Chapter -1

Second-half preliminaries

-1.1 Opening remarks

Bayesian statistical methods are not just another set of techniques for statisticians. They provide us with a different way of thinking about statistical inference and uncertainty. A posterior distribution encodes our uncertainty about quantities and events of interest. Frequentist statisticians do point estimation, create confidence intervals and hypothesis tests. Bayesians have analogues for all these things, but the interpretations are far more natural.

-1.2 What I expect you to know

-1.2.1 Integration and summation by inspection

Let $g(x)$ be a function such that $g(x) = cf(x)$ where $f(x)$ is a probability density function and c is a constant, then

$$\int_{\mathcal{X}} g(x)dx = \int_{\mathcal{X}} cf(x)dx = c\int_{\mathcal{X}} f(x)dx = c.$$

Example -1.2.1

(a) $\int_{-\infty}^{\infty} e^{-0.5(x-5)^2}dx = \sqrt{2\pi}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{-0.5(x-5)^2}dx = \sqrt{2\pi}$ N(5,1)

(b) $\int_0^{\infty} x^4 e^{-\frac{x}{2}}dx = 2^5\Gamma(5)\int_0^{\infty} \frac{x^4e^{-\frac{x}{2}}}{2^5\Gamma(5)}dx = 2^5\Gamma(5)$ Ga(5,0.5)

If $f(x)$ is a probability function and $g(x)$ is supported on the same set of discrete values $\mathcal{X}$: $g(x) = cf(x)$, then

$$\sum_{\mathcal{X}} g(x) = \sum_{\mathcal{X}} cf(x) = c\sum_{\mathcal{X}} f(x) = c.$$

Example -1.2.2

$\sum_{n=0}^{\infty} \frac{3^n}{n!} = e^3$ Po(3)

-1.2.2 The distributions in Bayesian inference

Prior distribution

We are uncertain about θ that takes some value from the set Θ. We use the prior density, $\pi(\theta)$, to encode our uncertainty about θ. If we had to guess a value for θ, we might report the mean. Before we see data, this is called our prior mean:

$$E_\theta(\theta) = \int_\Theta \theta\pi(\theta)d\theta.$$

The likelihood

The likelihood tells us about how likely a value of θ is for different data x (in relative terms). The likelihood is only specified up to a constant of proportionality:

$$l(\theta; x) \propto \pi(x \mid \theta).$$

The likelihood encodes our beliefs about the data-generating process, and it links the data to the uncertain parameter θ.

Note that I tend to use $L(\theta; x)$ for the log-likelihood.

Preposterior (or prior-predictive) distribution

Given that we are uncertain about θ and we believe x is generated from some stochastic process, we are uncertain about the value of x that we will observe. The preposterior distribution encodes this uncertainty:

$$\pi(x) = \int_\Theta \pi(x, \theta)d\theta = \int_\Theta \pi(x \mid \theta)\pi(\theta)d\theta = E_\theta[\pi(x \mid \theta)].$$

The posterior distribution

After we observe some data x^* say, we update our beliefs using

$$\pi(\theta \mid x^*) \propto \pi(x^* \mid \theta)\pi(\theta),$$

where $\pi(\theta \mid x^*)$ is our posterior density.
We find the constant of proportionality through

$$\pi(\theta \mid x^*) = c\pi(x^* \mid \theta)\pi(\theta) \implies \int_\Theta \pi(\theta \mid x^*)d\theta = 1 = c\int_\Theta \pi(x^* \mid \theta)\pi(\theta)d\theta$$

$$\implies \frac{1}{c} = \int_\Theta \pi(x^* \mid \theta)\pi(\theta)d\theta = \pi(x^*)$$

This constant $\pi(x^*)$ is called the evidence and is an instance of the preposterior distribution.

The (posterior-)predictive distribution

After observing x^*, we are still uncertain about θ, and, hence, we are still uncertain about the next data value x. This uncertainty is encoded in our predictive distribution:

$$\begin{aligned}\pi(x \mid x^*) &= \int_\Theta \pi(x, \theta \mid x^*)d\theta \\ &= \int_\Theta \pi(x \mid \theta, x^*)\pi(\theta \mid x^*)d\theta \\ &= E_{\theta|x^*}[\pi(x \mid \theta)].\end{aligned}$$

Chapter 7

Directed acyclic graphs

7.1 Historical notes

Directed acyclic graphs (DAGs) and modern Bayesian inference have an intertwined history. DAGs are used to represent probabilistic relationships between different variables. These representations are particularly useful for representing complex systems in which the variables are interdependent and for making predictions about the values of these variables based on observations.

Bayesian inference is particularly well-suited for use with DAGs, because the DAG allows the probabilistic relationships between variables to be represented in a graphical form. This makes it easier to understand and reason about the dependencies between variables and to make predictions based on observations.

The use of directed acyclic graphs (DAGs) for Bayesian inference dates back to the work of Judea Pearl, who is considered a pioneer in the field of artificial intelligence and the development of Bayesian networks. Pearl's work on DAGs and Bayesian inference began in the 1980s, and he is credited with developing the first algorithms for computing the probability of events based on the structure of a DAG.

7.2 Conditional independence

Consider three variables X, Y, Z. The issue is

$$\text{Pr}(X, Y, Z) = \text{Pr}(Z|X, Y)\text{Pr}(Y|X)\text{Pr}(X) = \text{Pr}(Y|Z, X)\text{Pr}(Z|X)\text{Pr}(X) = \dots$$

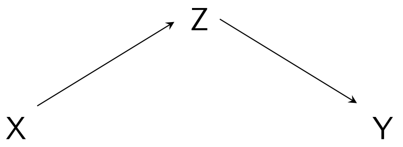
One definition of independence of X and Y (X $\perp$ Y) is, if $\text{Pr}(X, Y) = \text{Pr}(X)\text{Pr}(Y)$, then $X \perp Y$. If X is independent of Y given Z, we say that X is conditionally independent of Y and write $X \perp Y|Z$.

Example 7.2.1

X: my great grandmother's height,
Y: a measurement of my sister's height,
Z: my sister's height.
If I knew Z, Y would give me no extra information about X, as $X \perp Y|Z$.
In terms of the joint distribution, I can write

$$\begin{aligned}\text{Pr}(X, Y, Z) &= \text{Pr}(Y|X, Z)\text{Pr}(Z|X)\text{Pr}(X)\\ &= \text{Pr}(Y|Z)\text{Pr}(Z|X)\text{Pr}(X).\end{aligned}$$

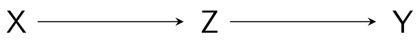
This example could be captured in a graph:



We can associate graphs with possible factorisations of joint probability distributions. In the following, conditional independence information has simplified the graphs.

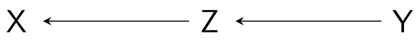
$\text{Pr}(X, Y, Z) = \text{Pr}(X)\text{Pr}(Z|X)\text{Pr}(Y|X, Z)$, and because we have $X \perp Y|Z$,

$\text{Pr}(X, Y, Z) = \text{Pr}(X)\text{Pr}(Z|X)\text{Pr}(Y|Z)$.



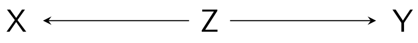
In the case of $X \perp Y|Z$, we can also write:

$\text{Pr}(X, Y, Z) = \text{Pr}(Y)\text{Pr}(Z|Y)\text{Pr}(X|Y, Z)$, $\text{Pr}(X, Y, Z) = \text{Pr}(Y)\text{Pr}(Z|Y)\text{Pr}(X|Z)$.



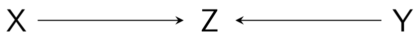
Yet another alternative is derived by considering

$\text{Pr}(X, Y, Z) = \text{Pr}(Z)\text{Pr}(Y|Z)\text{Pr}(X|Z)$ because $\text{Pr}(X|Y, Z) = \text{Pr}(X|Z)$ as $X \perp Y|Z$.



The final possible graph with two arrows does not have the same interpretation.

$\text{Pr}(X, Y, Z) = \text{Pr}(X)\text{Pr}(Y)\text{Pr}(Z|X, Y) = \text{Pr}(X, Y)\text{Pr}(Z|X, Y)$ as $X \perp Y$.



This will lead to subtle problems in representing (in)dependence via graphs and in having information flow through a graphical model.

7.3 Directed acyclic graphs

We have a collection of variables $X_1, \dots, X_n$.

We can represent the dependence structure between these variables in a graph made up of nodes, which represent the variables, and arrows, which represent direct relationships.

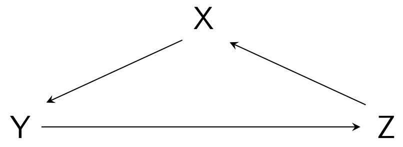
If an arrow runs from X_i to X_j, then X_i is said to be a parent of X_j (and X_j is a child of X_i). The use of arrows make this a directed graph, and we can link to probability factorisation through the following.

We write the set of all parents of X_j as $X_{\mathsf{parents}(j)}$. We can write

$$\mathsf{Pr}(X_1, \dots, X_n) = \prod_{i=1}^{n} \mathsf{Pr}(X_i | X_{\mathsf{parents}(i)}).$$

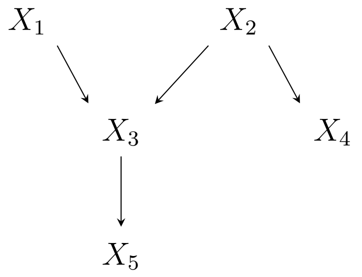
We call the variables without parents *exogenous* and variables with parents *endogenous* (*exo-* means outer, *endo-* means inner and *-genous* means originating).

Because we are representing joint probability distributions, we need to avoid cyclic behaviour. Consider three variables X, Y and Z,



$$\mathsf{Pr}(X, Y, Z) \neq \mathsf{Pr}(X|Z)\mathsf{Pr}(Y|X)\mathsf{Pr}(Z|Y)$$

Example 7.3.1

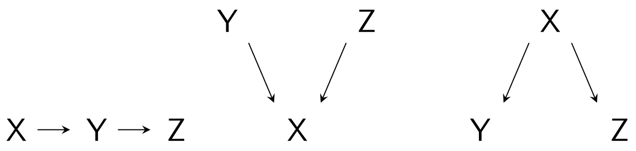


X_1 and X_2 are exogenous. We can decompose the joint density for $X_1, \ldots, X_5$ given this structure.

$$\mathsf{Pr}(X_1, \ldots, X_5) = \mathsf{Pr}(X_1)\mathsf{Pr}(X_2)\mathsf{Pr}(X_3|X_1, X_2)\mathsf{Pr}(X_4|X_2)\mathsf{Pr}(X_5|X_3)$$

So, $X_1 \perp X_2$, $X_1 \perp X_5|X_3$, $X_2 \perp X_5|X_3$ and $X_3 \perp X_4|X_2$.

The three basic building blocks for these types of graphs are chains, colliders and forks.



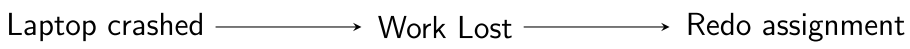
Pick any two variables, X_i and X_j, where X_j is not a parent of X_i. Consider the distribution of X_i and X_j conditional on the parents of X_i. There are two possibilities:

1. X_j is not a descendant of X_i, then conditioning on $X_{\mathsf{parents}(i)}$ has removes the influence of X_j: that is, $X_i \perp X_j|X_{\mathsf{parents}(i)}$.

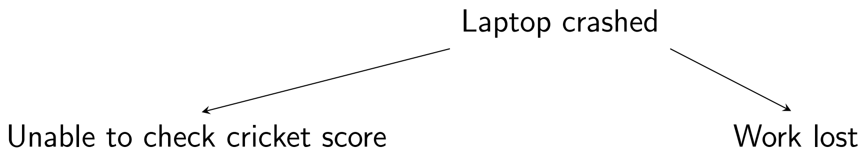
2. X_j is a descendant of X_i, then conditioning on $X_{\mathsf{parents}(i)}$ does not affect the relationship between the two variables, so $X_i \not\perp X_j|X_{\mathsf{parents}(i)}$.

Example 7.3.2

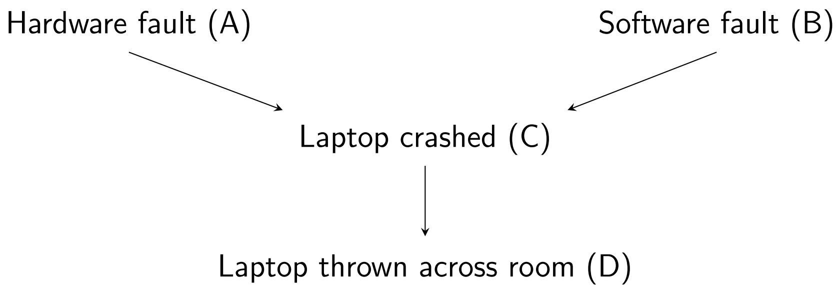
CHAIN:



FORK:



COLLIDER:



We have $A \perp B$, $A \not\perp B|C$ and $A \not\perp B|D$.

7.4 The Markov property

A sequence of random variables $X_1, X_2, X_3, \ldots$ forms a Markov process when the past is independent of the future given the present:

$$X_{t+1} \perp \{X_{t-1}, X_{t-2}, \ldots, X_1\}|X_t \Rightarrow \{X_{t+1}, X_{t+2}, \ldots\} \perp \{X_{t-1}, X_{t-2}, \ldots\}|X_t$$

This is the Markov property. DAG models have a similar property:

$$X_i \perp X_{\text{non-descendants}(i)}|X_{\text{parents}(i)}$$

This is the **directed graph Markov property**, and the Markov property above can be thought of as a special case:

$$X_1 \to X_2 \to X_3 \to \ldots$$

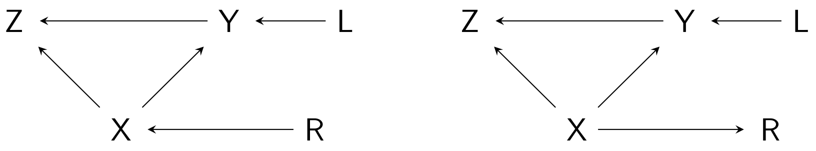
All joint distributions that conform to a given DAG have a common set of conditional independence relationships due to the Markov property.

7.5 Use in Bayesian Statistics

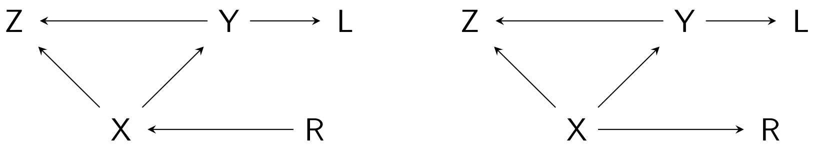
DAGs form the basis of Bayesian networks.

Example 7.5.1

A Bayesian network can be based on categorical variables.
Z = will chemical harm humans? Y = will chemical harm mice? X = chemical reactive?
L = will the experiment for mice return positive?
R = will a different experiment show reactivity?
If we have L ⊥ R, then there are two possible DAGs:



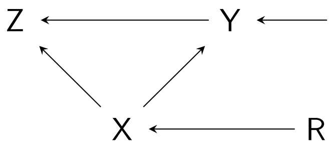
If we have L$\not\perp$ R, then there are two possible DAGs



We can also use the graphs to establish independence and conditional independence to help speed up full conditional calculations.

Example 7.5.2

Continuing the previous example. We observe $X = x$, and we wish to make inferences about the other variables.



For our sampler, we require

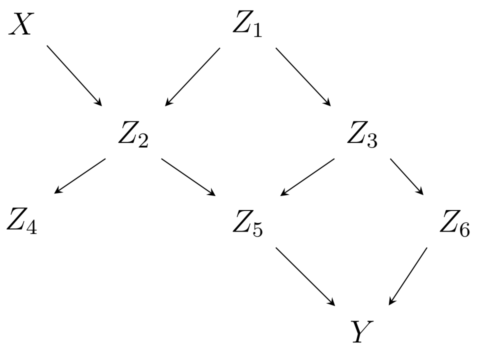
$$\begin{aligned}
\pi\left(L^{(i+1)} \middle| R^{(i)}, x, Y^{(i)}, Z^{(i)}\right) &\propto \pi\left(L^{(i+1)}\right) \pi\left(Y^{(i)} \middle| L^{(i+1)}, x\right) \\
\pi\left(R^{(i+1)} \middle| L^{(i+1)}, x, Y^{(i)}, Z^{(i)}\right) &\propto \pi\left(R^{(i+1)}\right) \pi\left(x \middle| R^{(i+1)}\right) \\
\pi\left(Y^{(i+1)} \middle| L^{(i+1)}, R^{(i+1)}, x, Z^{(i)}\right) &\propto \pi\left(Y^{(i+1)} \middle| L^{(i+1)}, x\right) \pi\left(Z^{(i)} \middle| x, Y^{(i+1)}\right) \\
\pi\left(Z^{(i+1)} \middle| L^{(i+1)}, R^{(i+1)}, x, Y^{(i+1)}\right) &\propto \pi\left(Z^{(i+1)} \middle| x, Y^{(i+1)}\right)
\end{aligned}$$

7.6 D-separation

We have three variables, X, Y and Z, and a set of variables $\mathcal{S}$ such that X,Y $\notin \mathcal{S}$.
Consider an undirected path from X to Y. The path from X to Y is "blocked" conditioning on Z $\iff$ the path contains a chain or a fork: Z is the middle node
OR
the path contains a collider like $A \to C \leftarrow B$ and $C \neq Z$, and Z is not a descendent of C.
If a path from X to Y is not blocked conditioning on Z, we say it is "active".

Note, if the path contains a collider such that $A \to Z \leftarrow B$, the path may become active conditioning on Z.
We can extend this to conditioning on $\mathcal{S}$: a path is blocked if just one element of $\mathcal{S}$ blocks a path.
When $\mathcal{S}$ blocks every path between X and Y, we say that $\mathcal{S}$ d-separates X and Y and this implies X$\perp$Y$|\mathcal{S}$.

Example 7.6.1



Consider the (undirected) paths between X and Y.
(1) $X \rightarrow Z_2 \rightarrow Z_5 \rightarrow Y$
(2) $X \rightarrow Z_2 \leftarrow Z_1 \rightarrow Z_3 \rightarrow Z_5 \rightarrow Y$
(3) $X \rightarrow Z_2 \leftarrow Z_1 \rightarrow Z_3 \rightarrow Z_6 \rightarrow Y$
(4) $X \rightarrow Z_2 \rightarrow Z_5 \leftarrow Z_3 \rightarrow Z_6 \rightarrow Y$
Consider the set $\{Z_3, Z_5\}$; are X and Y d-separated by this set?
In path (1), we have $Z_2 \rightarrow Z_5 \rightarrow Y$ (a chain) with Z_5 in our conditioning set $\Rightarrow$ path (1) blocked.

In path (2), we have $Z_1 \rightarrow Z_3 \rightarrow Z_5$ (a chain) with Z_3 in our conditioning set $\Rightarrow$ path (2) blocked.

In path (3), we have $Z_1 \rightarrow Z_3 \rightarrow Z_6 \Rightarrow$ path (3) blocked.

In path (4), we have $Z_5 \leftarrow Z_3 \rightarrow Z_6$ (a fork) with Z_3 in our conditioning set $\Rightarrow$ path (4) blocked.

All paths are blocked by $\{Z_3, Z_5\} \iff$ X and Y are d-separated by $\{Z_3, Z_5\} \iff$ X $\perp$ Y $|\{Z_3, Z_5\}$.

Consider the set $\{Z_2, Z_6\}$; are X and Y d-separated by this set?
In path (1), $X \rightarrow Z_2 \rightarrow Z_5 \Rightarrow$ path (1) is blocked. In path (2), our conditioning set only occurs as part of $X \rightarrow Z_2 \leftarrow Z_1$, so the path is active and X and Y are not d-separated by $\{Z_2, Z_6\}$ and $X \not\perp Y|\{Z_2, Z_6\}$.

Chapter 8

Hierarchical models

8.1 Introduction

Hierarchical models underpin a lot of real-world modelling. Sometimes we will have few or no data available that directly inform us about the things that we are interested in so borrow from available data that tell us about related things. This relates to the ideas in the previous chapter on DAGs because we will be interested in the flow of information from one set of observations to another.

