, *SIAM Review* 49:211–35; and Gelman and Nolan, 2002, *Am Stat* 56:308–11). Then, have your audience create a synthetic series of coin tosses in an attempt to fool you into believing they are real. Your challenge is to guess correctly which is the real and which the synthetic series of tosses. I have simulated 100 tosses on my computer, and requested my wife create 100 "tosses" to try to trick you. The 200 tosses are presented below.

```
TTHTHHTHHH    HHHTTTHTHH
THTTHTTHTH    THHHHHHHHT
TTHHTHHTHT    THTTTTHTTH
HTHTTTHHTH    THHTTHHHTT
THHHHTTHTH    TTHHTHHHHH
HHTHHTHHTT    TTTTTHTHTH
THHHTHTHTT    TTTHHHTTTT
HHTHTHHTTT    HTTTTTHTHH
HHTHHTHHTH    THTTHHTTTH
THHTHHTHTT    THTTHHHTTH
```

We shall assume a model for both, in which tosses are Bernoulli(p), with $p = 50\%$ and no inference for the p, since it is not possible to bias a coin without special machinery by more than 1%. We will consider two test statistics

- T_1, the number of switches from H to T or T to H;
- T_2, the maximum run of consecutive Hs or Ts.

These were selected as the assumption of independence should lead to a characteristic distribution of switching from one face to the other, which may not be reflected by a human impostor who may be using the representativeness heuristic (Tversky and Kahneman, 1974, *Science* 185:1124–31).

The joint distribution of T_1 and T_2 are easily obtained via simulation and plotted as follows:

```
testquantities=function(series)
{
  T1=0
  for(k in 2:length(series))
  {
    if(series[k]!=series[k-1])T1=T1+1
  }
  t2=0*series
  for(k in 1:length(series))
  {
    for(j in k:length(series))
    {
      if(series[j]==series[k])t2[k]=t2[k]+1
      if(series[j]!=series[k])break()
    }
  }
  T2=max(t2)
  c(T1,T2)
}

NSIM=10000
tqsim=array(0,c(NSIM,2))
for(s in 1:NSIM)
{
  if(s%%1000==0)print(s)
  simu=rbinom(100,1,0.5)
  tqsim[s,]=testquantities(simu)
}
tqwife=testquantities(wife)
tqreal=testquantities(real)

library(KernSmooth)
de=bkde2D(tqsim,c(1,1))
image(de$x1,de$x2,de$fhat,xlab='T1: number switches',
      ylab='T2: length of runs',col=gray(seq(1,0,-1/12)))
points(tqreal[1],tqreal[2],col=2,pch='R',cex=2)
points(tqwife[1],tqwife[2],col=2,pch='S',cex=2)
```

The plot makes clear that the synthetic one created by a non-statistician has too many switches and too short runs. You should be able to work out which of the two sets was created by my wife now.

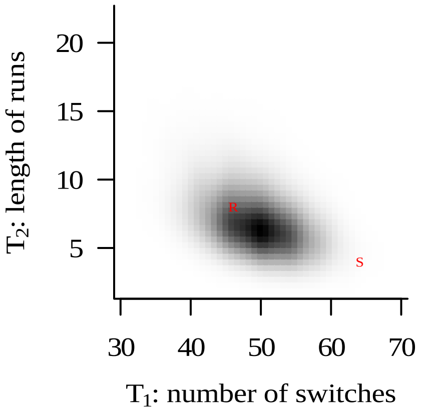

Figure 4.11: **Model checking for coins example**. The joint distribution of two test quantities are plotted along with the real (R) and synthetic (S) values.

4.4.4 Example 2: coal mining accidents

In the analysis of coal mining accidents, we made two potentially strong assumptions.

- We assumed the distribution of accidents be Poisson. This means the variance equals the mean. If we have neglected some important heterogeneity, the variance could be greater.

- We assumed independence from one year to the next, conditional on the parameters.