A key text here is Bayesian Data Analysis by Gelman *et al.* (2013), and there are some useful observations in Kendall's Advanced Theory of Statistics Volume 2B: Bayesian Inference by O'Hagan and Forster (2004) (Chapters 6 and 7 in particular).

8.2 Exchangeability

Bruno De Finetti (1906–1985) was an early modern-statistician who wrote a two volume book that was grandly named "*Theory of Probability*". In that book, he laid out an axiomatic approach to probability that underpins modern day Bayesian statistics. A key idea formalised in his work was exchangeability.

A sequence of random variables is said to be *exchangeable* if we can reorder the sequence without changing the joint distribution of those random variables. Mathematically, we have a sequence $X_1, X_2, X_3, \ldots$ and a bijective permutation operator $\sigma : \mathbb{N} \to \mathbb{N}$. If these random variables are exchangable, then

$$F_{X_1,X_2,X_3,\ldots}(X_1, X_2, X_3, \ldots) = F_{X_1,X_2,X_3,\ldots}\left[X_{\sigma(1)}, X_{\sigma(2)}, X_{\sigma(3)}, \ldots\right],$$

and *vice versa*. This may remind you of the modelling assumption that random variables are independent and identically distributed (i.i.d), and there is a strong relationship. In fact, independence implies exchangeability, the converse is not necessarily true.

Example 8.2.1

Imagine that I have selected six local councils and recorded each council's overspend in £m for the last financial year. Let us label these six unknowns as $x_1, \ldots, x_6$. What can you say about x_6, the overspend for the sixth council?

I have not given you any additional information so you should model them exchangeably, and each unknown value would be assigned an appropriate distribution over the real-line.

We now get the values for five of the six councils: 18, 5, 2, 7 and 8. A reasonable posterior predictive value for x_6 would be centred around 8 with a fair bit of spread, 0 to 25 perhaps.

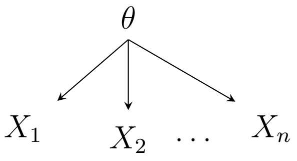
Changing the indices will not change this posterior estimate. Therefore, the x_i are exchangeable.

However, the x_i are certainly not independent: we would rightly assume that the overspend for the sixth council is similar to the observed spends.

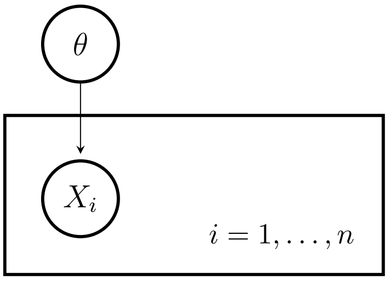
Example 8.2.2

Pólya Urn

Typically, we will talk about exchangeability of unobserved data variables conditional on known parameter values (which will lead to the assumption of conditional independence).



An alternative representation that will be more useful later is:



Here, we have

$$X_i \perp X_j | \theta \;\; \forall \; i \neq j.$$

8.3 Hierarchical models

Formally, a hierarchical model is a statistical model with multiple levels with each level representing a broader grouping of individual experimental or observed units. We will be using notation of the following form to capture the multiple levels for the units:

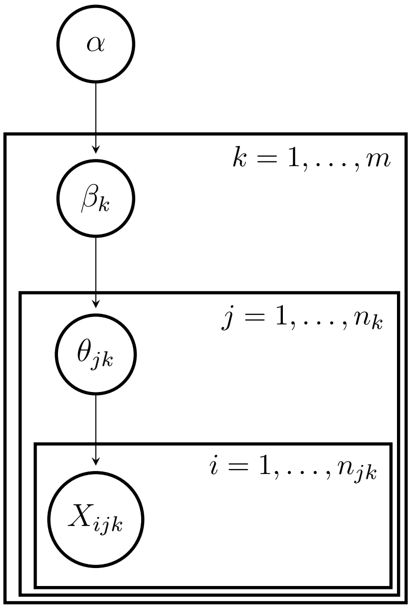
$$x_{ijk}$$

Here, the k indexes the overall group the unit belongs to, j indexes the subgroup and i indexes the observation within the subgroup.

These groups do not need to be balanced in terms of the number of groups or units with those groups. This will lead us to having

$$n_{jk}$$

denoting sample size for the jth group within the kth overall group for a three-level hierarchy.

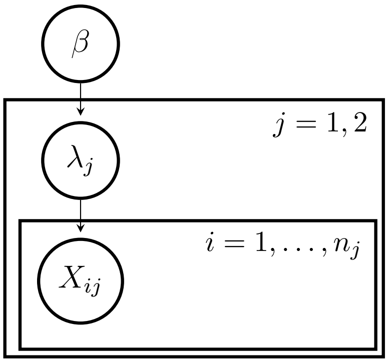


Note that we may model exchangeably at any of these level, and we could be modelling exchangeably at the highest level, but have strong dependencies at lower levels.

Example 8.3.1

Consider the following model:

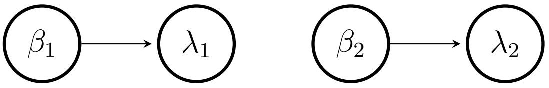
$$\begin{aligned} X_{ij}|\lambda_j &\sim \textsf{Poisson}(\lambda_j), \quad i = 1, \ldots, n_j, \\ \lambda_j|\beta &\sim \textsf{Exp}(\beta), \qquad j = 1, 2, \\ \beta &\sim \textsf{Exp}(1). \end{aligned}$$



We observe the following data:

$$\begin{aligned} n_1 = 3, &\quad \sum_{i=1}^{3} x_{i1} = 1, \\ n_2 = 10, &\quad \sum_{i=1}^{10} x_{i2} = 7. \end{aligned}$$

Now, suppose that we ignore the common β:



$$\begin{aligned} \lambda_1|x_{\bullet 1}, \beta_1 = 1 \sim \textsf{Gamma}(2, 4), &\quad \Longrightarrow \quad \textsf{Var}(\lambda_1|x_{\bullet 1}) \approx 0.127, \\ \lambda_2|x_{\bullet 2}, \beta_2 = 1 \sim \textsf{Gamma}(8, 11), &\quad \Longrightarrow \quad \textsf{Var}(\lambda_2|x_{\bullet 2}) \approx 0.067. \end{aligned}$$

Back to the full model, we have

$$\begin{aligned} \pi\left(\beta, \lambda_1, \lambda_2|x_{\bullet\bullet}\right) &\propto \pi(\beta)\pi(\lambda_1|\beta)\pi(\lambda_2|\beta)\pi(x_{\bullet 1}|\lambda_1)\pi(x_{\bullet 2}|\lambda_2) \\ &\propto \beta^2 \exp\left(-(\lambda_1 + \lambda_2 + 1)\beta\right) \lambda_1 \lambda_2^7 \exp\left(-3\lambda_1 - 10\lambda_2\right). \end{aligned}$$

Note that

$$\int_0^\infty \beta^2 \exp\left(-(\lambda_1 + \lambda_2 + 1)\beta\right) d\beta = \frac{\Gamma(3)}{(\lambda_1 + \lambda_2 + 1)^3}.$$

Therefore,

$$\pi\left(\lambda_1, \lambda_2|x_{\bullet\bullet}\right) \propto \frac{\lambda_1 \lambda_2^7 \exp\left(-3\lambda_1 - 10\lambda_2\right)}{(\lambda_1 + \lambda_2 + 1)^3}.$$

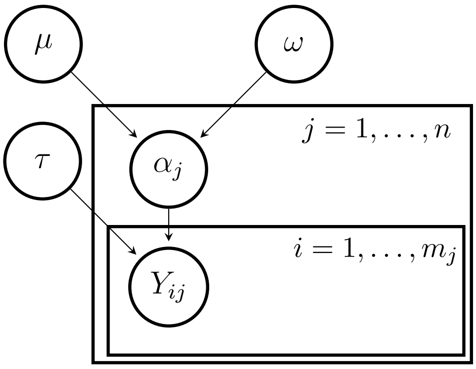
With the help of some numerical integration, we find that

$$\textsf{Var}(\lambda_1|x_{\bullet\bullet}) \approx 0.117 \text{ and } \textsf{Var}(\lambda_2|x_{\bullet\bullet}) \approx 0.065.$$

Example 8.3.2

Random-effects models are the archetypal hierarchical model:

$$\begin{aligned}
Y_{ij}|\alpha_j,\tau &\sim \mathsf{N}(\alpha_j,\tau^{-1}), \quad i=1,\ldots,m_j, \quad j=1,..,n,\\
\alpha_j|\mu,\omega &\sim \mathsf{N}(\mu,\omega^{-1}), \quad j=1,..,n,\\
\mu &\sim \mathsf{N}(0,1),\\
\tau &\sim \mathsf{Gamma}(2,1),\\
\omega &\sim \mathsf{Gamma}(1,1).
\end{aligned}$$



The influence of the prior distribution in general is to pull the likelihood towards the prior. When several parameters have a common prior mean, the posterior estimates will all be pulled towards the common mean. This means that Bayesian estimates of this nature will be less spread out than non-Bayesian estimates. This is known as *shrinkage*.

Example 8.3.3

For the random-effects model of **Example 8.3.2**, we can derive the following full conditional distribution for each α_j:

$$\alpha_j|y_{\bullet\bullet},\alpha_{-j},\mu,\tau,\omega \sim \mathsf{N}\left[\frac{\omega\mu+\tau\sum y_{ij}}{\omega+m_j\tau},(\omega+m_j\tau)^{-1}\right].$$

A full conditional is not the same as the posterior, but it is a slice through it and we can see shrinkage in action for the group means α_j with the overall mean μ being used in an expression for the mean alongside the data-based estimate.

When tracking the influence of data on our posterior beliefs, it can be useful to quantify the relative reduction in variance. Define *resolution* to be:

$$R_x(\theta)=1-\frac{\mathsf{Var}(\theta|x)}{\mathsf{Var}(\theta)}.$$

If the resolution is 1, we have nothing left to learn about θ.

8.4 Probabilistic programming

Probabilistic programming is a form of programming that enables the (semi-)automation of Bayesian inference. It is different to normal programming in that it allows for the incorporation of uncertainty into the programming process. By using probabilistic programming, we can develop models that take into account the uncertainty of the data and make more informed decisions whilst taking advantage of the conditional independence structure in our models.

There are quite a few well established options that we could utilise:

- **BUGS**: Bayesian inference Using Gibbs Sampling — sampling from full conditionals,
- **JAGS**: Just Another Gibbs Sampler — cross-platform version of BUGS,
- **INLA**: Integrated Nested Laplace Approximation — approximate Bayesian inference for Markov random fields,
- **Stan**: named in honour of Stanislaw Ulam (1909-1984) — has several inbuilt algorithms for automated posterior sampling (plus tools for model selection).

The main algorithm that we will utilise in Stan is the no U-turns sampler (NUTS), which is a variant of Hamiltonian Monte Carlo (HMC). The HMC algorithm is an efficient MCMC algorithm that uses Hamiltonian dynamics to sample from a probability distribution. The simulation then follows the trajectories of the Hamiltonian dynamics allowing it to explore a large number of parameter values in a relatively short amount of time.

1. Initialise parameters
2. For each iteration,
   a) Choose a random momentum vector,
   b) Compute the Hamiltonian,
   c) Evolve the system using Hamiltonian dynamics,
   d) Compute the acceptance probability,
   e) Accept or reject.

HMC is particularly useful for exploring multi-modal distributions, which traditional MCMC methods struggle with. A really good explanation of the concepts involved in HMC and NUTS can be found here:

https://elevanth.org/blog/2017/11/28/build-a-better-markov-chain/

Stan model code is made up of three core components (this code is from **Example 8.3.1**):

```
data {
  int<lower=0> N;                 // num individuals
  int<lower=1> J;                 // num groups
  int<lower=1,upper=J> group[N];  // group for individual
  int<lower=0> X[N];              // observed random variables
}
parameters {
  real<lower=0> beta;         // hyperparameter
  real<lower=0> lambda[J];    // rate by group
}
model {
  // Prior
  beta ~ gamma(1, 1);
  for (j in 1:J)
      lambda[j] ~ gamma(1, beta);

  // Likelihood
  for (n in 1:N)
      X[n] ~ poisson(lambda[group[n]]);
}
```

Note that, in R, the model gets compiled into C++ so every time the model is changed R needs to compile again (which can take some time). This can be problematic if we are investigating changes to the prior parameters. A useful workaround is to include the prior parameter values in the data specification:

```
data {
  int<lower=0> N;                 // num individuals
  int<lower=1> J;                 // num groups
  int<lower=1,upper=J> group[N];  // group for individual
  int<lower=0> X[N];              // observed random variables
  real<lower=0> beta_shape;       // shape parameter for beta ~ Gamma(.,.)
  real<lower=0> beta_rate;        // rate parameter for beta ~ Gamma(.,.)
}
parameters {
  real<lower=0> beta;         // hyperparameter
  real<lower=0> lambda[J];    // rate by group
}
model {
  // Prior
  beta ~ gamma(beta_shape, beta_rate);
  for (j in 1:J)
```

```
      lambda[j] ~ gamma(1, beta);

  // Likelihood
  for (n in 1:N)
      X[n] ~ poisson(lambda[group[n]]);
}
```

There are four other useful elements that we can add to a Stan model to extend its utility:

- `transformed data`,
- `transformed parameters`,
- `generated quantities`,
- `functions`.

The R code for running a Stan model and producing posterior samples is relatively easy:

```
# load the library (note that you need Rtools installed)
library(rstan)

# compile the model
model <- stan_model(file = 'Hierarchical models/Poisson model.stan')

# generate a posterior distribution
fit <- sampling(model,
                chains = 4,
                iter = 100000,
                data = list(N = 3 + 10,
                            J = 2,
                            group = c(rep(1,3),
                                      rep(2,10)),
                            X = c(0,1,0,
                                  2,0,0,1,2,0,0,1,1,0)))
```

Then you can summarise the fits and extract the samples from the `stan fit` object:

```
# summary stats for all model parameters
summary(fit)

# extract the posterior samples
l1_samples <- extract(fit, pars = 'lambda[1]')$'lambda[1]'
```

```
l2_samples <- extract(fit, pars = 'lambda[2]')$'lambda[2]'

# plot histograms
hist(l1_samples)
hist(l2_samples)

# calculate variances
var(l1_samples)
var(l2_samples)
```

8.5 Priors

If a prior is not specified in Stan for a model parameter, θ say, then it will default to the following

$$\pi(\theta) \propto 1 \quad \forall\ \theta.$$

Example 8.5.1

We have

$$X|\mu \sim \text{Pareto}(\mu, 1),$$
$$\pi(\mu) \propto 1, \quad \mu > 0.$$

The posterior is then

$$\pi(\mu|X = x) \propto \frac{\mu}{x^2}.$$

We clearly have a problem because

$$\begin{aligned}\int_0^\infty \frac{\mu}{x^2} d\mu &= \lim_{t\to\infty} \int_0^t \frac{\mu}{x^2} d\mu \\ &= \lim_{t\to\infty} \frac{t^2}{2x^2},\end{aligned}$$

and the integral is undefined.

Apart from the challenge of potentially improper posterior distributions, we also have the issue of change of variables.

Example 8.5.2

Let's pretend that we know nothing about θ apart from it being positive:

$$\pi_\theta(\theta) \propto 1, \quad \theta > 0.$$

What do we know about $\phi = 1/\theta$?

$$\begin{aligned}\pi_\phi(\phi) &= F'_\theta(1/\phi) \\ &= \pi_\theta(1/\phi)\left|\frac{\partial(1/\phi)}{\partial\phi}\right| \\ &= \frac{1}{\phi^2},\end{aligned}$$

which does not seem so flat.

8.6 Expert elicitation

We should incorporate existing knowledge into a Bayesian analysis through careful selection of a prior distribution.

8.6.1 Cromwell's rule

In a letter to the Church of Scotland, Cromwell stated "I beseech you, in the bowels of Christ, think it possible you may be mistaken". For us, Cromwell's rule amounts to never assigning probabilities of 0 or 1, which both signify certainty. If we are certain *a priori*, then no amount of evidence may change our minds.

Example 8.6.1

Consider a coin that is thought to be fair. One person says that they are sure the proportion of heads, θ, is 0.5 to 1 d.p ($\theta \sim$ Uni$(0.45, 0.55)$). Another person is even more certain, and they assign $\theta \sim Be(1000, 1000)$ ($\Pr(\theta > 0.55) = 4 \times 10^{-6}$).
Graph sketched in lecture.
The coin is tossed 500 times and 499 times we have heads.
1^{st} person: posterior $\Pr(\theta > 0.55|$data$) = 0$,
2^{nd} person: posterior $\Pr(\theta > 0.55|$data$) = 0.9999997$.

8.6.2 Elicitation methods

Throughout, we will assume we have some continuous, univariate parameter θ that we want to specify a prior distribution for.

People struggle to give judgements on statistical constructs like mean and variance. It is much more reliable to focus on:

- Mode (most likely value),
- Median (equal chance of being above or below),
- Percentiles (what value of θ is there 10% chance of being above?),
- Probabilities (what's the probability that $\theta > 0$?),
- Plausible ranges (give a range of plausible values, e.g. 6σ-rule).

8.6.3 The bisection method

Q1 Specify a value of θ such that there is an equal chance of the true value being above and below (θ_m).

Q2 Imagine this true value θ is definitely below θ_m. Specify a value of θ in range $(-\infty, \theta_m)$ such that there is an equal chance of the true value being above or below (θ_l).

Q3 Imagine this true value θ is definitely above θ_m. Specify a value of θ in range (θ_m, ∞) such that there is an equal chance of the true value being above or below (θ_u).

This method is all about judging equal chances, which is easier than specifying probabilities or quartiles directly.

Example 8.6.2

What is my sister's height?

Having the three quartiles does not specify a distribution uniquely, and we may need to make a judgement about a suitable probability distribution.

If we are fitting a normal distribution to judgements (θ_l, θ_m and θ_u), we might use:

- $\theta_m = \mu$,

- $\theta_l = \mu - 0.6745\sigma$,
- $\theta_u = \mu + 0.6745\sigma$.

If we cannot determine μ and σ^2 exactly, we can choose μ and σ^2 that minimise:

$$(\theta_l^* - \theta_l)^2 + (\theta_m^* - \theta_m)^2 + (\theta_u^* - \theta_u)^2.$$

Here θ_l^* is the true lower quartile from $N(\mu, \sigma^2)$.

Example 8.6.3

My sister's height continued.

- $\theta_m^* = 5'9" = 175$cm,
- $\theta_l^* = 5'7" = 170$cm,
- $\theta_u^* = 5'11" = 180$cm.

Use website:
https://jeremy-oakley.shinyapps.io/SHELF-single/

Here are the key principles of elicitation:

I Transparency.

II Ask questions that the expert can answer - training, flexibility.

III Question the expert → Fit distribution → Feedback distribution → are revisions needed? (if yes: back to start)(if no: use this in analysis).

IV Whatever prior is chosen, we should consider sensitivity of our analysis to that choice.

When eliciting opinions from someone, it is important to consider the heuristics and biases that come into the process. For instance,

- Overconfidence,
- Availability,
- Anchoring.

Eliciting information from a group introduces another level of issues stemming from group dynamics. A gentle introduction to all of these concepts can be found in Daniel Kahnemann's book: Thinking Fast and Slow.

8.6.4 What is my prior worth?

To consider the influence of the prior we could ask the following: what number of observations is our prior worth?

Example 8.6.4

$x|\theta \sim \text{Bin}(n, \theta)$ and $\theta \sim \text{Be}(\alpha, \beta)$: $\theta|X = x \sim \text{Be}(\alpha + x, \beta + n - x)$. By properties of the beta distribution, $\text{E}(\theta) = \frac{\alpha}{\alpha+\beta}$ and $\text{E}(\theta|x) = \frac{\alpha+x}{\alpha+\beta+n}$.
We can rewrite:

$$\text{E}(\theta|x) = \frac{(\alpha+\beta)\text{E}(\theta)}{\alpha+\beta+n} + \frac{n\hat{\theta}}{\alpha+\beta+n}$$

where $\hat{\theta} = \frac{x}{n}$.
This is a weighted average of $\text{E}(\theta)$ and the data estimate, $\hat{\theta}$. In this average, the prior is worth $(\alpha + \beta)$ and the data estimate is worth n.

Chapter 9

More models

The more tools you have available, the better statistician you will be. In this chapter, we will consider some model building blocks (compounding and mixtures), issues with fitting models (identifiability specifically) and some examples of Bayesian models in fairly common contexts.

9.1 Constructing other distributions

We should not feel bound by the standard set of distributions for our modelling. We can make any density we wish. If we have some function $f(x) : \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$, all we need to be able turn it into a density is

$$0 < \int_{-\infty}^{\infty} f(x)\ dx < \infty.$$

That said, there are natural ways to create distributions that have features like extra spread or multiple modes using compounding and mixtures.

A *compound distribution* is formed by having a distribution assigned to parameters in another distribution and integrating out the uncertain parameters (this is similar to some of the hierarchical constructions that we have seen in the previous chapter).

Example 9.1.1

A t-distribution can be thought of as a compound distribution.
Consider

$$\begin{aligned} X|\tau &\sim \mathsf{N}(0, \tau^{-1}), \\ \tau &\sim \mathsf{Gamma}(\alpha, \beta). \end{aligned}$$

We want

$$\pi(x) = \int_0^\infty \pi(x|\tau)\pi(\tau)\ d\tau.$$

We know that

$$\pi(x|\tau)\pi(\tau) = \frac{\beta^\alpha}{\sqrt{2\pi}\Gamma(\alpha)}\tau^{\alpha-1/2}\exp\left[-\left(\beta + x^2/2\right)\tau\right].$$

Spotting a Gamma density, we have that

$$\pi(x) = \frac{\beta^\alpha}{\sqrt{2\pi}\Gamma(\alpha)}\frac{\Gamma(\alpha+1/2)}{(\beta+x^2/2)^{\alpha+1/2}}.$$

Ignoring constants, we see that

$$\begin{aligned}\pi(x) &\propto \left(\beta + \frac{x^2}{2}\right)^{-\alpha-1/2} \\ &\propto \left(1 + \frac{x^2}{2\beta}\right)^{-\frac{2\alpha+1}{2}},\end{aligned}$$

which is a t-distribution with 2α degrees of freedom, location parameter 0 and scale parameter $\sqrt{\beta/\alpha}$.

Example 9.1.2

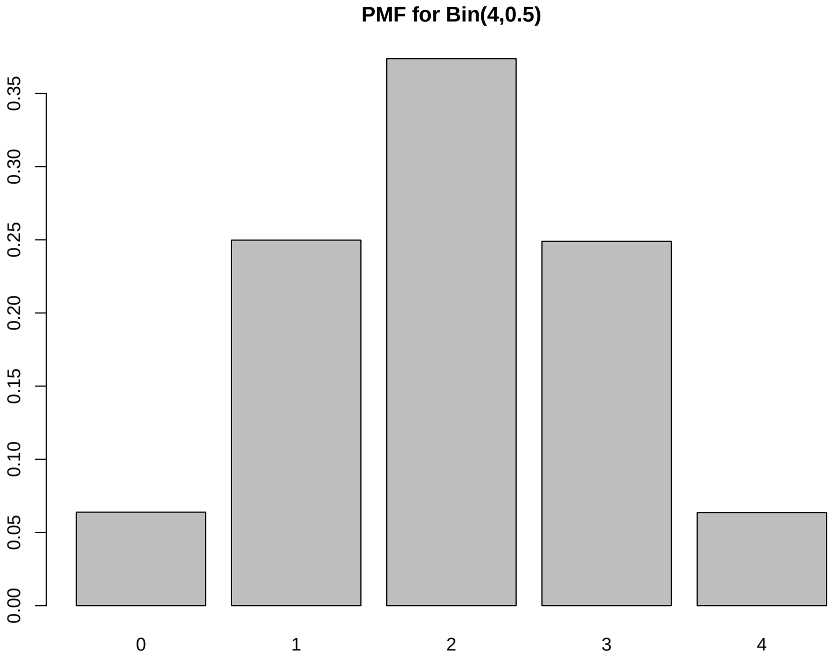
A discrete example:

$$\begin{aligned}Y|n &\sim \text{Bin}(n, 0.5), \\ n-1 &\sim \text{Poisson}(\lambda).\end{aligned}$$

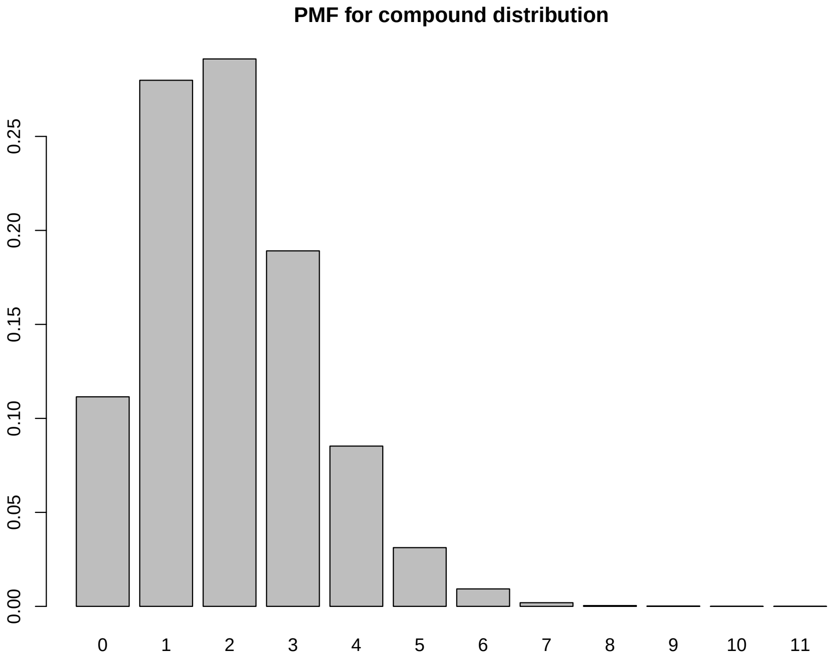
We have that

$$\pi(y) = \frac{e^{-\lambda}}{2y!}\left[\sum_{n=1}^\infty \frac{n}{(n-y)!}\left(\frac{\lambda}{2}\right)^{n-1}\right].$$

Compare the probability mass function for $Y \sim \text{Bin}(4, 0.5)$, which has variance 1,



with the probability mass function for our compound distribution with $\lambda = 3$, which has variance of approximately 1.75.



Another neat way of creating useful distributions is through mixtures. There are two primary types of mixture: sums and products. Let $\pi_1(\theta)$ and $\pi_2(\theta)$ be pdfs for θ, then we have the following valid densities

$$\pi(\theta) \propto w_1\pi_1(\theta) + w_2\pi_2(\theta)$$

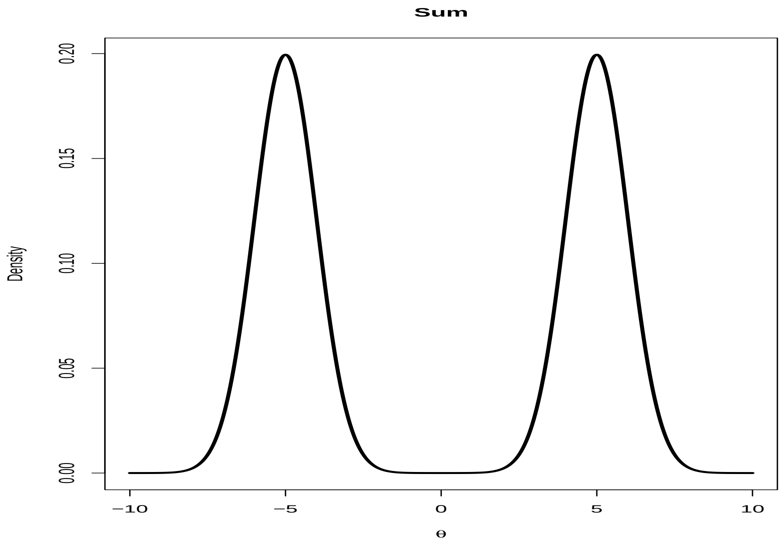
and

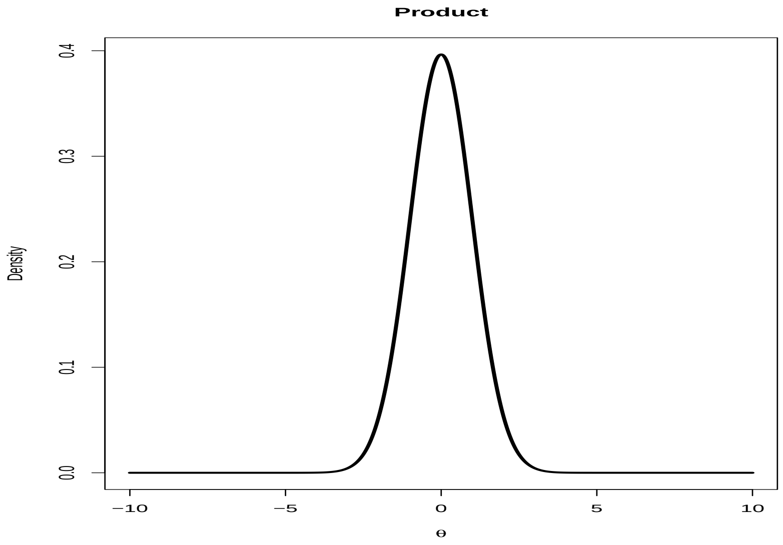
$$\pi(\theta) \propto \pi_1(\theta)^{w_1}\pi_2(\theta)^{w_2}$$

This extends to any number of mixture components.

Example 9.1.3

Let's have two densities, $\pi_1(\theta)$ and $\pi_2(\theta)$, based on N(-5,1) and N(5,1) respectively, and $w_1 = w_2 = 0.5$.





There are three other types of distribution manipulation that are commonly used to create custom distributions: *mirroring*, *folding* and *truncating*.

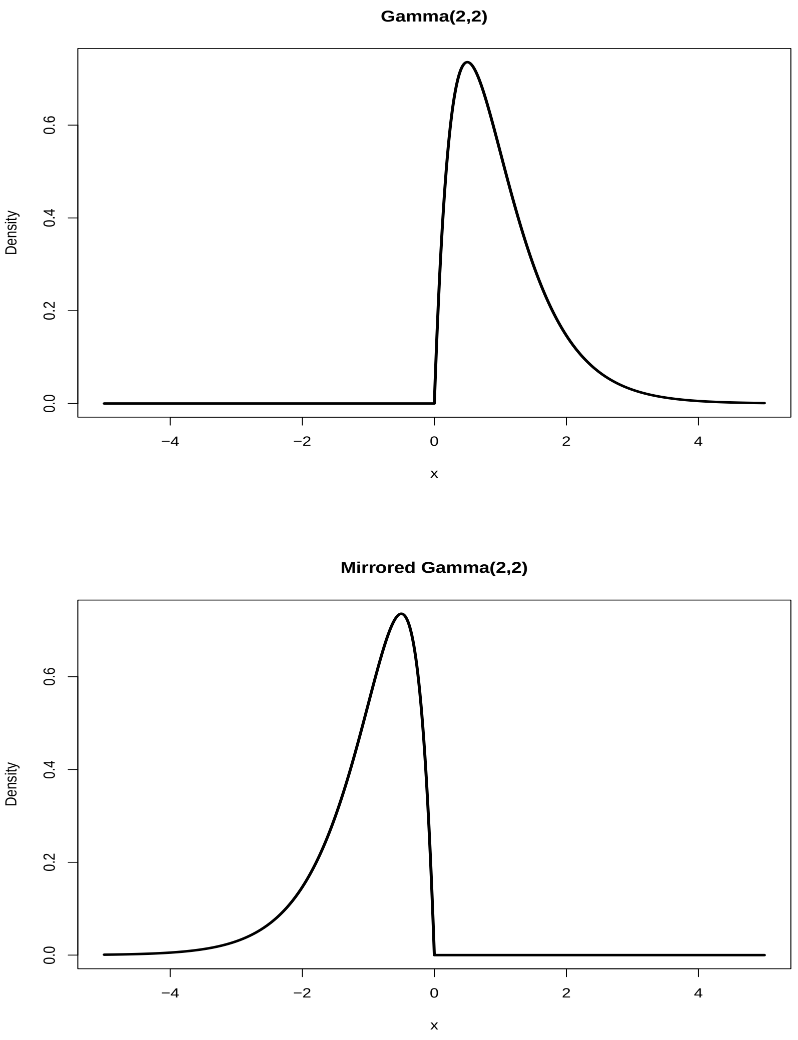
Mirroring a distribution essentially is reflecting the density about a lower or upper bound. This is usually accompanied by a shift in the distribution. Let $f(x)$ be a density for x that has a lower bound at a. A mirrored version of $f(x)$, $g(x)$ say, would be

$$g(x) = \begin{cases} f(2a-x) & x < a, \\ 0 & \text{otherwise.} \end{cases}$$

This is clearly still a valid density:

$$\int_{-\infty}^{\infty} g(x)dx \ = \ \int_{-\infty}^{a} f(2a-x)dx \ = \ -\int_{\infty}^{a} f(u)du \ = \ 1.$$

Example 9.1.4



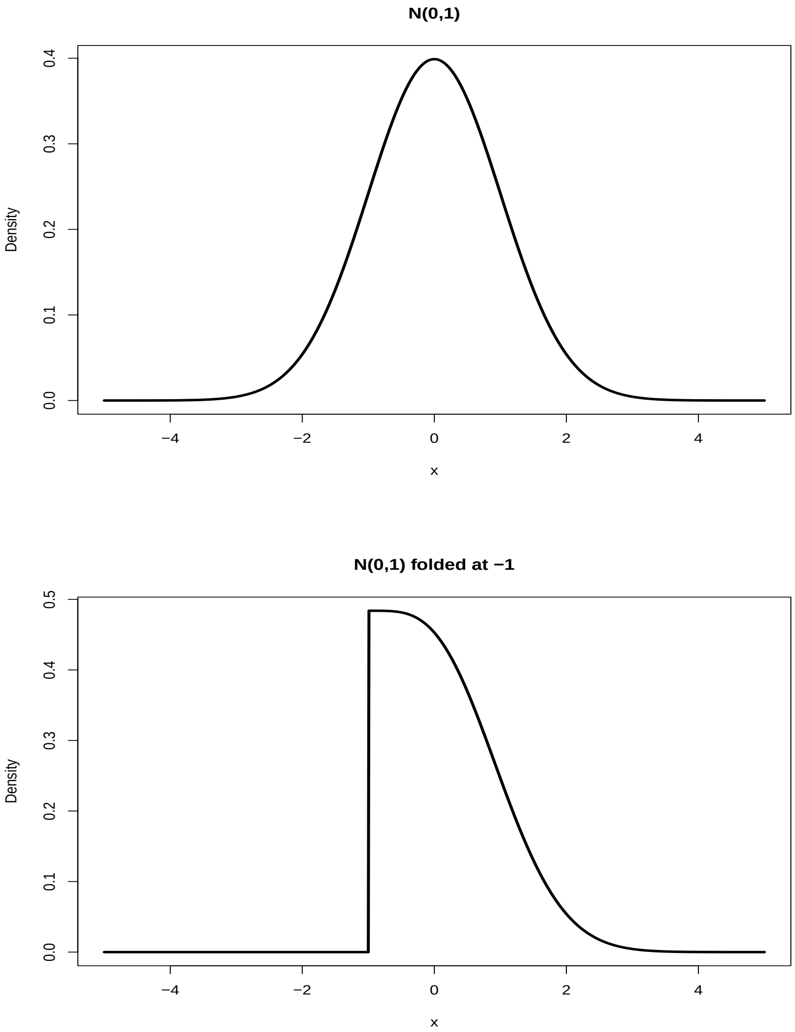
Folding a distribution means reflecting part of density in a boundary that you have imposed. It is useful for turning distributions defined on the real line into bounded distributions. Let $f(x)$ be a density that is positive for all $x \in \mathbb{R}$. A folded version of $f(x)$ at boundary a, $g(x)$ say, could be

$$g(x) = \begin{cases} f(x) + f(2a - x) & x > a, \\ 0 & \text{otherwise.} \end{cases}$$

This is a valid density:

$$\begin{aligned} \int_{-\infty}^{\infty} g(x)dx &= \int_a^{\infty} f(x) + f(2a - x)dx = \int_a^{\infty} f(x)dx + \int_a^{\infty} f(2a - x)dx \\ &= \int_a^{\infty} f(x)dx + \int_{-\infty}^{a} f(u)du = 1. \end{aligned}$$

Example 9.1.5



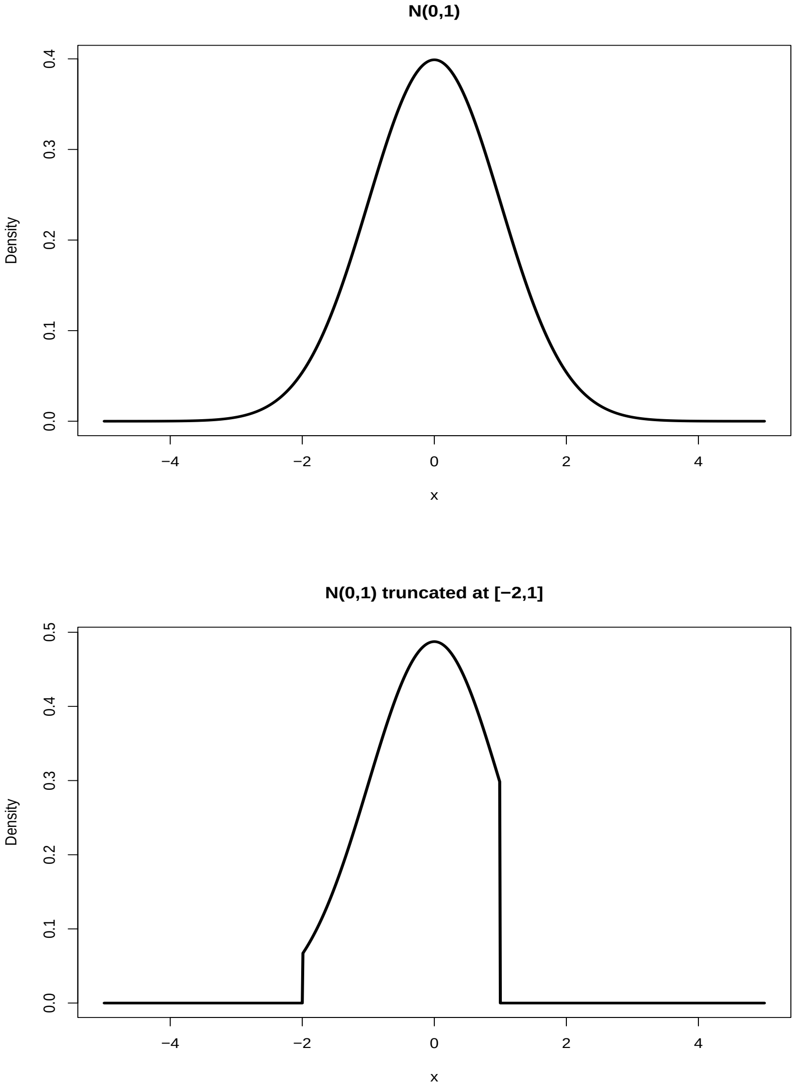
Truncation is much less subtle. We impose a bound or bounds on a distribution. Let $f(x)$ be a density that is positive for all $x \in \mathbb{R}$. A truncated version of $f(x)$ over $[a, b]$, $g(x)$ say, would be

$$g(x) = \begin{cases} \frac{f(x)}{\int_a^b f(y)dy} & x \in [a, b], \\ 0 & \text{otherwise.} \end{cases}$$

This too is a valid density:

$$\int_{-\infty}^{\infty} g(x)dx \;=\; \frac{\int_a^b f(x)dx}{\int_a^b f(y)dy} \;=\; 1.$$

Example 9.1.6



Transformations should also not be forgotten. Transformations are used throughout statistics to move data onto scales that are more amenable to modelling. In the following table, we have some unknown θ that we are transforming into ϕ.