To test the former, let us define T_0 to be the ratio of sample variance from 1851 to 1885 (i.e. before the inferred change point) to the sample mean, and T_1 be the same but for 1900 to 1962. These we can calculate from the data trivially. To determine their distribution from the model fit, we can simulate a data set for each draw from the posterior and calculate these, then take the overall distribution. Again, we will plot the joint distribution. The code follows.

```
tf=c(which(data$y==1851),which(data$y==1885),
     which(data$y==1900),which(data$y==1962))
T0data=var(data$D[tf[1]:tf[2]])/mean(data$D[tf[1]:tf[2]])
T1data=var(data$D[tf[3]:tf[4]])/mean(data$D[tf[3]:tf[4]])

T0model=T1model=rep(0,MCMCits)
```

```
for(iteration in 1:MCMCits)
{
  if(iteration%%100==0)print(iteration)
  theta=storage[iteration,]
  mu=theta$mu0*(data$y<=theta$tau)+theta$mu1*(data$y>theta$tau)
  sim=rpois(length(mu),mu)
  T0model[iteration]=var(sim[tf[1]:tf[2]])/mean(sim[tf[1]:tf[2]])
  T1model[iteration]=var(sim[tf[3]:tf[4]])/mean(sim[tf[3]:tf[4]])
}

library(KernSmooth)
de=bkde2D(cbind(T0model,T1model),c(.05,.05))
image(de$x1,de$x2,de$fhat,xlab=expression(T[1]*': early COV'),
      ylab=expression(T[2]*': late COV'),col=gray(seq(1,0,-1/12)))
points(T0data,T1data,col=2,pch='+',cex=2)
```

The graph is presented below. Note that the predicted test quantities are consistent with the data.

To test the independence assumption, we might quantify the auto-correlation in the time series (note that some auto-correlation is expected even with independence due to the change in means). This is very easy to do:

```
T2data=cor(data$D[-1],data$D[-data$N])
## in for loop:
T2model[iteration]=cor(sim[-1],sim[-data$N])
```

The observed correlation is perfectly consistent with what the model predicts.

4.4.5 Example 3: association football results

For the previous football example, we found that there was little evidence of differences in ability between teams, a surprising result to football fans. To test this, let's simulate league results under the fitted model and then assess the spread in final points (teams receive 3 for a win and 1 for a draw) under the data and fitted model. We will first draw samples from a non-hierarchical model with the sampled thresholds θ, i.e. a model with home effects only. We will then use the posterior samples of β_i—which applies for the whole season—to generate a sample of results.

We shall make the test quantity the kernel density estimate of the distribution of points, and perform a small number of simulations, plotting the

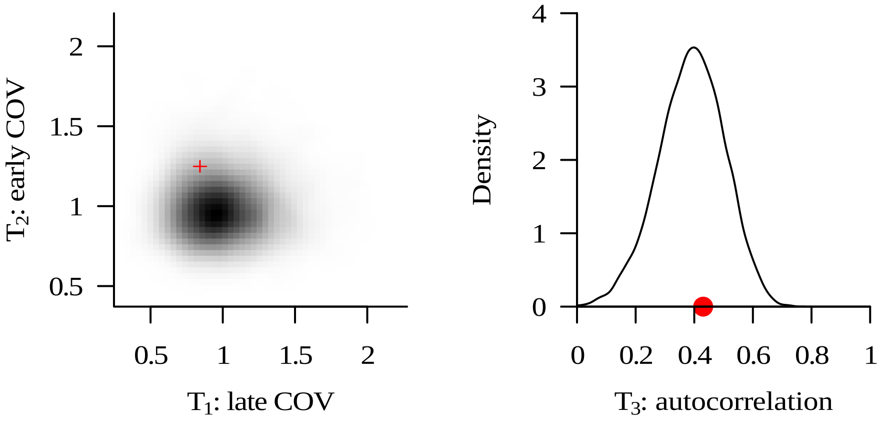

Figure 4.12: **Model checking for coal mining accidents example**. The joint distribution of two test coefficients of variation are plotted along with the real values (left); the third is plotted right.

densities for each run. Here is the code to do so (the version with no random effects has `beta=rep(0,nteams)`:

```
pointsdata=rep(0,nteams)
for(k in 1:length(result$result))
{
  i=result$homeid[k]
  j=result$awayid[k]
  if(result$result[k]==1){pointsdata[i]=pointsdata[i]+3}
  if(result$result[k]==0){pointsdata[i]=pointsdata[i]+1;
                          pointsdata[j]=pointsdata[j]+1}
  if(result$result[k]==-1){pointsdata[j]=pointsdata[j]+3}
}

set.seed(78634)
NSIM=20
points=array(0,c(NSIM,nteams))

for(s in 1:NSIM)
{
  index=sample(1:nits,1)
  theta=stored.parameters$theta[index,]
  beta=stored.parameters$beta[index,]
  results=array(0,c(nteams,nteams))
  for(i in 1:nteams)
  {
    for(j in (1:nteams)[-i])
    {
      z=rnorm(1,beta[i]-beta[j],1)
      if(z<theta[1])results[i,j]=-1
      else if(z<theta[2])results[i,j]=0
      else results[i,j]=1
    }
  }
  for(i in 1:nteams)
  {
    home=results[i,-i]
    away=-results[-i,i]
    points[s,i]=3*sum(away==1)+1*sum(away==0)+
                3*sum(home==1)+1*sum(home==0)
  }
}
```

Both plots suggest the data are indeed consistent with the fitted model, and that the variability in wins–draws–losses, and of final league positions, is consistent with pure chance.