Name	Function	Inverse	Domain	Range
Standardise	$\frac{\theta-\mu_\theta}{\sigma_\theta}$	$\phi\sigma_\theta + \mu_\theta$	$\mathbb{R}$	$\mathbb{R}$
Normalise	$\frac{\theta-\min(\theta)}{\max(\theta)-\min(\theta)}$	$[\max(\theta) - \min(\theta)]\,\phi + \min(\theta)$	$\mathbb{R}$	[0,1]
Power	θ^α	$\phi^{1/\alpha}$	$\mathbb{R}$ or $\mathbb{R}^+$	$\mathbb{R}$ or $\mathbb{R}^+$
Logarithm	$\log(\theta)$	$\exp(\theta)$	$\mathbb{R}^+$	$\mathbb{R}$
Logit	$\log\left(\frac{\theta}{1-\theta}\right)$	$\frac{1}{1+\exp(-\phi)}$	(0,1)	$\mathbb{R}$
Probit	$\Phi(\theta)$	$\Phi^{-1}(\phi)$	(0,1)	$\mathbb{R}$

9.2 Identifiability

An aspect of a statistical model is *identifiable* when it cannot be changed without there being a change in the distribution of the observed variables. More precisely, a statistical model is identifiable if it is possible to find the single true values of this model's parameters after observing an infinite number of observations.

If we can alter part of a model with no consequences for the data distributions, then that part of the model is *unidentifiable*. In this case, two or more sets of parameters give rise to equivalent data distributions.

Example 9.2.1

$X_i \sim N(\mu_1 + \mu_2, \sigma^2)$, where $i = 1, \ldots, n$.
σ^2 is identifiable.
μ_1 and μ_2 are unidentifiable.
This is clear from the log-likelihood

$$L(\mu_1, \mu_2, \sigma^2 | x_\bullet) = -\frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu_1 - \mu_2)^2$$

where we can always change μ_1 to compensate for any change to μ_2, but we cannot counteract changes to σ^2 so easily. However, $\mu = \mu_1 + \mu_2$ is identifiable.

In a Bayesian context, suppose that $\pi(x|\boldsymbol{\theta})$ depends on some function of $\boldsymbol{\theta}$, $\boldsymbol{g}(\boldsymbol{\theta})$ say, but not on the rest, $\boldsymbol{h}(\boldsymbol{\theta})$ say. When we derive the posterior, we get

$$\begin{aligned}\pi(\boldsymbol{\theta}|x) &\propto \pi(x|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\ &= \pi(x|\boldsymbol{g}(\boldsymbol{\theta}))\pi(\boldsymbol{\theta}) \\ &= \pi(x|\boldsymbol{g}(\boldsymbol{\theta}))\pi(\boldsymbol{h}(\boldsymbol{\theta})|\boldsymbol{g}(\boldsymbol{\theta}))\pi(\boldsymbol{g}(\boldsymbol{\theta})) \\ &\propto \pi(\boldsymbol{g}(\boldsymbol{\theta})|x)\pi(\boldsymbol{h}(\boldsymbol{\theta})|\boldsymbol{g}(\boldsymbol{\theta})).\end{aligned}$$

Therefore, the conditional posterior distribution of $\boldsymbol{h}(\boldsymbol{\theta})$ given $\boldsymbol{g}(\boldsymbol{\theta})$ is the same as the conditional prior distribution of $\boldsymbol{h}(\boldsymbol{\theta})$ given $\boldsymbol{g}(\boldsymbol{\theta})$, and, no matter how many we observe, we can never learn $\boldsymbol{h}(\boldsymbol{\theta})$ precisely.

Example 9.2.1 continued.
Here, we have $\boldsymbol{\theta} = (\mu_1, \mu_2, \sigma^2)^T$, $\boldsymbol{g}(\boldsymbol{\theta}) = (\mu_1 + \mu_2, \sigma^2)^T$ and $\boldsymbol{h}(\boldsymbol{\theta}) = \mu_1 - \mu_2$.
From the likelihood, we know that $\pi(\underline{x}|\boldsymbol{\theta}) = \pi(\underline{x}|\boldsymbol{g}(\boldsymbol{\theta}))$,
and we will find that

$$\pi(\boldsymbol{\theta}|\underline{x}) \propto \pi(\underline{x}|\boldsymbol{g}(\boldsymbol{\theta}))\pi(\boldsymbol{h}(\boldsymbol{\theta})|\boldsymbol{g}(\boldsymbol{\theta})).$$

So we will never find the difference and will never be able to determine the values of μ_1 and μ_2 beyond what we have specified in our prior.

When using maximum likelihood estimation, a lack of identifiability leads to ill-posed optimisation. In these cases, penalty or regularisation terms are added to the log-likelihood to remove the problems. In Bayesian inference, the priors automatically do the job for us:

$$\log[\pi(\theta|x)] = L(\theta|x) + \log[\pi(\theta)].$$

This does not mean that Bayesians should ignore this though. Identifiability issues still appear when sampling from posteriors (especially when the priors are weak), and we should always investigate dependencies between parameters in the posterior.

9.3 Linear regression

Linear regression is one of the most used and simplest statistical models. The textbook Bayesian formulation of a linear regression model is straightforward:

$$\begin{aligned}y_i|\alpha, \beta, \sigma^2, x_i &\sim \mathsf{N}(\alpha + \beta x_i, \sigma^2), \\ \alpha &\sim \mathsf{N}(0, \sigma_\alpha^2), \\ \beta &\sim \mathsf{N}(0, \sigma_\beta^2), \\ \sigma^2 &\sim \mathsf{InvGamma}(a, b),\end{aligned}$$

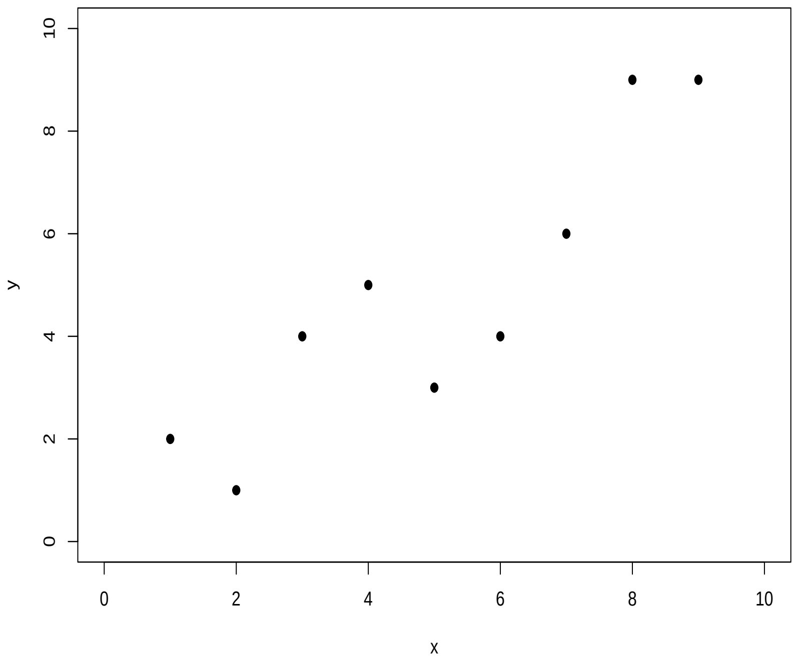
where we have used the standard conjugate form for the priors and the hyperparameters a, b, σ_α^2 and σ_α^2 need to be specified.

This model can be coded in the following way in Stan:

```
data {
  int<lower=0> N;          // num observations
  real x[N];               // observed explanatory variable
  real y[N];               // observed response variable
}
parameters {
  real alpha;             // intercept
  real beta;              // gradient
  real<lower=0> sigma_2; // error variance
}
model {
  // Prior
  alpha ~ normal(0,10000);
  beta ~ normal(0,10000);
  sigma_2 ~ inv_gamma(0.01,0.01);

  // Likelihood
  for (n in 1:N)
      y[n] ~ normal(alpha + beta*x[n], sqrt(sigma_2));
}
```

Example 9.3.1

Let's have the following data that we believe follows a straight line relationship:

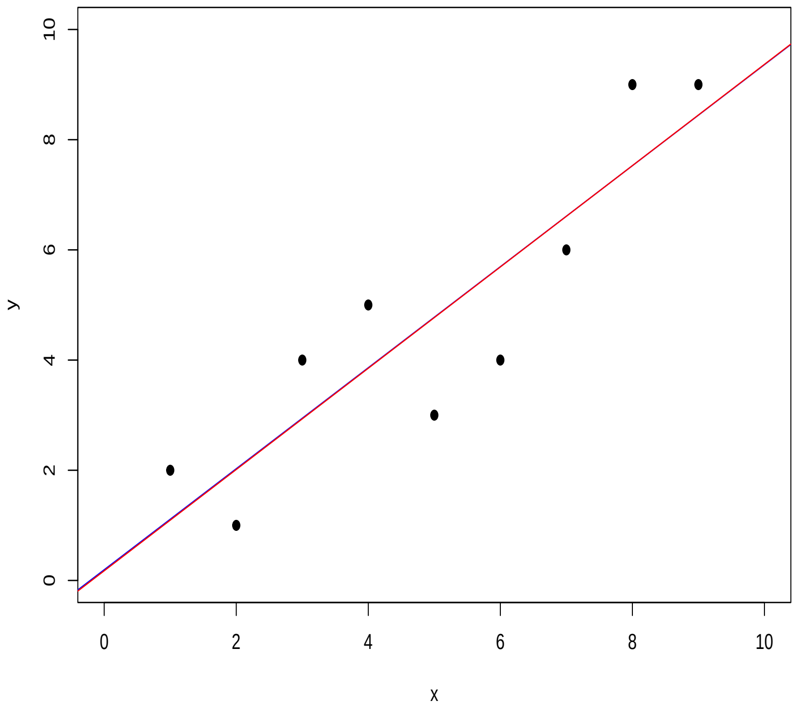


Estimating parameters using least squares fitting in R is trivial:

```
Call:
lm(formula = y ~ x, data = reg_data)

Coefficients:
(Intercept)            x
     0.1944       0.9167
```

Compare this with summaries from Stan:

```
$summary
              mean     se_mean        sd        50%      n_eff     Rhat
alpha    0.1754029 0.012460406 1.1587019  0.1859671   8647.270 1.000327
beta     0.9191603 0.002216499 0.2068693  0.9176843   8710.781 1.000408
sigma_2  2.5749959 0.028843551 1.9171726  2.0599911   4417.993 1.000472
lp__    -8.7725645 0.026130836 1.8224132 -8.3245564   4863.928 1.000254
```



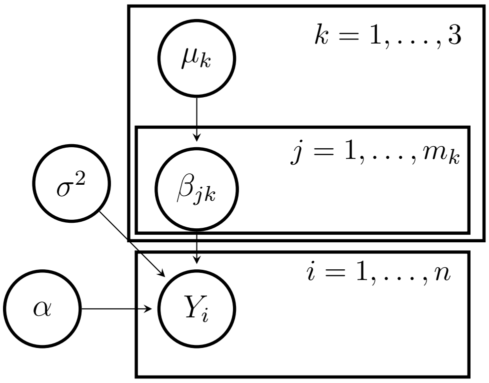
Extension to multiple linear regression (with m explanatory variables) is straightforward:

$$\begin{aligned} y_i|\alpha,\boldsymbol{\beta},\sigma^2,\boldsymbol{x}_i &\sim \mathsf{N}(\alpha+\boldsymbol{\beta}^T\boldsymbol{x}_i,\sigma^2), & i&=1,\dots,n,\\ \alpha &\sim \mathsf{N}(0,\sigma_\alpha^2), &&\\ \beta_j &\sim \mathsf{N}(0,\sigma_\beta^2), & j&=1,\dots,m,\\ \sigma^2 &\sim \mathsf{InvGamma}(a,b), && \end{aligned}$$

In Stan, we have to extend the data format to handle arrays:

```
data {
  int<lower=0> N;          // num observations
  int<lower=1> M;          // num explanatory variables
  real x[N,M];             // observed explanatory variables
  real y[N];               // observed response variable
}
```

We can also imagine building hierarchical priors in this setting if we believed that certain explanatory variables had similar effects on the response. If we had six explanatory variables with three different expected behaviours (specifically, the first four are similar and the final two are not similar to any other), we could have:



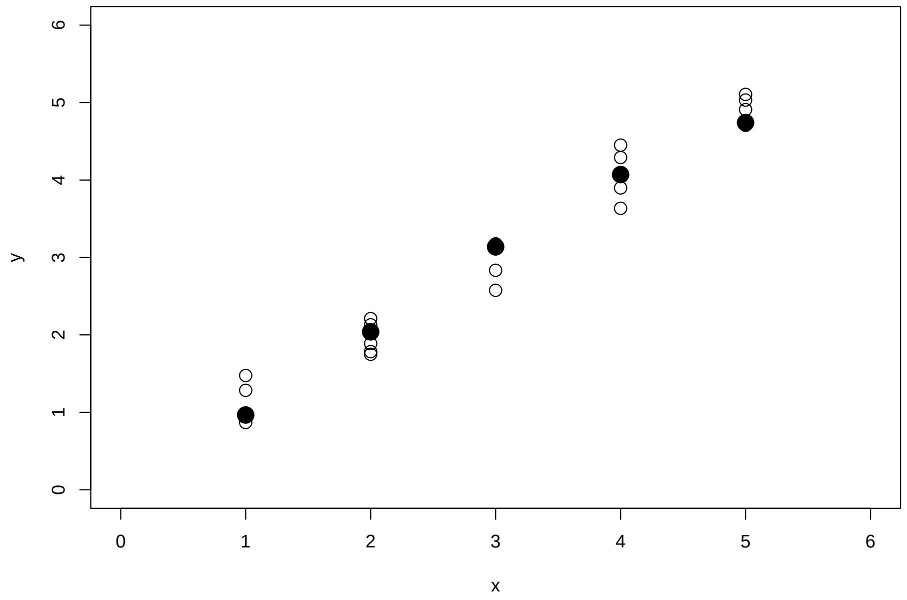
9.3.1 Errors-in-variables

Errors-in-variables regression is a very natural extension of the linear regression models of the previous section. A strong assumption in linear regression is that there is no (or relatively little) error in the recorded x values. Relaxing this assumption gives rise to the errors-in-variables model, which is simple enough to write down:

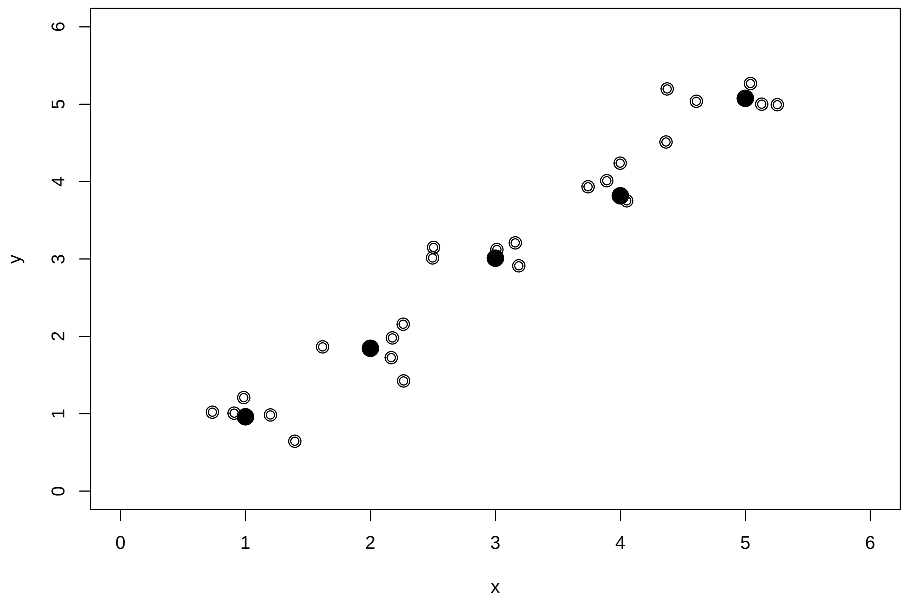
$$\begin{aligned} \widetilde{y}_i &= \alpha+\beta\widetilde{x}_i,\\ y_i|\sigma_y^2 &\sim \mathsf{N}(\widetilde{y}_i,\sigma_y^2),\\ x_i|\sigma_x^2 &\sim \mathsf{N}(\widetilde{x}_i,\sigma_x^2), \end{aligned}$$

for $i = 1, \dots, n$.

In simple linear regression, our errors act in a straightforward manner. In the following plot, we have filled dots as the "true" values and unfilled dots as possible observations of that truth.



In errors-in-variables regression, our errors come in two directions.



Thinking about the errors-in-variables model, we can see that we have an identifiability issue (dropping the i index for clarity):

$$\begin{aligned} y &= \widetilde{y} + \epsilon_y \\ &= \alpha + \beta\widetilde{x} + \epsilon_y, \\ &= \alpha + \beta\,(x - \epsilon_x) + \epsilon_y, \\ &= \alpha + \beta x + (\epsilon_y - \beta\epsilon_x). \end{aligned}$$

Now, if we have $\epsilon_x \sim \mathsf{N}(0, \sigma_x^2)$ and $\epsilon_y \sim \mathsf{N}(0, \sigma_y^2)$, then we will have

$$(\epsilon_y - \beta\epsilon_x) \sim \mathsf{N}(0, \sigma_y^2 + \beta^2\sigma_x^2).$$

There are multiple ways to attempt to solve this identifiability issue in a non-Bayesian setting. The most well known is to assume that the ratio σ_y^2/σ_x^2 is known. This would be a very strong piece of information and is enough to constrain the problem because you are effectively going from two parameters to one:

$$\sigma_y^2 = c\sigma_x^2$$

with c known.