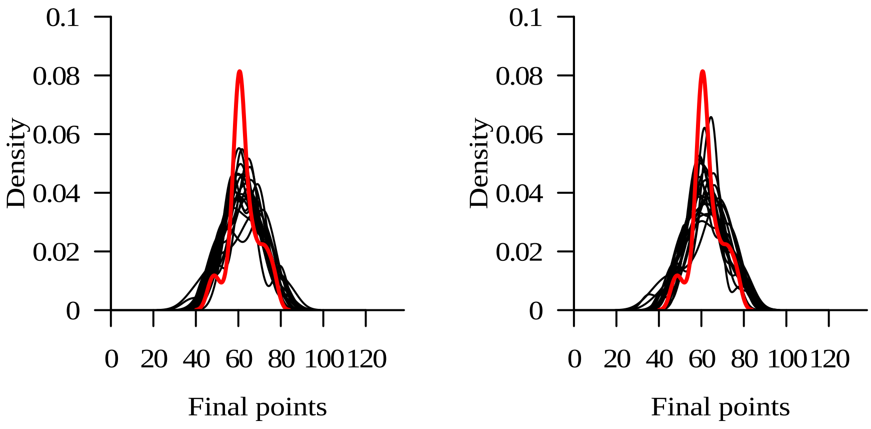

Figure 4.13: **Model checking for association football example**. The distribution of final points, as estimated with a kernel density estimate, are plotted along with the density estimate from the real data.

4.4.6 Example 4: leukaemia remission

The final example comes from an important early study of leukaemia treatment (which led to major successes against this cancer), previous considered in a tutorial, by Freireich et al (1963, *Blood* 21:699–716). In this study, young children with leukæmia were randomised to either the treatment arm or a placebo, and the time to relapse with the disease was recorded. These data are:

Placebo: 1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12, 15, 17, 22, 23

Treatment: 6, 6, 6, 7, 10, 13, 16, 22, 23, 6+, 9+, 10+, 11+, 17+, 19+, 20+, 25+, 32+, 32+, 34+, 35+

where a "+" symbol indicates right censoring (e.g. 6+ means the relapse occurred at some unspecified point after the 6th day). Although the patients all died eventually, this study led to better treatments and had a massive effect on future research into leukæmia treatments.

We will fit a Weibull model to both datasets, with the likelihood being the product of densities for the placebo arm and a product of densities and survival functions for the treatment arm:

$$\begin{aligned} \mathrm{p}(t_1^p,\ldots,t_{21}^p|k,\lambda) &= \prod_{i=1}^{21} k\lambda\left(\lambda t_i^p\right)^{k-1}\exp\left\{-(t_i^p\lambda)^k\right\} \\ \mathrm{p}(t_1^t,\ldots,t_{21}^t|k,\lambda) &= \prod_{i=1}^{9} k\lambda\left(t_i^t\lambda\right)^{k-1}\exp\left\{-(t_i^t\lambda)^k\right\}\times\prod_{i=10}^{21}\exp\left\{-(t_i^t\lambda)^k\right\} \end{aligned}$$

We can fit the model very simply in R (code not provided). To test whether a Weibull is a good enough fit, we can compare the Kaplan–Meier estimates of survival (see e.g. Hosmer et al, 2008, Applied Survival Analysis) for the data against the posterior predictive distribution of the survival function.

```
times=seq(0,35,0.1)
CIt=array(0,c(3,length(times)))
CIp=array(0,c(3,length(times)))
for(i in 1:length(times))
{
  st=pweibull(times[i],storage$kt,1/storage$lt,lower.tail=FALSE)
  sp=pweibull(times[i],storage$kp,1/storage$lp,lower.tail=FALSE)
  CIt[,i]=quantile(st,c(.025,.5,0.975))
  CIp[,i]=quantile(sp,c(.025,.5,0.975))
}
```

```
library(survival)
tp=c(1,1,2,2,3,4,4,5,5,8,8,8,8,11,11,12,12,15,17,22,23)
tt=c(6,6,6,7,10,13,16,22,23,6,9,10,11,17,19,20,25,32,32,34,35)
ct=c(rep(0,9),rep(1,12))
kmp=survfit(Surv(tp)~1)
kmt=survfit(Surv(tt,ct)~1)
```

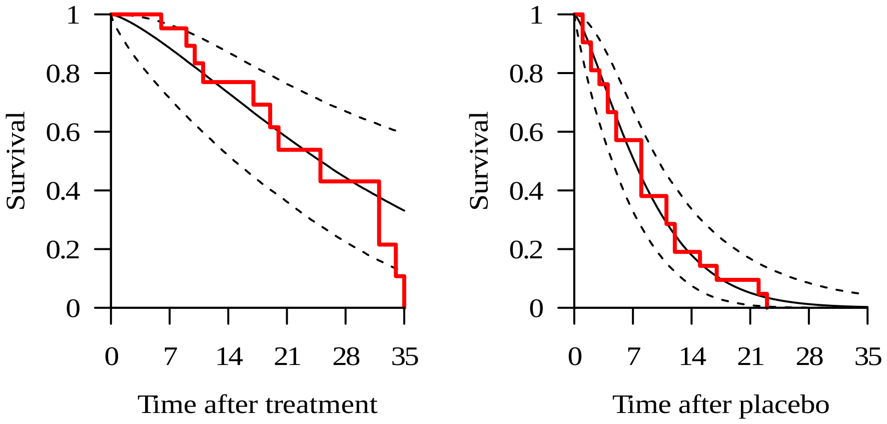

Figure 4.14: **Model checking for leukaemia example**. The distribution of survival functions (posterior median and 95% interval, black), are plotted along with the Kaplan–Meier estimate from the data (red).

4.5 Model comparison

Often, there is not a single, clear choice of model: several are possible explanations of the data, and even after checking model fit via the posterior predictive distribution, we may still have multiple models that provide a satisfactory fit. If the models illustrate different facets of the data, you might report them all, but if they all do much the same thing, how do you decide which to report as the main analysis?

There are two popular choices in classical statistics to select between models. One, the likelihood ratio test, is valid only if the models are nested (for example, a linear regression is nested within a quadratic, an exponential model within a gamma). If the models are M_1 (the smaller one) and M_2 (the larger), with parameters θ and (θ, ϕ), respectively, and $L_1 = 2\max_\theta \mathrm{p}(data|\theta, M_1)$ and $L_2 = 2\max_{(\theta,\phi)} \mathrm{p}(data|\theta, \phi, M_2)$ then the likelihood ratio test statistic, $L_2 - L_1$ should be χ^2 with $\dim(\phi)$ degrees of freedom, if in fact the smaller model is true (and with an unknown distribution that takes higher values if the larger is).

The second is Akaike's Information Criterion (AIC), which is defined to be, for model M_k with parameters θ_k,

$$AIC_k = -2\max_\theta \mathrm{p}(data|\theta_k, M_k) + 2\dim(\theta_k).$$

Models are compared using the absolute difference in AIC scores: if two models have scores within 2 of each other, they are considered indistinguishable in terms of their fit; if the difference is within 7 (or 10, depending on the authority), the one with lower AIC is clearly better than the other; while if the difference is greater, the fit is overwhelmingly better for the one with the lower AIC. (To see the rationale for these thresholds, consider two nested models where the larger model has one more parameter and the difference in AIC scores is 2: this corresponds to a likelihood ratio test statistic of 4 on 1 degree of freedom, i.e. a p-value of just about 5%. If the difference in AIC scores were 7, the p-value would be less than 1%, if 10, the p-value would be less than 0.1%. Note, though, that this argument is only valid for nested models and depends on the asymptotic distribution of the test statistic and maximum likelihood estimators; also, it naturally depends on the appropriateness of null hypothesis significance testing, which seems a bit silly in this context.)

There are also Bayesian analogues of these measures. The Deviance Information Criterion (DIC) is similar to the AIC, and is readily obtained from MCMC output. The posterior model probability is a more direct measure of the support for any model but is more challenging to derive. They are covered in the following sections.

4.5.1 Deviance information criterion

The *deviance* of a model for a dataset is -2 times the log-likelihood (the maximum deviance therefore has a role in both AIC and likelihood ratio test). The DIC is similar to the AIC in that it is the sum of two terms: one coming from the likelihood, and thus indicating concordance with the data, and the other from something like the size of the parameter vector. The overall formula is:

$$DIC = D_{\text{avg}} + p_d$$

where

$$D_{\text{avg}} = \int -2\log\{\mathrm{p}(data|\theta)\}\mathrm{p}(\theta|data)\,\mathrm{d}\theta,$$

i.e. is the posterior mean deviance—which can be derived simply for an MCMC sample by storing and then averaging the log likelihoods and multiplying by -2—and

$$p_d = D_{\text{avg}} + 2\log\{\mathrm{p}(data|\hat{\theta})\},$$

the *effective number of parameters*. The latter uses the difference between the average deviance and the deviance at a point estimate (such as the posterior mean), denoted $\hat{\theta}$.