As noted in Section 9.2, we automatically circumnavigate this issue in a Bayesian setting through the use of proper prior distributions. The Bayesian model set-up is interesting in that all the $\widetilde{y}_i$ and $\widetilde{x}_i$ become unknown parameters. We have a formula for $\widetilde{y}_i$, but we need to specify a prior distribution for the $\widetilde{x}_i$. In Stan, this can be handled in the following way:

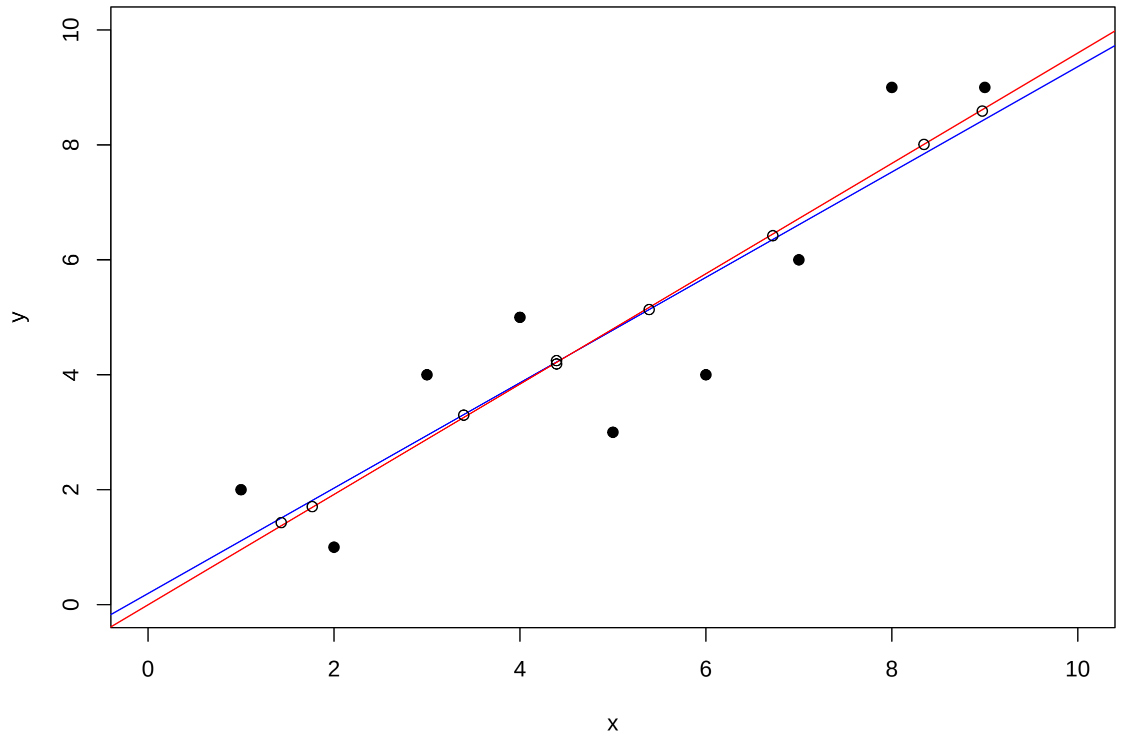
```
parameters {
  real alpha;                        // intercept
  real beta;                         // gradient
  real<lower=0> sigma_2_x;           // error variance for x
  real<lower=0> sigma_2_y;           // error variance for y
  real<lower=0,upper=10> true_x[N];  // the true values of x
}
model {
  // Prior
  alpha ~ normal(0,10000);
  beta ~ normal(0,10000);
  sigma_2_x ~ inv_gamma(0.01,0.01);
  sigma_2_y ~ inv_gamma(0.01,0.01);
  true_x ~ uniform(0,10);   // This is a vectorised form.

  // Likelihood
  for (n in 1:N){
      x[n] ~ normal(true_x[n], sqrt(sigma_2_x));
      y[n] ~ normal(alpha + beta*true_x[n], sqrt(sigma_2_y));
      }
}
```

If we also want to get out the $\widetilde{y}_i$, they can be computed in a `transformed parameters` block:

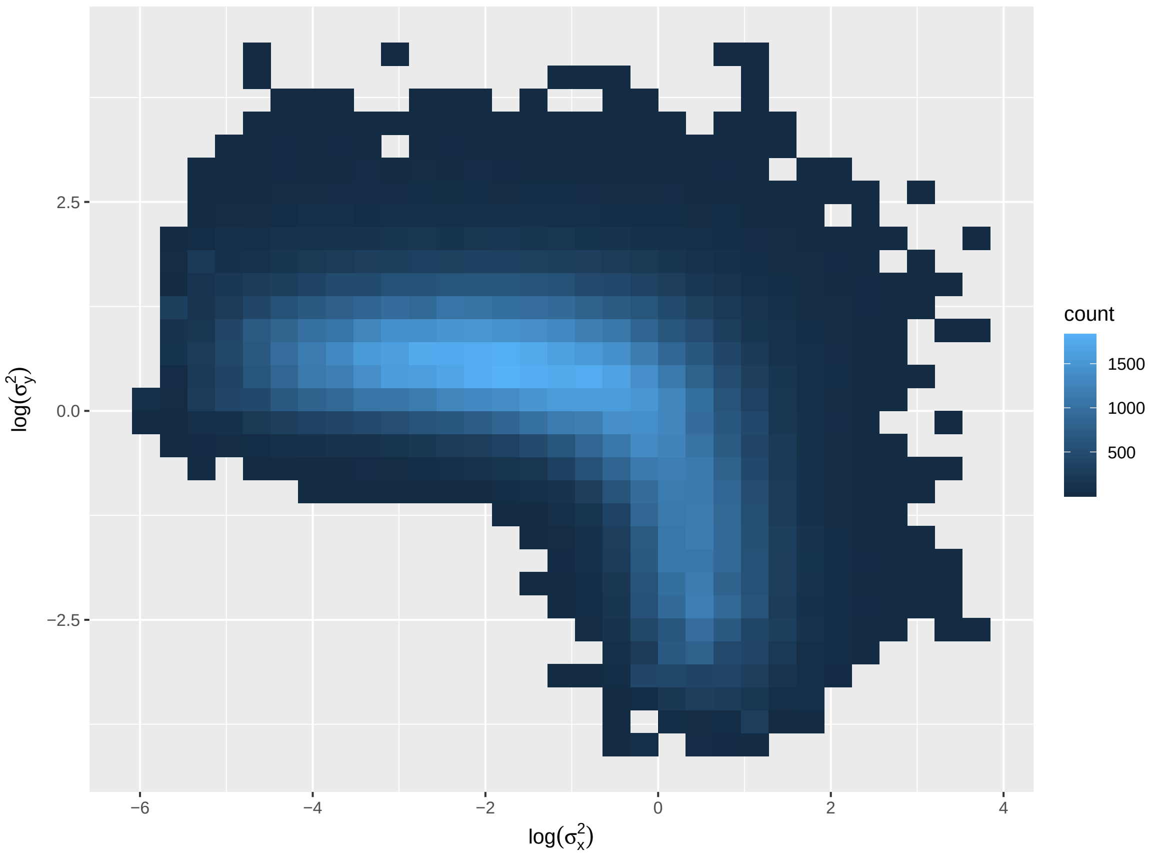
```
transformed parameters {
  real true_y[N];   // the true values of y

  for (n in 1:N)
      true_y[n] = alpha + beta * true_x[n];
}
```

Example 9.3.2

Going back to our example for simple linear regression. We may ask what would the relationship be if we applied an errors-in-variables model.



There is not a great deal of difference in the fits (probably because the errors are similar in both directions and the prior information is weak). We will see an example in the practical where it does make a real difference. The unfilled circles here are the posterior mean estimates of the true set of x and y pairs.
There is still a hangover in the posterior samples from the identifiability issue though. If we plot the 2d-density estimate for $\log(\sigma_x^2)$ and $\log(\sigma_y^2)$, there is clear dependence of the type that can make sample from the posterior problematic.



9.3.2 Logistic regression

Now, we further extend into generalised linear models. Logistic regression marries linear regression with classification through the use of a Bernoulli likelihood rather than a normal one. Let y_i be the ith observation of whether something is true (encoded as 1) or not (encoded as 0). A simple logistic regression model is as follows:

$$\begin{aligned} y_i|\alpha,\beta,x_i &\sim \text{Bernoulli}\left[g(x_i,\alpha,\beta)\right], \\ g(x_i,\alpha,\beta) &= \frac{1}{1+\exp(-\alpha-\beta x_i)}, \\ \alpha &\sim \text{N}(0,\sigma_\alpha^2), \\ \beta &\sim \text{N}(0,\sigma_\beta^2), \end{aligned}$$

where the hyperparameters have been chosen for x_i have been standardised. From earlier,

$$\frac{1}{1+\exp(-\alpha-\beta x_i)}$$

is the inverse of the logit transformation: the logistic function.

Stan has functions to help with this type of model:

```
data {
  int<lower=0> N;
  vector[N] x;
  int<lower=0,upper=1> y[N];
}
parameters {
  real alpha;
  real beta;
}
model {
  // Prior
  alpha ~ normal(0,100);
  beta ~ normal(0,100);

  // Likelihood
  y ~ bernoulli_logit(alpha + beta * x);   // This is in vectorised form.
}
```

Example 9.3.3

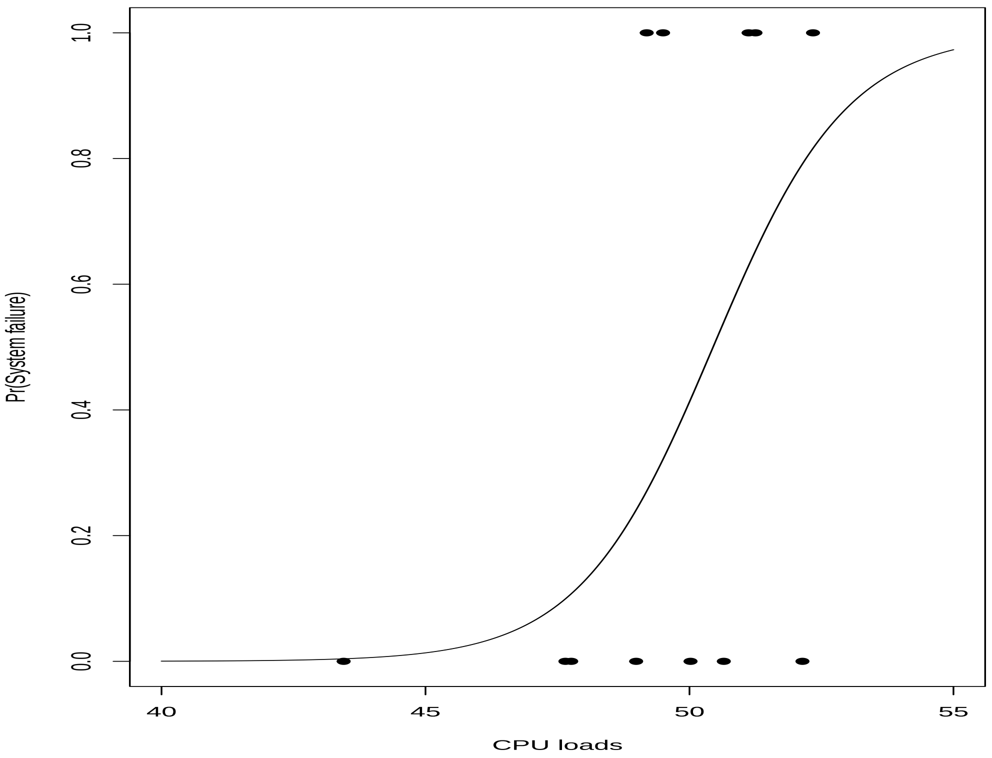
Consider the following data on system failures with respect to average CPU load.

CPU load (%)	System failure
47.65	FALSE
48.99	FALSE
50.65	FALSE
49.19	TRUE
51.25	TRUE
52.34	TRUE
52.14	FALSE
49.50	TRUE
43.45	FALSE
47.76	FALSE
50.02	FALSE
51.12	TRUE

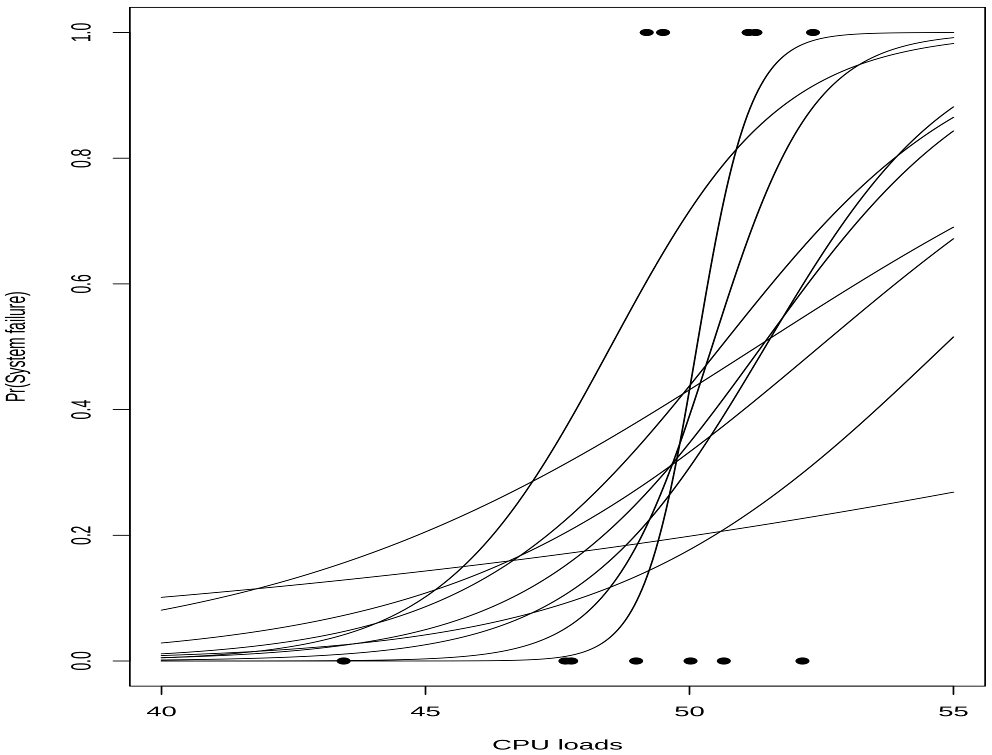
We can use the logistic regression model code to get estimates of the effect CPU load seems to have on system failure.

```
             mean     se_mean         sd
alpha -39.9551183 0.167568775 24.8039672
beta    0.7920968 0.003337940  0.4942338
```

It is more interesting to look at the logistic function with the posterior mean estimates plotted against the data.



It is even better to consider the range of relationships given in the posterior distribution:



9.4 Compositional data

Compositional data are a type of multivariate data where all the values are restricted to the interval $(0, \kappa)$ and the values for each observation must sum to κ. The value of κ is usually either 1 (when we are considering proportions) or 100 (when we are considering percentages).

Example 9.4.1

Five rocks have been analysed to check their metal composition in terms of iron, nickel and other metals:

$$X = \begin{pmatrix} 0.25 & 0.12 & 0.63 \\ 0.21 & 0.11 & 0.68 \\ 0.29 & 0.12 & 0.59 \\ 0.19 & 0.10 & 0.71 \\ 0.29 & 0.14 & 0.57 \end{pmatrix}$$

We can calculate the correlation matrix:

$$R = \begin{pmatrix} 1.00 & 0.87 & -0.99 \\ & 1.00 & -0.93 \\ & & 1.00 \end{pmatrix}$$

These data cause problems to standard statistical methods because:

1. The data are bounded unlike the most-used multivariate distributions.
2. There is at least one perfect linear relationship in the variables.
3. We are forced to have negative correlation, and this can distort interpretation.

In fact, p-dimensional compositional data are defined on a special subset of $\mathbb{R}^p$: the simplex

$$\mathcal{S}^p = \left\{ \mathbf{x} = [x_1, x_2, \ldots, x_p] \in \mathbb{R}^p \middle| x_i > 0, i = 1, 2, \ldots, p; \sum_{i=1}^{p} x_i = \kappa \right\}.$$

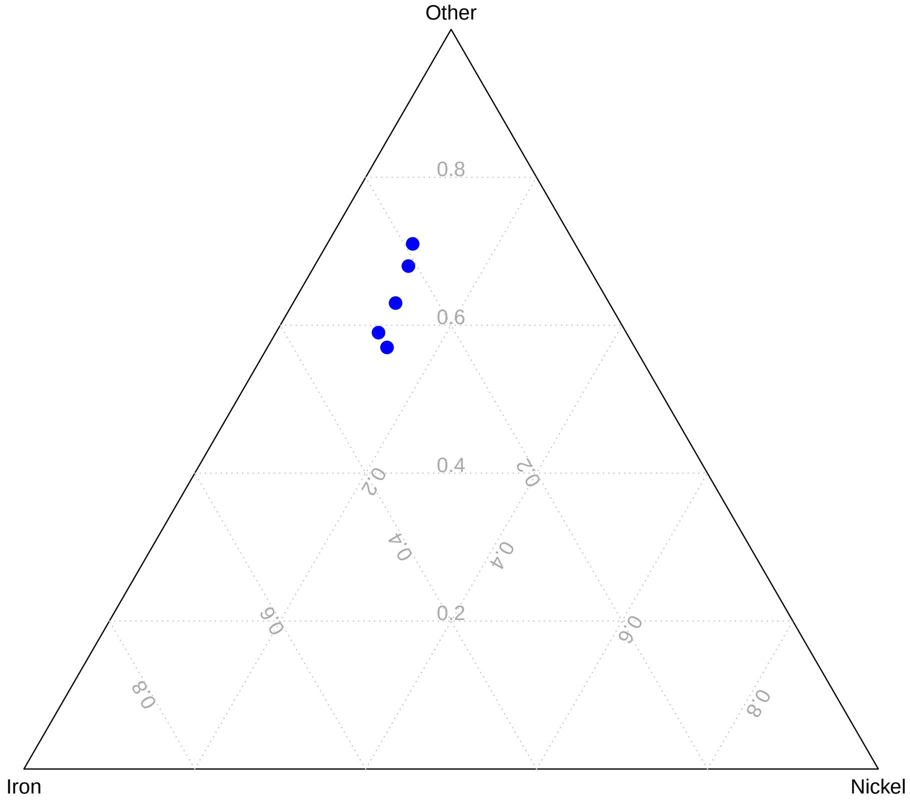
Effectively, the data points are in $p - 1$ dimensions because if we know the value for $p - 1$ of the variables, we can calculate the value of the other using:

$$x_i = \kappa - \sum_{j \neq i} x_j.$$

A common way of displaying compositional data is through a ternary plot that takes advantage of the data being constrained on the simplex. For $p = 3$, we just need a triangle.

Example 9.4.2

Here we display the data from the previous example in a ternary plot.



The archetypal distribution defined over the simplex is the *Dirichlet distribution*. If we have compositional data supported over $(p-1)$-$(0,1)$-simplex, then

$$\pi(\mathbf{x}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{p} x_i^{\alpha_i - 1}$$

where the multivariate Beta function is

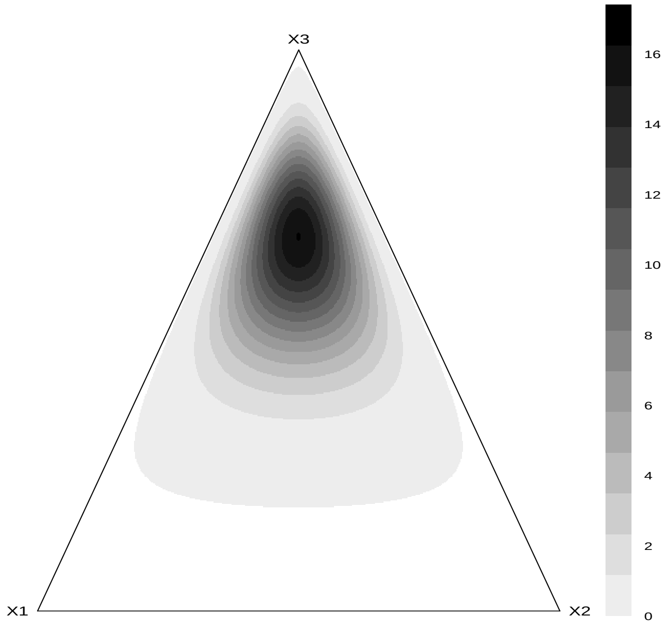
$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{p} \Gamma(\alpha_i)}{\Gamma\big(\alpha_0\big)}$$

and where

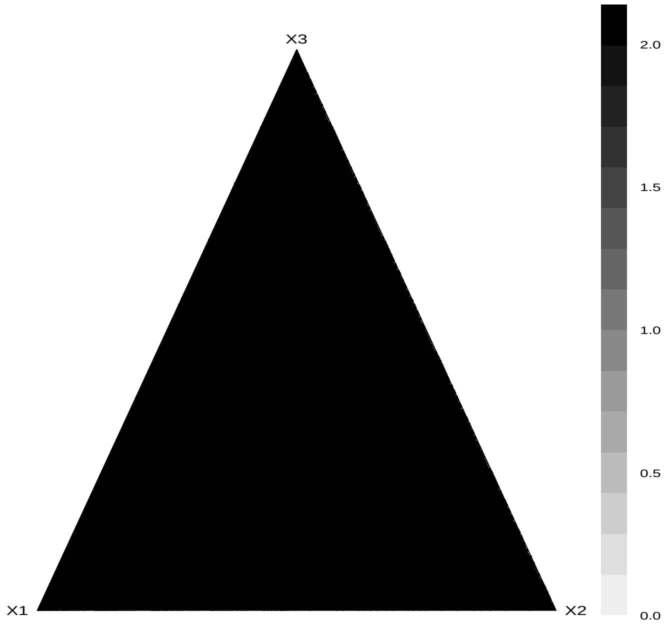
$$\alpha_0 = \sum_{i=1}^{p} \alpha_i.$$

The parameters of the Dirichlet distribution act in a very similar way to the parameters of a Beta distribution. In fact, there is a strong connection between the two as the marginal distributions of the x_i are Beta ($x_i \sim$ Be $\left(\alpha_i, \sum_{-i} \alpha_j\right)$ in fact).

Here is a 2d density plot on the simplex for a Dir$(3,3,9)$ distribution.



And here is the same for a Dir$(1,1,1)$ distribution.