The DIC is designed to work not with the actual, but the effective, number of parameters, to make it more useable for hierarchical models. Consider a model in which $\theta_i \sim N(\mu, \sigma)$ and θ_i influences the distribution of x_i. If i takes values from 1 to n, then there are $n+2$ actual parameters in the model (the θ_is, μ, and σ). However, if σ is very small, then all the θ_is are very close to μ, so μ acts as if it were the only parameter in the model. The effective number of parameters would therefore be closer to one than n.

Spiegelhalter et al (REF) suggest treating the difference in DICs as being on the same scale as the AIC, albeit noting that there are potential sampling errors in DIC, due to its estimate coming from an MCMC sample.

The recipe to calculate the DIC is to

- run an MCMC sampler, storing the loglikelihood;
- calculate a point estimate of the parameters, and the loglikelihood at that value;
- calculate the effective number of parameters (which you should check to ensure it is sensible) and then the DIC;
- repeat for one or more other models.

Note that as with the AIC, the DIC tells you nothing about absolute goodness of fit, only relative goodness of fit, and it allows this by the absolute difference in scores, not the relative.

4.5.2 Posterior model probability

A more complex, if satisfying, method of selecting between models derives the posterior probability that model M_k generated the data,

$$\mathrm{p}(\tilde{M} = M_k | data) \tag{4.1}$$

on the assumption that the data come from one of models $M_1, \ldots, M_K$. Although I agree with Box' dictum that all models are wrong (and some useful), I find it useful to postulate that one of a set of models is actually correct to weigh the evidence in favour of the models. This probability is derived by conditioning on the parameters and integrating them out. In practice, this is easiest to do by building an MCMC sampler that generates samples from the space of models and parameters, i.e. proposes switches between models and changes within models.

Two things make this difficult:

- one needs to be able to propose jumps from one parameter configuration in model i to another parameter configuration in model j, when the interpretation of those parameters may be radically different;
- some clever proposals involve deterministic functions of the parameters and auxilliary variables introduced to ensure reversibility, and to get the reversibility conditions correct involves working with Jacobians.

We will not consider problems that require so-called reversible jump MCMC (i.e. the latter point) as, to be honest, it is rather hard, but the reader is invited to refer to Green's seminal work on the topic (REF).

One way reversible jump MCMC can be avoided is to run one model at a time, first, and a reasonable estimate of the posterior derived (e.g. multivariate normal with mean and covariance taken from the pilot run). Then, in the main analysis, propose changes to the model, and at the same time, propose the parameters of the new model from this approximate distribution, noting that the proposals will not cancel under this approach. Proposals to the parameters using standard symmetric distributions such as Gaussians can follow if desired, and may improve mixing.

Evaluating the posterior model probability is harder than getting the DIC, but does bring with it one additional benefit: one can derive estimates that average over models. If you are interested in a quantity α that can be evaluated for each model, then the posterior distribution of α,

$$\mathrm{p}(\alpha|data) = \sum_m \mathrm{p}(\alpha|data, m)\mathrm{p}(m|data),$$

can be obtained trivially by evaluating it for each iteration of the MCMC sampler.

We shall see examples of posterior model probabilities and model averaging, and the DIC, in the next section.

4.5.3 Example 1: leukaemia remission

We consider the leukaemia chemotherapy trial by Freireich et al again. In this analysis, we shall consider four models for the remission times: Weibull, gamma, log-normal and exponential, of which the first three have two parameters each and the fourth one. The Weibull, gamma and log-normal allow very similar shapes and so it would be surprising if the data from 21 patients in each arm could distinguish them well.

Although it is quite possible to obtain DIC estimates within JAGS or BUGS (REFS), as I will be implementing the between-model proposals in R, it makes sense to use R for the DIC as well. I start by logging all parameters (except the means for the log-normal) to get parameters with support on all parts of the real line. This also, I found, standarised the spread of the posteriors, facilitating jumps between models later.