Example 9.4.3

Let's imagine that we believe the data from the previous examples follow a Dirichlet distribution. We might have the following model:

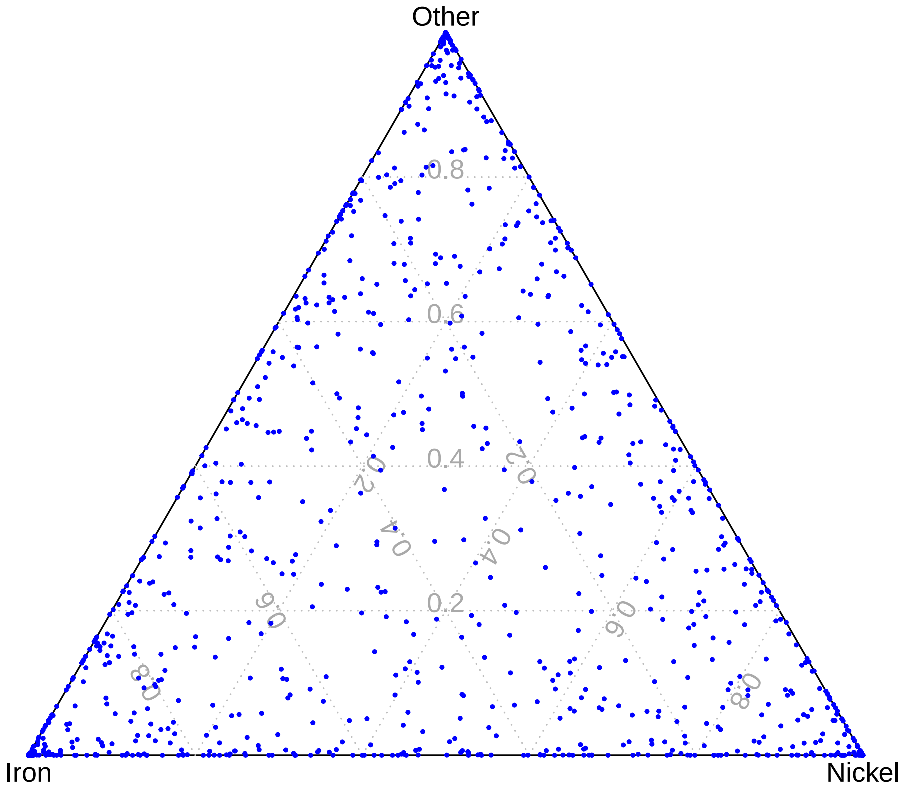
$$\begin{aligned} \mathbf{x}|\boldsymbol{\alpha} &\sim \text{Dir}(\boldsymbol{\alpha}), \\ \alpha_i &\sim \text{Gamma}(1,1) \text{ for } i = 1, \ldots, p. \end{aligned}$$

```
data {
  int<lower=0> N;
  matrix[N,3] x;
}
parameters {
  vector[3] alpha;
}
model {
  vector[3] x_vector;

  // Prior
  alpha ~ gamma(1,1);

  // Likelihood
  for (i in 1:N){
    x_vector = to_vector(x[i]);
    x_vector ~ dirichlet(alpha);
  }
}
```

We can also use Stan to investigate our preposterior for $\mathbf{x}$

```
parameters {
  vector[3] alpha;
}
model {
  // Prior
  alpha ~ gamma(1,1);
}
generated quantities {
  vector[3] x_vector;
  x_vector = dirichlet_rng(alpha);
}
```

Here is a ternary plot with 1000 samples from the preposterior distribution.

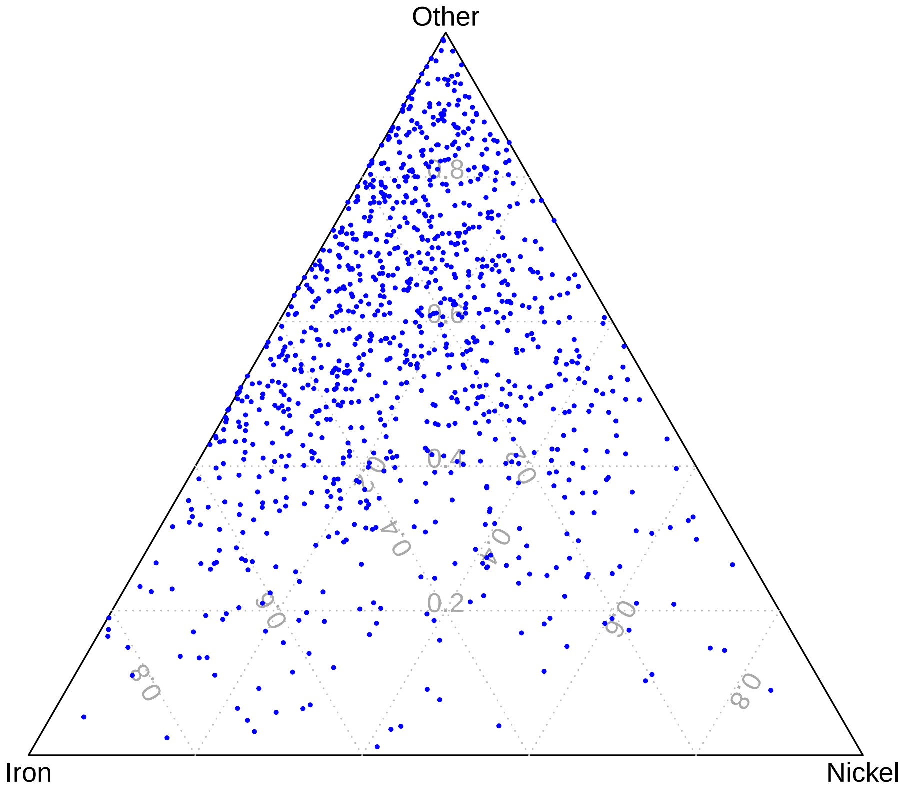


Bringing the data in, we get posterior statistics of $(E)(\alpha_1) = 1.93(0.78)$, $(E)(\alpha_2) = 1.18(0.48)$ and $(E)(\alpha_3) = 4.15(1.70)$, where the posterior standard deviations are given in the brackets.

We can add the following code to the bottom of our posterior sampling code to get a similar sample for our predictive distribution.

```
generated quantities {
  vector[3] x_;
  x_ = dirichlet_rng(alpha);
}
```

Here is a ternary plot with 1000 samples from the predictive distribution.



By using distributions defined over the simplex, we lose the great number of techniques that have been devised for unbounded or multivariate normal datasets. The data are effectively in $p-1$ dimensions so a transformation would be useful. However, a simple linear transformation would not remove the problems with bounds or forced correlation.

There are many different transformation that can be used, but we will focus on the additive log-ratio:

$$\begin{aligned} \mathbf{y} &= (y_1,\ldots,y_{p-1})^T = \mathsf{alr}(\mathbf{x}) \\ &= \left[\log\left(\frac{x_1}{x_p}\right),\ldots,\log\left(\frac{x_{p-1}}{x_p}\right)\right]^T \end{aligned}$$

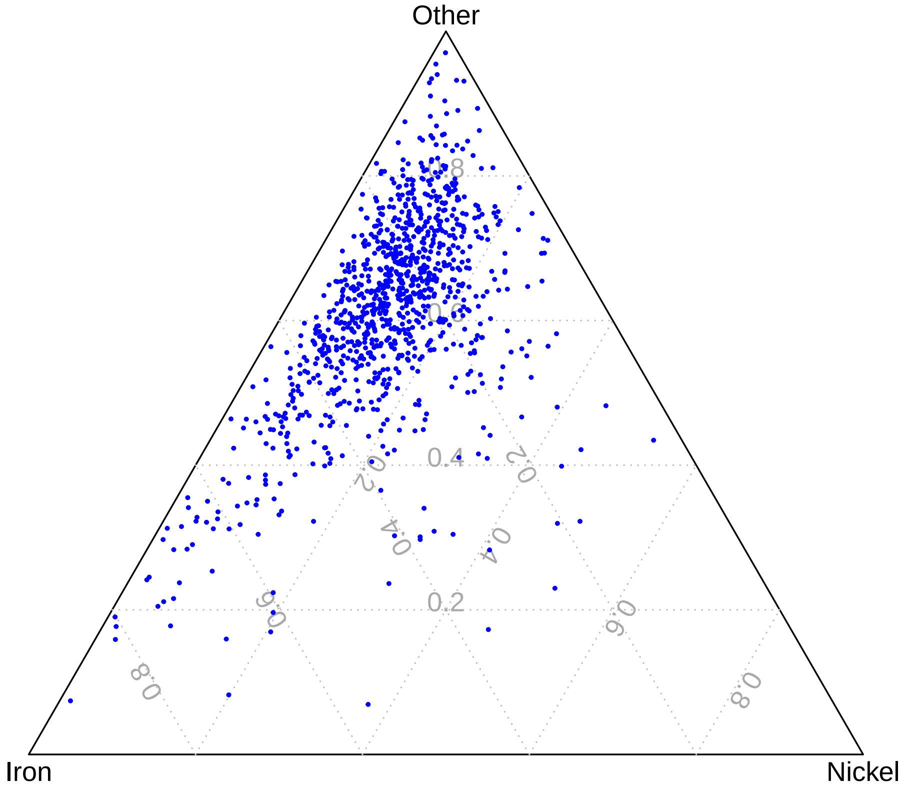
(here, $\mathbf{y} \in \mathbb{R}^{\,p-1}$ and note that the ordering of the variables is arbitrary). This transformation tends to be useful because it directly utilises the fact that compositional data gives us information about relative size alone. We may then proceed to use methods and distributions that are defined for unbounded real spaces. We need to decide which variable to choose as the divisor and in some cases this is straightforward if there is a clear interpretation. For instance, all types of household expenditure relative to food expenditure.

Example 9.4.4

We can apply the ALR transformation to our data within Stan:

```
data {
  int<lower=0> N;
  matrix[N,3] x;
}
transformed data {
  matrix[N,2] y;
  for (i in 1:N){
    for (j in 1:2){
      y[i,j] = log(x[i,j]/x[i,3]);
    }
  }
}
parameters {
  vector[2] mu;
  cov_matrix[2] Sigma;
}
model {
  vector[2] y_vector;

  // Prior
  mu ~ normal(0,100);
  Sigma ~ inv_wishart(2, diag_matrix(rep_vector(1.0, 2)));

  // Likelihood
  for (i in 1:N){
    y_vector = to_vector(y[i]);
    y_vector ~ multi_normal(mu, Sigma);
  }
}
generated quantities {
  vector[2] y_;
  vector[3] x_;
  real unnormalised_sum;

  y_ = multi_normal_rng(mu, Sigma);
  unnormalised_sum = exp(y_[1]) + exp(y_[2]) + 1;
  x_[1] = exp(y_[1])/unnormalised_sum;
  x_[2] = exp(y_[2])/unnormalised_sum;
  x_[3] = 1/unnormalised_sum;
}
```

Here is a ternary plot with 1000 samples from the predictive distribution.



These kind of constraints come up more often when considering priors for stochastic vectors. A *stochastic vector* is a vector whose elements must sum to one. We may have already seen these as parameters in a multinomial distribution or as rows in a right-stochastic matrix when considering discrete Markov chains.

A multinomial distribution for $x_1, \dots, x_p$ observed after n trials has the following probability mass function:

$$\pi(\mathbf{x}|\boldsymbol{\theta}) = \frac{n!}{x_1! \cdots x_p!} \prod_{i=1}^{p} \theta_i^{x_i}$$

where the $\theta_i > 0$ and $\sum \theta_i = 1$. The Dirichlet distribution works as a conjugate prior for $\boldsymbol{\theta}$ when we have multinomial data:

$$\begin{aligned} \pi(\boldsymbol{\theta}|\mathbf{x}) &\propto \prod_{i=1}^{p} \theta_i^{x_i} \prod_{i=1}^{p} \theta_i^{\alpha_i - 1} \\ &\propto \prod_{i=1}^{p} \theta_i^{\alpha_i + x_i - 1}. \end{aligned}$$

So a Dir$(\alpha_1, \dots, \alpha_p)$ prior for $\boldsymbol{\theta}$ becomes a Dir$(\alpha_1 + x_1, \dots, \alpha_p + x_p)$ posterior.

Example 9.4.5

We are to receive data about eye colour in the form of the number of observations in each of four mutually exclusive categories: [Amber, Brown, Hazel], [Blue], [Green], [Other]. *A priori*, we believe that, in the population we will study, that the first two categories are much more likely so we posit a Dir$(10, 10, 1, 1)$ prior.
The mean for each component of a Dirichlet distribution is

$$\mathsf{E}(\theta_i) = \frac{\alpha_i}{\sum \alpha_j}.$$

So our prior means are $\mathsf{E}(\theta_1) = \mathsf{E}(\theta_2) = 0.45$ and $\mathsf{E}(\theta_3) = \mathsf{E}(\theta_4) = 0.05$ to 2 d.p..
We receive a sample of 150 eye colours: 49, 62, 13, 26 for each category.
By conjugacy, we have a Dir$(59, 72, 14, 27)$ posterior, and our posterior means (to two decimal places) are

$$\begin{aligned}
\mathsf{E}(\theta_1|\mathbf{x}) &= 0.34, \\
\mathsf{E}(\theta_2|\mathbf{x}) &= 0.42, \\
\mathsf{E}(\theta_3|\mathbf{x}) &= 0.08 \text{ and} \\
\mathsf{E}(\theta_4|\mathbf{x}) &= 0.16
\end{aligned}$$

Chapter 10

Missing data

10.1 Introduction

Strategies for dealing with missing data take up a lot of space in the literature. It is easy to see why: experiments fail, surveys are not returned, files get corrupted etc..

But does it matter if some data are missing? When we update our priors given the likelihood, we use the data that we have seen. The data that we are yet to see can be dealt with by utilising the posterior predictive distribution. However, what if the data we have not seen are not exchangeable with the data that we have seen?

There are three common classifications for missing data:

(1) Missing Completely at Random

The data that are missing has no describable pattern. From one data observation to the next, there is no way that we will know how likely a missing value is. This is a strong assumption and a personal judgement about how the missing data should be treated.

(2) Missing at Random

In this case, we can describe the process for the occurrence of missing through a stochastic mechanism. For instance, we may know that there is a 5% chance that a data point will not be recorded regardless of the value. We can model this, but, when we have independent data, it is not worth it.

The missing at random situation is related to what you saw in the first half of the course with hidden Markov models and what you saw in the last chapter on errors-in-variables. In both of those cases, we had unknown latent variables with a clear statistical model to explain their relationship to what we can see.

(3) Missing Not at Random

Here, there is some rule or decision that has been taken to exclude certain data. Two key examples are

1. we have no data for a subpopulation,

2. we have not been able to measure something because of threshold effect.

The former can be handled using hierarchical models and the latter gives us censored data.

Example 10.1.1

Imagine a situation where we are expecting ten observations of count data. We receive the following:

12, missing, 9, 12, 8, 10, missing, missing, 7, 12.

If we have the following model,

$$\begin{aligned} X_i|\lambda &\sim \text{Po}(\lambda), \quad i = 1, \ldots, 10, \\ \lambda &\sim \text{Gamma}(\alpha, \beta), \end{aligned}$$

then we can calculate the posterior whilst treating the missing values as unknown parameters:

$$\begin{aligned} \pi(\lambda, \underline{x}_{\text{miss}}|\underline{x}_{\text{obs}}) &\propto \pi(\lambda, \underline{x}_{\text{miss}}, \underline{x}_{\text{obs}}) \\ &\propto \pi(\underline{x}_{\text{obs}}|\lambda)\pi(\underline{x}_{\text{miss}}|\lambda)\pi(\lambda). \end{aligned}$$

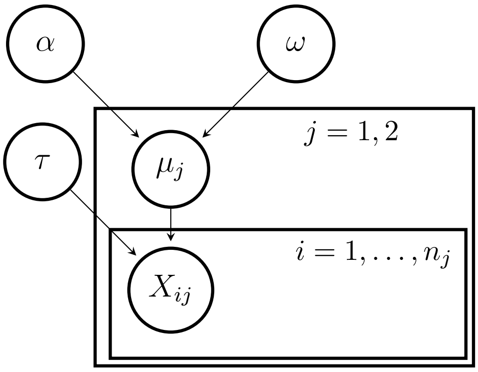
Now, we remove the nuisance parameters by summing over possible values of the missing variables:

$$\begin{aligned} \pi(\lambda|\underline{x}_{\text{obs}}) &\propto \pi(\underline{x}_{\text{obs}}|\lambda)\pi(\lambda) \sum_{x_2=0}^{\infty} \sum_{x_7=0}^{\infty} \sum_{x_8=0}^{\infty} \pi(\underline{x}_{\text{miss}}|\lambda) \\ &= \pi(\underline{x}_{\text{obs}}|\lambda)\pi(\lambda) \sum_{x_2=0}^{\infty} \pi(x_2|\lambda) \sum_{x_7=0}^{\infty} \pi(x_7|\lambda) \sum_{x_8=0}^{\infty} \pi(x_8|\lambda) \\ &= \pi(\underline{x}_{\text{obs}}|\lambda)\pi(\lambda) \end{aligned}$$

So considering the missing data at all was a waste of time.

Example 10.1.2

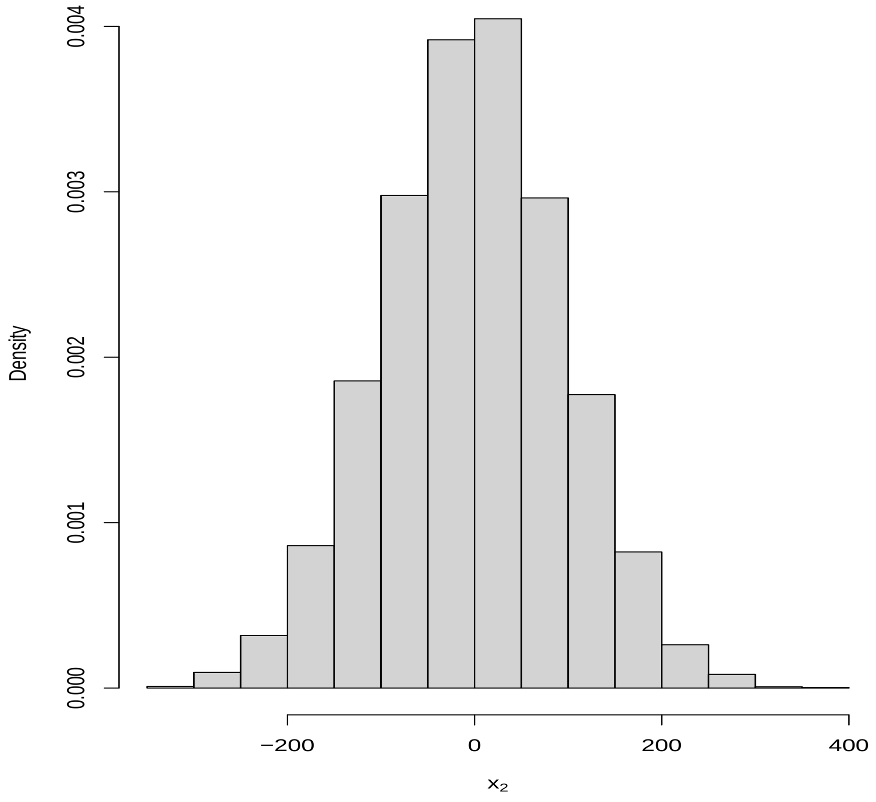
It might be unethical to experiment on a subpopulation. However, we have clear results for another subpopulation, and we strongly suspect that the results will carry over. This can be dealt with directly using hierarchical models.