To begin with, I created a function to evaluate the logposterior and its components for any model, the model being specified in `p$m`:

```
logposterior=function(p)
{
  tp=c(1,1,2,2,3,4,4,5,5,8,8,8,8,11,11,12,12,15,17,22,23)
  tt=c(6,6,6,7,10,13,16,22,23,6,9,10,11,17,19,20,25,32,32,34,35)
  ct=c(rep(0,9),rep(1,12))
  if(p$m=='weibull')
  {
    p$logprior=dunif(p$kp,0,100000)+dunif(p$lp,0,100000)+
               dunif(p$kt,0,100000)+dunif(p$lt,0,100000)
    p$loglikelihood=sum(dweibull(tp,exp(p$kp),exp(p$lp),log=TRUE))+
               sum(dweibull(tt[ct==0],exp(p$kt),exp(p$lt),log=TRUE))+
               sum(pweibull(tt[ct==1],exp(p$kt),exp(p$lt),log=TRUE,
                   lower.tail=FALSE))
    p$logposterior=p$loglikelihood+p$logprior
  }
  if(p$m=='gamma')
  {
    ##etc
  }
  p
}
```

The following simple function performs the Metropolis–Hastings step, for the code with no jumps between models:

```
 mh=function(p,o)
{
  reject=FALSE
  p=logposterior(p)
  la=p$logposterior-o$logposterior
  lu= -rexp(1)
  if(lu>la)reject=TRUE
  if(reject)p=o
  p
}
```

The code to run the MCMC sampler was somewhat wasteful, but simple to run:

```
for(m in c('exponential','lognormal','weibull','gamma'))
{
  p=list(kp=0,kt=0,lp=3,lt=4,ap=0.5,bp=-3,at=0.5,bt=-3,
         mp=1.8,mt=3.7,sp=0,st=0,rp=-3.5,rt=-3.5,m=m)
  p=logposterior(p)
  mcmcits=10000
  storage=data.frame(kp=rep(0,mcmcits),kt=rep(0,mcmcits),...etc...,
                     logposterior=rep(0,mcmcits),
                     logprior=rep(0,mcmcits),
                     loglikelihood=rep(0,mcmcits))
  for(it in 1:mcmcits)
  {
    if(it%%100==0)print(paste(m,it))
    if(p$m=='lognormal')
    {
      o=p;p$mp=rnorm(1,p$mp,0.25);p=mh(p,o)
      o=p;p$mt=rnorm(1,p$mt,0.25);p=mh(p,o)
      o=p;p$sp=rnorm(1,p$sp,0.5);p=mh(p,o)
      o=p;p$st=rnorm(1,p$st,0.5);p=mh(p,o)
    }
    if(p$m=='gamma')
    {
      o=p;p$ap=rnorm(1,p$ap,0.5);p=mh(p,o)
      o=p;p$at=rnorm(1,p$at,0.5);p=mh(p,o)
      o=p;p$bp=rnorm(1,p$bp,0.5);p=mh(p,o)
      o=p;p$bt=rnorm(1,p$bt,0.5);p=mh(p,o)
    }
    if(p$m=='weibull')
    {
      o=p;p$kp=rnorm(1,p$kp,0.25);p=mh(p,o)
      o=p;p$kt=rnorm(1,p$kt,0.25);p=mh(p,o)
      o=p;p$lp=rnorm(1,p$lp,0.25);p=mh(p,o)
      o=p;p$lt=rnorm(1,p$lt,0.25);p=mh(p,o)
    }
    if(p$m=='exponential')
    {
      o=p;p$rp=rnorm(1,p$rp,0.25);p=mh(p,o)
      o=p;p$rt=rnorm(1,p$rt,0.25);p=mh(p,o)
    }
    storage[it,]=c(p$kp,p$kt,p$lp,p$lt,p$ap,p$at,p$bp,p$bt,
                   p$mp,p$mt,p$sp,p$st,p$rp,p$rt,p$logposterior,
                   p$logprior,p$loglikelihood)
  }
  if(p$m=='exponential')dump('storage','outE.r')
  if(p$m=='lognormal')dump('storage','outL.r')
  if(p$m=='weibull')dump('storage','outW.r')
  if(p$m=='gamma')dump('storage','outG.r')
}
```

This initialises values for parameters that are not used and stores their values, but only proposes changes to parameters used in the model being fitted.

The posterior median and 95%I for the survival function is plotted in figure 4.15.

The DIC is calculated thus:

```
DavgE=mean(-2*pE$loglikelihood)
DavgG=mean(-2*pG$loglikelihood)
DavgL=mean(-2*pL$loglikelihood)
DavgW=mean(-2*pW$loglikelihood)
```

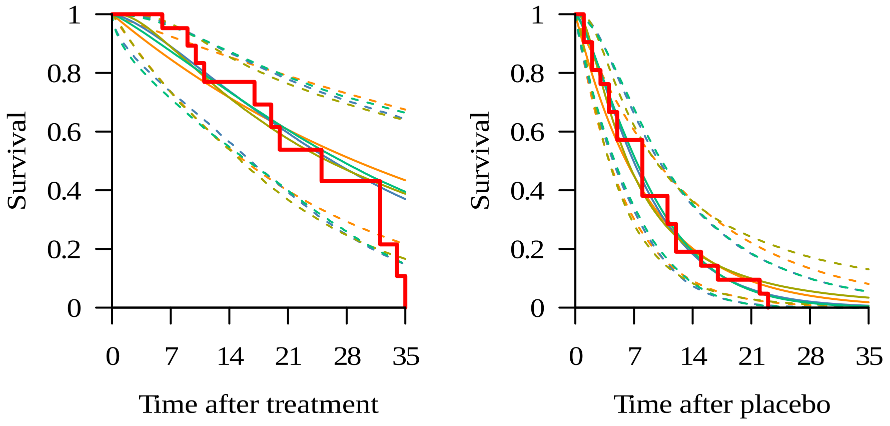

Figure 4.15: **Model checking for leukaemia example**. The distribution of survival functions (posterior median and 95% interval, orange [exponential], blue [gamma], olive [log-normal], and green [Weibull]), are plotted along with the Kaplan–Meier estimate from the data (red). The treatment arm is plotted left, the plabeco, right.

```
phat=list(kp=mean(pW$kp),kt=mean(pW$kt),lp=mean(pW$lp),
          ...etc...,rt=mean(pE$rt),m='exponential')
phat=logposterior(phat)
pdE=DavgE+2*phat$loglikelihood
# etc
```

The DIC scores are all less than half a unit from each other (exponential 221.12, gamma 221.41, log-normal 221.61, Weibull 221.47), suggesting the four models do as good a job at describing the data as each other. Note, though, that the DIC does use an asymptotic argument which is probably invalid for a sample size of 21 on each arm.

To derive the posterior probability for each of the models, I derive (multi) variate normal proposals for the parameters using the previous run, by defining a list of moments:

```
between=list( mu_tE=mean(pE$rt),
              sigma_tE=sd(pE$rt),
              mu_pE=mean(pE$rp),
              sigma_pE=sd(pE$rp),
              MU_tG=c(mean(pG$at),mean(pG$bt)),
              SIGMA_tG=cov(cbind(pG$at,pG$bt)),
              MU_pG=c(mean(pG$ap),mean(pG$bp)),
              SIGMA_pG=cov(cbind(pG$ap,pG$bp)),
              ...etc...
            )
```

These are plotted in figure 4.17 alongside the actual draws, indicating that the (multivariate) normal distributions used provide reasonable descriptions of the posteriors.

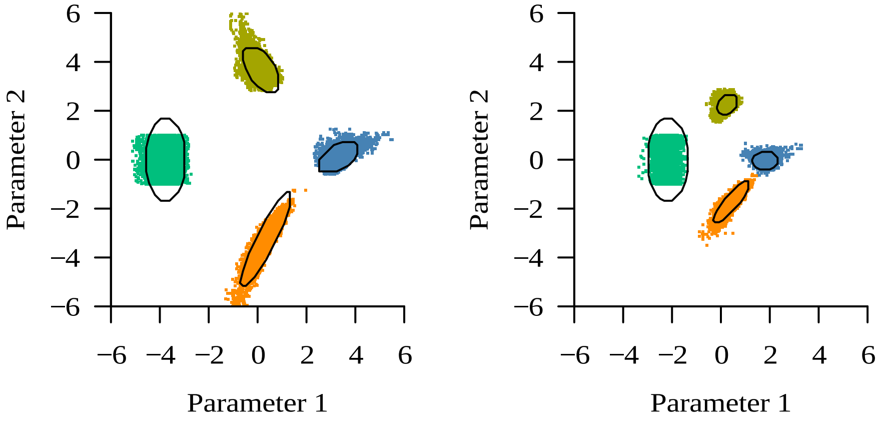

Figure 4.16: **Between model proposal distribution for parameters**. The posterior samples are indicated by scatter plots (orange [exponential], blue [gamma], olive [log-normal], and green [Weibull]), and the 95% highest density region for the normal approximations is overlaid in black. The exponential model has but a single parameter and so an arbitrary second parameter was simulated from a uniform distribution for this plot. The treatment arm is plotted left, the plabeco, right.