We can imagine that we do not observe any X_{i2}, but we can use our model to look at the predictive distribution for this unseen population.

```
data {
  int<lower=0> N;
  real x1[N];
}
parameters {
  real mu[2];
  real alpha;
  real<lower=0> tau;
  real<lower=0> omega;
}
model {
  // Prior
  tau ~ gamma(10,1);
  alpha ~ normal(0,100);
  omega ~ gamma(10,1);
  mu ~ normal(alpha,sqrt(1/omega));

  // Likelihood
  for (i in 1:N){
    x1[i] ~ normal(mu[1],sqrt(1/tau));
  }
}
generated quantities {
  real x2;
  x2 = normal_rng(mu[2],sqrt(1/tau));
}
```

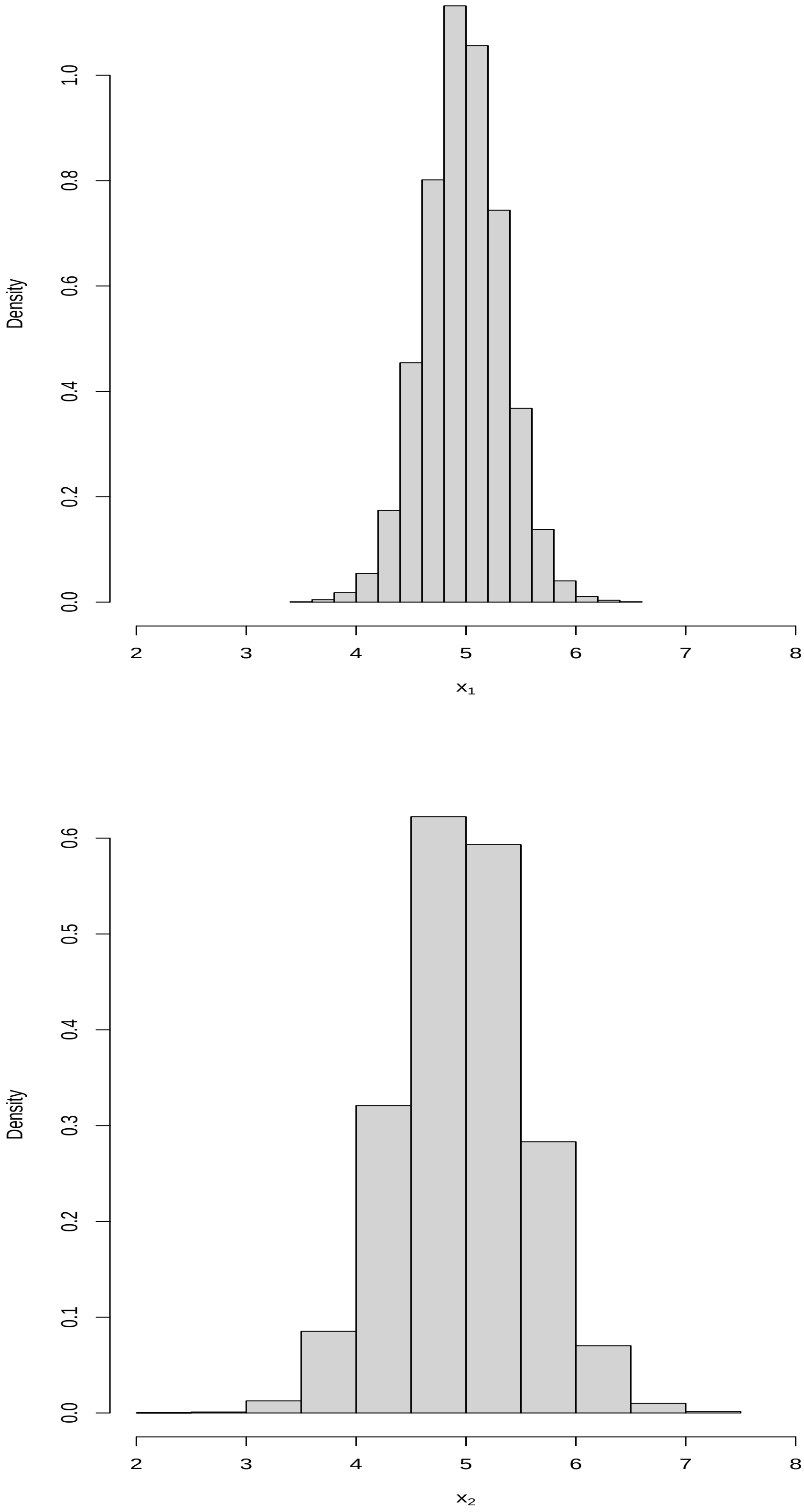
If we comment out the likelihood in the Stan model, we get a sample of X_{i2} from the preposterior distribution.



We use the following code to sample from the predictive distributions using the model:

```
fit <- sampling(model,
                data = list(N = 12,
                            x1 = c(5.3,5.1,4.8,4.5,
                                   5.5,5.2,5.0,5.0,
                                   5.1,4.6,4.3,5.3)),
                iter = 10000)
```

So we expect our predictive distribution to be centred in the vicinity of 5 because we have not explicitly modelled any expected bias between the two subpopulations.


Density
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5
6
7
8
x_1
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
3
4
5
6
7
8
x_2

10.2 Missing explanatory variables

A more interesting situation arises when we are modelling the relationship between response and explanatory variables. In particular, what if some of the explanatory measurements are missing?

Example 10.2.1

Let's consider simple linear regression:

$$\begin{aligned}
y_i|\alpha, \beta, \sigma^2, x_i &\sim \mathsf{N}(\alpha + \beta x_i, \sigma^2), \\
\alpha|\sigma^2_\alpha &\sim \mathsf{N}(0, 10000), \\
\beta|\sigma^2_\beta &\sim \mathsf{N}(0, 10000), \\
\sigma^2 &\sim \mathsf{InvGamma}(0.01, 0.01),
\end{aligned}$$

We have the following information:

x	y
1.02	2.67
1.52	3.45
1.89	4.49
1.91	4.50
2.51	5.62
2.62	5.70
?	2.21
?	3.45

If we are going to treat the missing values as unknown, we need to put a prior distribution on them:

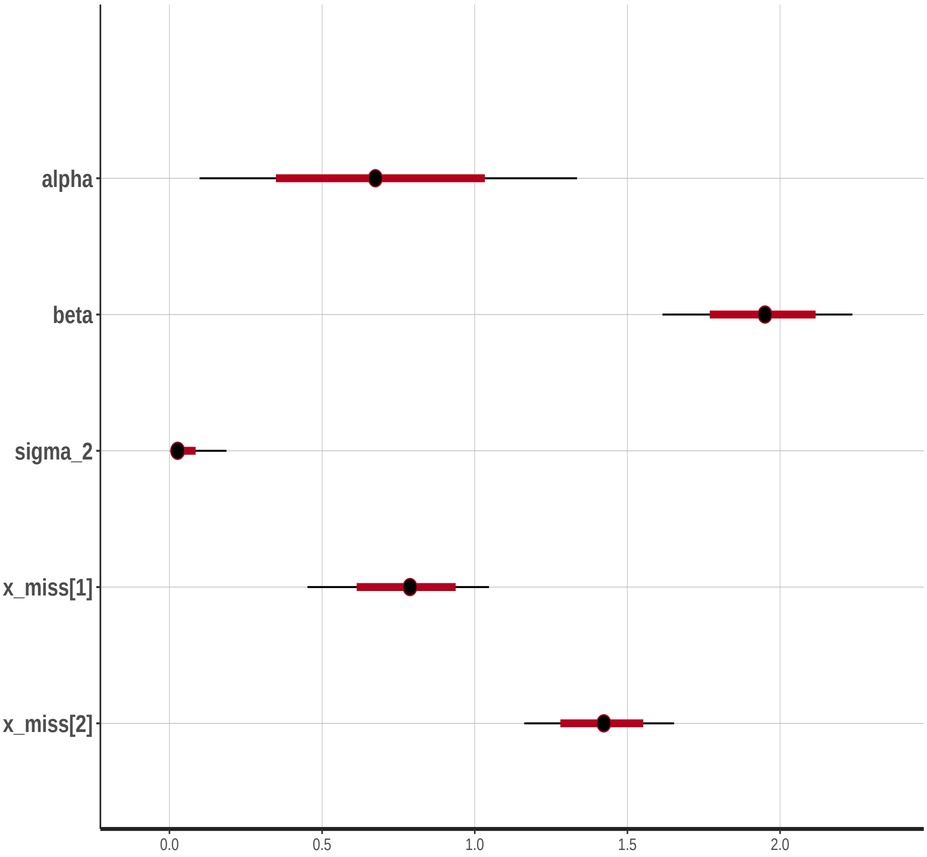
$$x_i \sim \mathsf{Uniform}(0, 5), \quad i = 7, 8.$$

The first blocks in the Stan model become:

```
data {
  int<lower=0> N;   // num observations
  int<lower=0> M;   // num missing
  real x[N-M];      // observed explanatory variable
  real y[N];        // observed response variable
}
parameters {
  real alpha;                        // intercept
  real beta;                         // gradient
  real<lower=0> sigma_2;             // error variance
  real<lower=0,upper=5> x_miss[M];  // missing explanatory variables
}
```

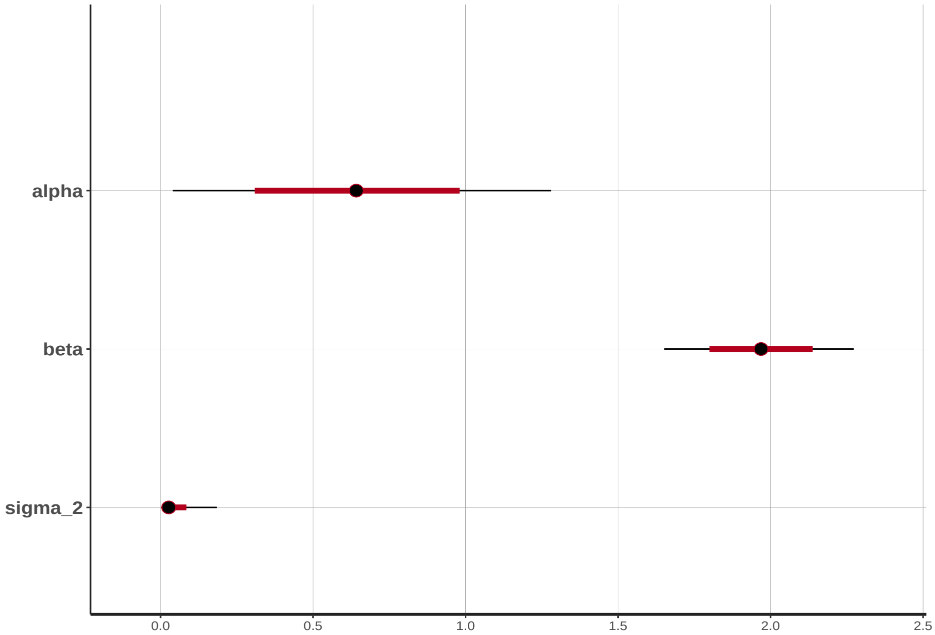
And our model becomes:

```
model {
  // Prior
  alpha ~ normal(0,100);
  beta ~ normal(0,100);
  sigma_2 ~ inv_gamma(0.01,0.01);
  x_miss ~ uniform(0,5);

  // Likelihood
  for (n in 1:(N-M))
      y[n] ~ normal(alpha + beta*x[n], sqrt(sigma_2));
  for (n in 1:M)
      y[N-M+n] ~ normal(alpha + beta*x_miss[n], sqrt(sigma_2));
}
```

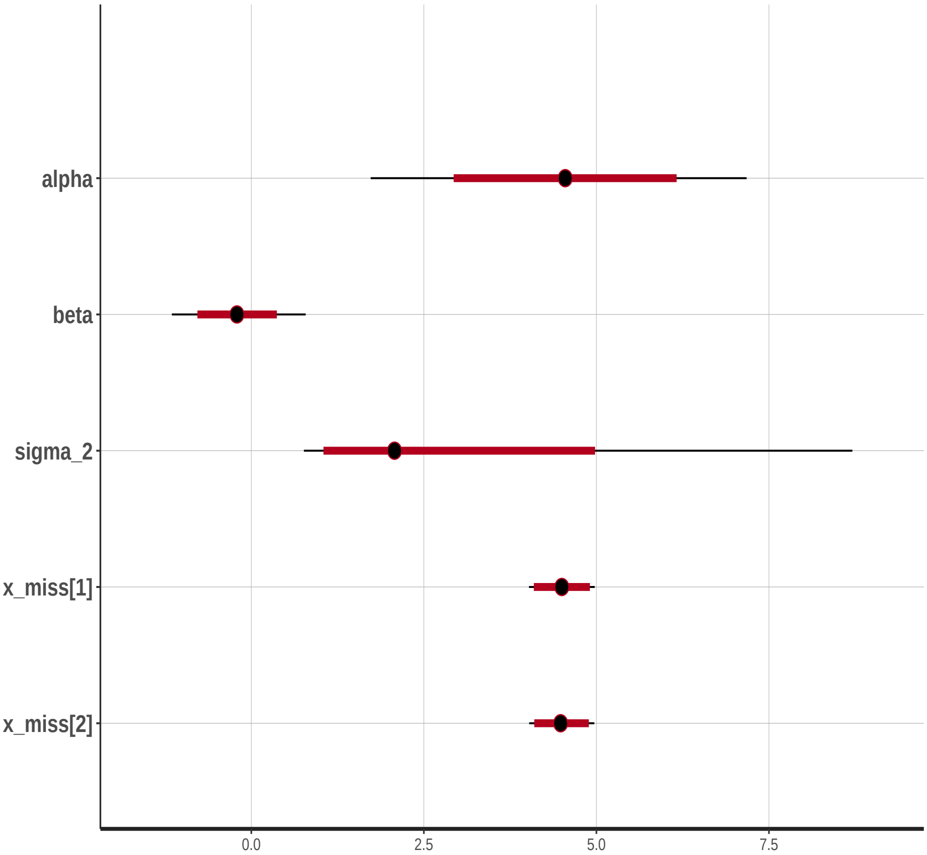


In the plot above, the thick red lines are 80% credible intervals and the thin black lines are 95% credible intervals.

Was there any point in including the missing data? Here are the marginal posteriors ignoring the data rows with missing x.



But what if we know the missing x should have been in [4,5]?

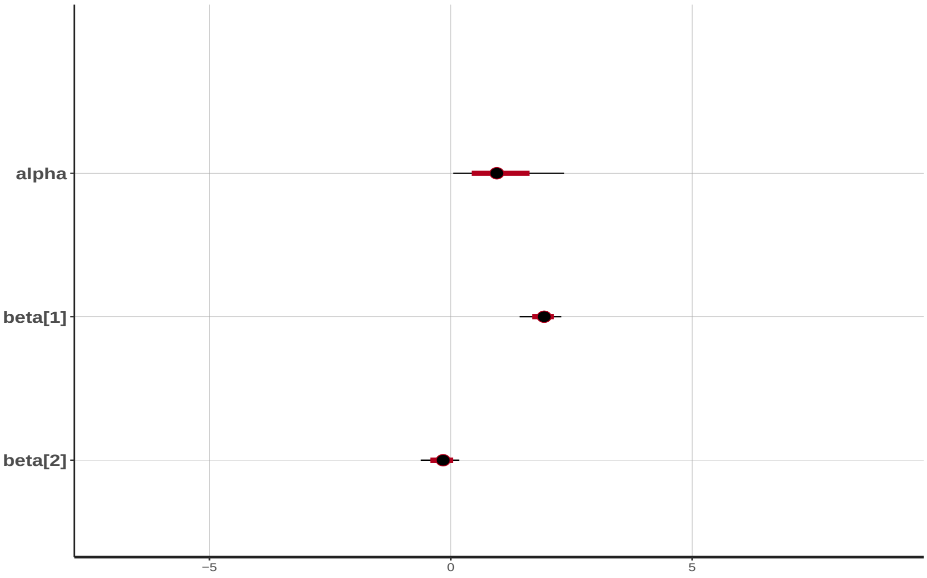


Example 10.2.2

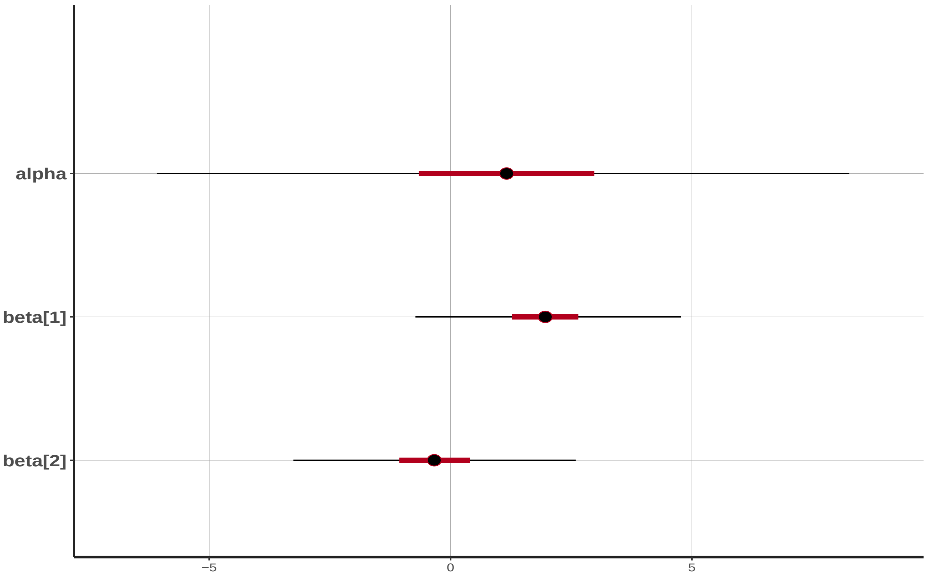
If we start ignoring missing data with the following data, we will lose half of the observations.

x_1	x_2	y
1.02	?	2.67
1.52	2.01	3.45
1.89	1.02	4.49
1.91	1.35	4.50
2.51	?	5.62
2.62	1.75	5.70
?	4.25	2.21
?	4.00	3.45

```
data {
  int<lower=0> N;  // num observations
  int<lower=0, upper=2> m[N];  // missing indicator
  real x[N,2];     // observed explanatory variable (-9999 for missing)
  real y[N];       // observed response variable
}
parameters {
  real alpha;                        // intercept
  real beta[2];                      // gradient
  real<lower=0> sigma_2;             // error variance
  real<lower=0,upper=5> x_full[N,2];  // complete data
}
transformed parameters {
  real mu[N];
  for (n in 1:N) {
    if (m[n] == 1) {
        mu[n] = alpha + beta[1]*x_full[n,1]+beta[2]*x[n,2];
    }
    else if (m[n] == 2) {
        mu[n] = alpha + beta[1]*x[n,1]+beta[2]*x_full[n,2];
    }
    else {
        mu[n] = alpha + beta[1]*x[n,1]+beta[2]*x[n,2];
    }
}
model {
  // Prior
  alpha ~ normal(0,100);
  beta ~ normal(0,100);
  sigma_2 ~ inv_gamma(0.01,0.01);
  for (i in 1:N) {
      for (j in 1:2) {
        x_full[i,j] ~ uniform(0,5);
      }
  }

  // Likelihood
  for (n in 1:N)
      y[n] ~ normal(mu[n], sqrt(sigma_2));
}
```



And, if we ignore the rows with the missing data...



10.3 Censored data

There are many situations when studying time to failures, recoveries and transitions where the items being studied do nothing during the observation period. One example is waiting for components to fail during stress testing. This is an example of *right censored data*.

Left censored data occur when the data point falls before our observation period. For example, the component has failed before we have started our observations.

Note that censored data are not the same as data coming from a truncated distribution. The latter gives zero probability to any data beyond the range of the truncation interval.

Imagine that we have times of failures from the following model,

$$X_i|\lambda \sim \text{Exp}(\lambda), \quad i = 1, \ldots, n,$$

for the n observations that occurred during our observation period $(0, T)$, and we have m experimental units yet to return as failures when we get to time T. When we have censoring, the likelihood is a mix of standard data-generating probability density functions and cumulative density functions:

$$\begin{aligned} \pi(\text{Data}|\lambda) &\propto \prod_{i=1}^{n} \pi(x_i|\lambda) \prod_{i=1}^{m} \left[\int_T^\infty \pi(x|\lambda) dx\right] \\ &\propto \lambda^n \exp\left(-\lambda \sum x_i\right) \exp\left(-\lambda T\right)^m . \end{aligned}$$

In Stan, we can handle this type of likelihood directly by using `target +=`:

```
model {
  // Likelihood
  target += n*log(lambda) - lambda*sum(x) - m*lambda*T;
}
```

Or we can use the original model as usual and treat the censored data as parameters:

```
parameters {
  real<lower=0> lambda;
  real<lower=T> x_censored[m];
}
model {
  // Likelihood
  x ~ exp(lambda);              // in vectorised form
  x_censored ~ exp(lambda);     // in vectorised form
}
```

Example 10.3.1

We are modelling time to failure of a certain component in hours using the model given in this section. We will stop observing after 100 hours.