A Metropolis–Hastings function designed to handle between model moves follows:

```
betweenmh=function(p,o)
{
  reject=FALSE
  if(!reject)
  {
    p=logposterior(p)
    if(o$m=='gamma')q_to_old=dmvnorm(c(o$ap,o$bp),between$MU_pG,
                                     between$SIGMA_pG,log=TRUE)+
                             dmvnorm(c(o$at,o$bt),between$MU_tG,
                                     between$SIGMA_tG,log=TRUE)
    if(o$m=='lognormal')q_to_old=etc
    if(o$m=='exponential')q_to_old=dnorm(c(o$rp),between$mu_pE,
                                         between$sigma_pE,log=TRUE)+
                                   dnorm(c(o$rt),between$mu_tE,
                                         between$sigma_tE,log=TRUE)
    if(p$m=='gamma')q_to_new=dmvnorm(c(p$ap,p$bp),between$MU_pG,
                                     between$SIGMA_pG,log=TRUE)+
                             dmvnorm(c(p$at,p$bt),between$MU_tG,
                                     between$SIGMA_tG,log=TRUE)
    if(p$m=='lognormal')q_to_new=etc
    la=p$logposterior-o$logposterior + q_to_old - q_to_new
    lu= -rexp(1)
    if(lu>la)reject=TRUE
  }
  if(reject)p=o
  p
}
```

The same code described above is then run, but with the following inserted in the loop over iterations:

```
  o=p
  p$m=sample(c('exponential','gamma','lognormal','weibull'),1)
  if(p$m=='exponential')
  {
    p$rt=rnorm(1,between$mu_tE,between$sigma_tE)
    p$rp=rnorm(1,between$mu_pE,between$sigma_pE)
  }
  if(p$m=='gamma')
  {
    NEWt=rmvnorm(1,between$MU_tG,between$SIGMA_tG)
    NEWp=rmvnorm(1,between$MU_pG,between$SIGMA_pG)
    p$at=NEWt[1];p$bt=NEWt[2]
    p$ap=NEWp[1];p$bp=NEWp[2]
  }
  ...etc...
  p=betweenmh(p,o)
```

To obtain model averages, we can use the following.

```
p=storage
times=seq(0,35,0.1)
```

```
CIt=array(0,c(3,length(times)))
for(i in 1:length(times))
{
  st=pexp(times[i],exp(p$rt),lower.tail=FALSE)*(p$mE==1)+
     pgamma(times[i],exp(p$at),exp(p$bt),lower.tail=FALSE)*(p$mG==1)+
     plnorm(times[i],p$mt,exp(p$st),lower.tail=FALSE)*(p$mL==1)+
     pweibull(times[i],exp(p$kt),exp(p$lt),lower.tail=FALSE)*(p$mW==1)
  CIt[,i]=quantile(st,c(.025,.5,0.975))
}
```

and similarly for the placebo arm.

The posterior model probabilities are obtained by taking means (for instance, `mean(p$mE)`) and are 14% (exponential), 51% (gamma), 16% (log-normal) and 19% (Weibull)—although there is a slight preference for the gamma, all are plausible models that could have generated the data (conditional on one of them having done so).

Model averaged survival curves are plotted in figure 4.17.

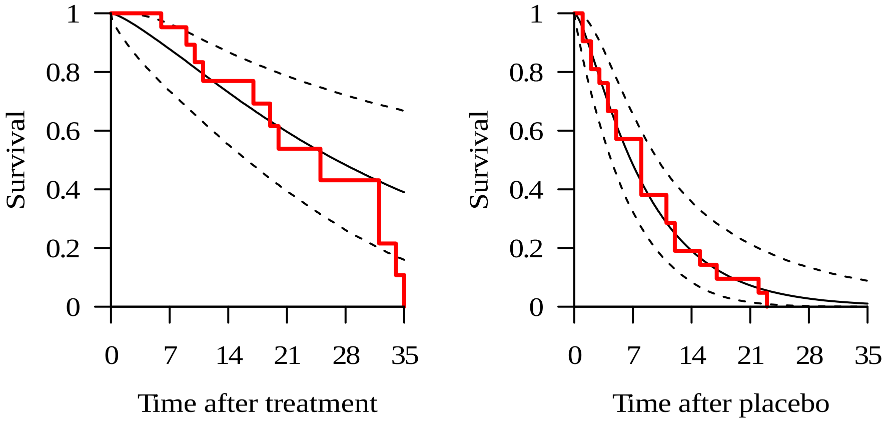

Figure 4.17: **Model averaged estimates of survival function for leukaemia example**. The distribution of survival functions (posterior median and 95% interval, black), are plotted along with the Kaplan–Meier estimate from the data (red). The treatment arm is plotted left, the plabeco, right.