A priori, we believe that the mean failure time will be about 80 hours (probably in the range 60-100). This means that we believe that the failure rate is about $1/80 = 0.0125$ (with a range of 1/100-1/60). Of course, the rate needs to be positive so we believe that a Gamma distribution with mode

$$\frac{\alpha-1}{\beta}$$

and variance

$$\frac{\alpha}{\beta^2}$$

would be adequate. Setting the prior mode to be 1/80 and a four standard deviation range to be 4/600 (a standard deviation of 0.00167), we can rearrange the formulae to get $\alpha = 58.3$ and $\beta = 4580$ to three significant figures.

In the 100 hours, we see seven failures at times 15.4, 18.5, 60.7, 80.54, 84.12, 91.11 and 91.13. By conjugacy, we have

$$\lambda|\underline{x} \sim \text{Gamma}(65.3, 5020),$$

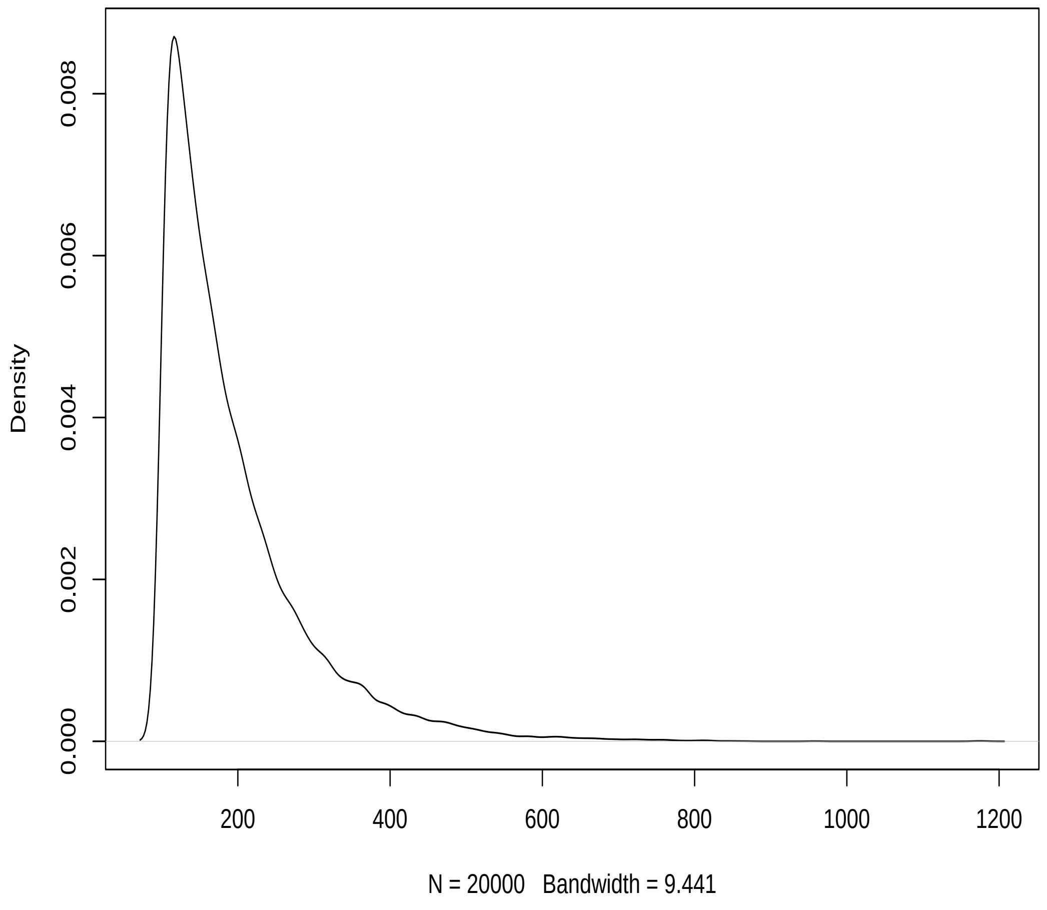
which gives us a posterior mode of 0.0130 and standard deviation of 0.00161.

If we know that there were ten additional units that did not fail during the 100 hours, then we can compute use Stan to get at the posterior for λ.

```
# Extract the posterior samples
post_sample <- extract(fit)

# Approximate the posterior mode for lambda
post_sample$lambda[which.max(post_sample$lp__)]

# Calculate the posterior standard deviation
sd(post_sample$lambda)
```

We get an approximate posterior mode of 0.0110 and standard deviation of 0.00135. We can also get an impression of when the remaining units might have failed using the posterior sample for the censored data.

```
# Kernel density estimation for x[1]
plot(density(post_sample$x_[,1]), main = NULL)
```


Density
0.000
0.002
0.004
0.006
0.008
200
400
600
800
1000
1200
N = 20000 Bandwidth = 9.441

Chapter 11

Model selection

11.1 Our modelled world

Let us have a set of k models that we will consider in our analyses:

$$\underline{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}\,.$$

To be clear, a model in this context is a joint probability distribution over data and parameters of interest. If we were to specify a different prior for a parameter and keep the same likelihood structure, we would be dealing with a different model. The purpose of our modelling may be to predict what might occur if we were to have more data. For simplicity, let us assume that these future data are denoted $\mathbf{x}$.

Given $\underline{\mathcal{M}}$, we will be operating in one of the following regimes:

(1) M-closed

We believe the true data generating process is in an element of $\underline{\mathcal{M}}$ with the prior distributions set up so as not to rule out the true parameterisation. In this case, we have a simple equation for our predictions of $\mathbf{x}$:

$$\pi(\mathbf{x}) = \sum_{i=1}^{k} \pi\left(\mathbf{x}|\mathcal{M}_i\right)\pi\left(\mathcal{M}_i\right)$$

by the law of total probability. As we collect more data, we will get $\pi\left(\mathcal{M}_j\right)$ getting closer to one if $\mathcal{M}_j$ is the true model.

(2) M-complete

We believe that a true data generating process exists, and, despite us being able to conceptualise it, we cannot put it in a from that makes the model accessible to us for computational purposes. However, in this case, we assume $\underline{\mathcal{M}}$ include the best models that we could currently utilise.

Here, we cannot really use the same formula as in the M-closed scenario because assigning prior probabilities for the models does not really make sense because we know that none of them is the true model. Therefore, we are restricted to reporting $\pi\left(\mathbf{x}|\mathcal{M}_i\right)$ for each of our models and focus shifts to evaluating individual model performance.

(3) M-open

The reality is that we are almost always in a situation where we know the true model is not in $\underline{\mathcal{M}}$. Again, assigning prior probabilities for the models does not make sense, and we are left trying to evaluate predictive performance.

The solutions to handling (2) and (3) are not entirely satisfactory. In the remainder of this chapter, we will be focussing on the M-closed situation.

11.2 Nested models

The ideal situation is for us to have our alternative models contained within an overarching model. Nested models are usually used to describe regression models where simpler models (with fewer explanatory variables and interactions) are within an overarching model containing all variables and possible interactions. In our context, we want all the models in $\underline{\mathcal{M}}$ to be a special case of a all-encompassing model.

Example 11.2.1

Imagine that we have two models:

$\mathcal{M}_1$:

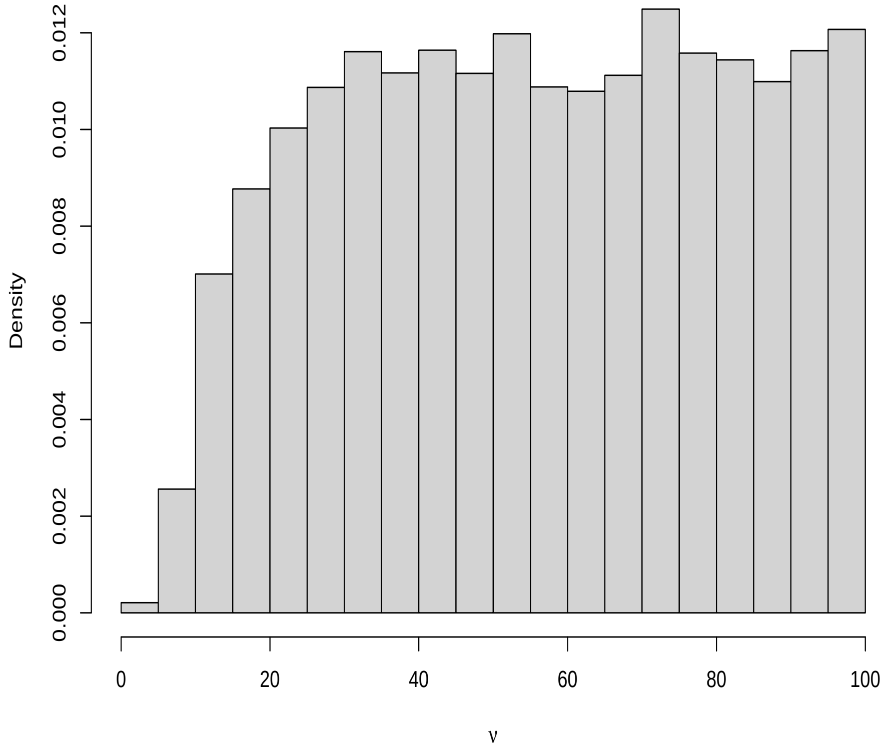
$$\begin{aligned} X_i|\mu,\sigma &\sim \mathsf{N}(\mu,\sigma^2), \quad i=1,\ldots,n, \\ \pi(\mu,\sigma); & \end{aligned}$$

$\mathcal{M}_2$:

$$\begin{aligned} X_i|m,\nu,s &\sim \mathsf{t}_\nu(m,s), \quad i=1,\ldots,n, \\ \pi(m,\nu,s). & \end{aligned}$$

As $\nu \to \infty$, $\mathcal{M}_2 \to \mathcal{M}_1$ (provided that the priors on the parameters are consistent). In fact, by the time $\nu = 30$, we are getting close to parity.

Now, let's imagine that we have collected 100 observations of x (that are taken from a normal distribution), and we sample from our posterior for ν under $\mathcal{M}_2$. We get the following histogram:



This would seem to give support to $\mathcal{M}_1$ because the ν are mainly favoured over values that would mean that the t-distribution and the normal would be almost indistinguishable.

A direct way to include seemingly non-compatible data-generating models within a single overarching model is to use a mixture distribution:

$$\begin{aligned} Z_i|\boldsymbol{\theta} &\sim \text{Categorical}(\boldsymbol{\theta}), \quad i=1,..,n, \\ X_i|z_i=1,\boldsymbol{\alpha}_1 &\sim G_1(\boldsymbol{\alpha}_1), \quad i=1,..,n, \\ &\vdots \\ X_i|z_i=k,\boldsymbol{\alpha}_k &\sim G_k(\boldsymbol{\alpha}_k), \quad i=1,..,n, \\ \pi(\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k,\boldsymbol{\theta}). \end{aligned}$$

In this formulation, the z_i are latent variables that indicate which model the data have been drawn from. In practice, we will be uncertain about which model gave rise to the data and we will need to integrate out those variables, and we need to consider the interplay between the different α_j.

Example 11.2.2

We have two competing models for data that are known to be positive:

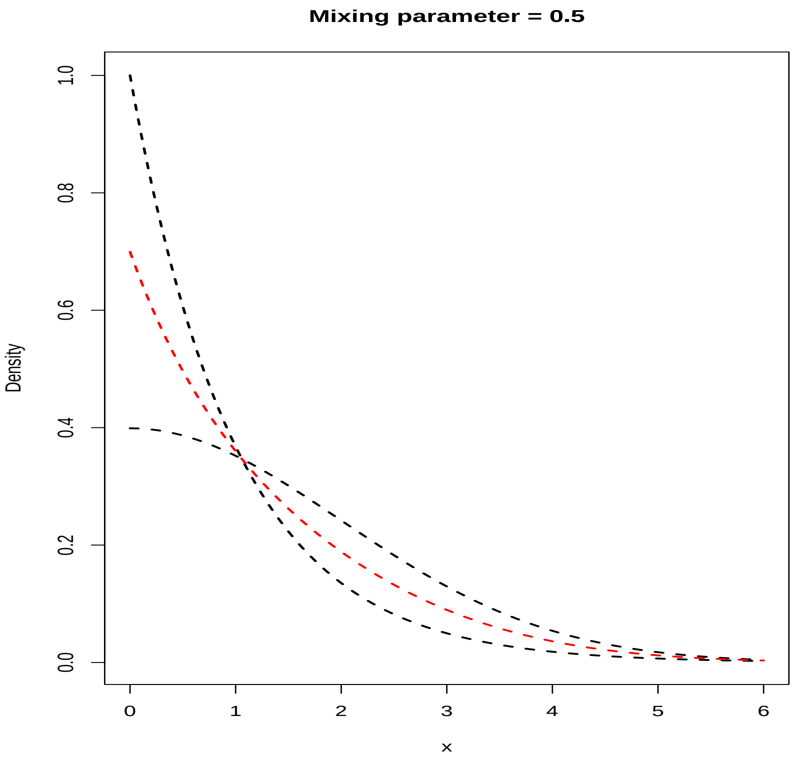
$$\begin{aligned} z_i|\theta &\sim \text{Bernoulli}(\theta), \\ x_i|z_i = 1, \lambda &\sim \text{Exp}(\lambda), \\ x_i|z_i = 0, \sigma &\sim \text{HalfNormal}(\sigma), \quad i = 1, .., n. \end{aligned}$$

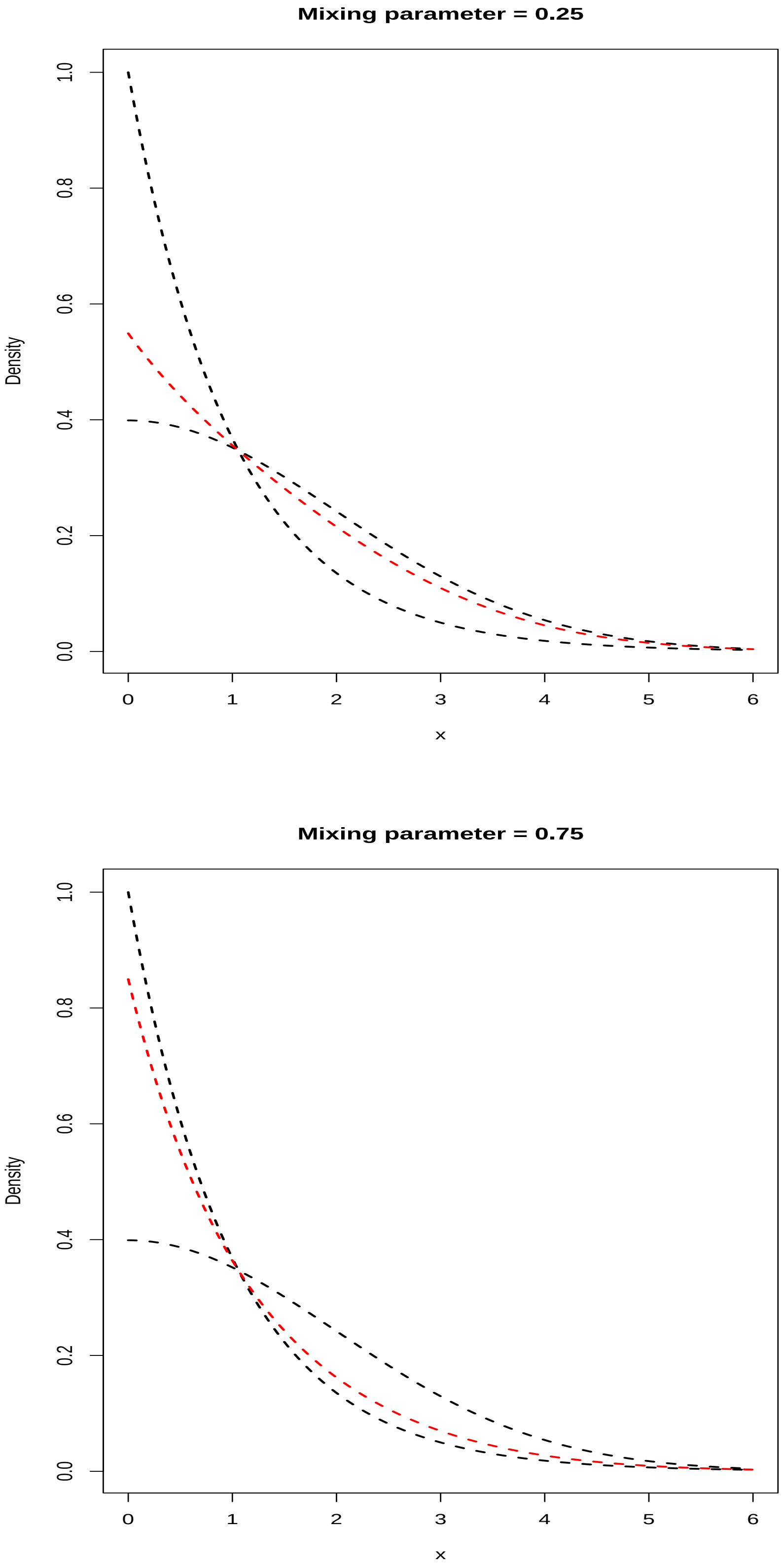
If we let the pdf associated with mixture element j be $\pi_j(x|\text{parameters})$, the likelihood here (dropping the index on x and z) is

$$\begin{aligned} l(\lambda, \sigma, \theta; x) &\propto \pi(x|\lambda, \sigma, \theta) \\ &= \sum_{j=1}^{2} \pi(x|\lambda, \sigma, z)\pi(z|\theta) \\ &= \theta\pi_1(x|\lambda) + (1-\theta)\pi_2(x|\sigma). \end{aligned}$$

We must be careful about the interpretation of θ in this model. If we find that θ is zero or one, then we can be confident that one of the likelihood forms is favoured over the other. However, it is more likely that we get a distribution for θ that supports values inbetween the two extremes.

Imagine we are mixing a Exp(1) distribution with a HalfNormal(2) distribution. Different θ give rise to different shapes, and a model that behaves at odds to the two original distributions.




Mixing parameter = 0.25
Density
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
x
Mixing parameter = 0.75
Density
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
x

11.3 Posterior odds

Throughout the module, we have been assuming some data generating model $\mathcal{M}$, but we have not been explicitly conditioning on it:

$$\pi(\theta \mid \mathbf{x}, \mathcal{M}) \propto \pi(\mathbf{x} \mid \theta, \mathcal{M})\pi(\theta|\mathcal{M}).$$

Recall that, finding the proportionality constant, we get

$$\pi(\theta \mid \mathbf{x}, \mathcal{M}) = \frac{\pi(\mathbf{x} \mid \theta, \mathcal{M})\pi(\theta|\mathcal{M})}{\pi(\mathbf{x}|\mathcal{M})},$$

where $\pi(\mathbf{x}|\mathcal{M})$ is the *evidence* for model $\mathcal{M}$.

If we are in the M-closed situation and we have two competing models for some data $\mathbf{x}$ say: $\mathcal{M}_1$ and $\mathcal{M}_2$. Each model will have its own set of parameters that we will need to deal with: θ_1 and θ_2 say. *A priori*, my odds for $\mathcal{M}_1$ against $\mathcal{M}_2$ are

$$\frac{\pi(\mathcal{M}_1)}{\pi(\mathcal{M}_2)} = \frac{\pi(\mathcal{M}_1)}{1-\pi(\mathcal{M}_1)} = O_f(\mathcal{M}_1).$$

After observing data, my posterior odds for $\mathcal{M}_1$ against $\mathcal{M}_2$ are given by

$$\frac{\pi(\mathcal{M}_1|\mathbf{x})}{\pi(\mathcal{M}_2|\mathbf{x})} = \frac{\pi(\mathcal{M}_1)}{\pi(\mathcal{M}_2)}\frac{\pi(\mathbf{x}|\mathcal{M}_1)}{\pi(\mathbf{x}|\mathcal{M}_2)}$$

or

POSTERIOR ODDS = PRIOR ODDS × BAYES FACTOR.

A question remains over how we calculate $\pi(\mathbf{x}|\mathcal{M}_i)$. This is an instance of the preposterior distribution under $\mathcal{M}_i$.

Example 11.3.1

Consider two models:

$$\begin{aligned}\mathcal{M}_1 &: X|\theta \sim \mathsf{Bin}(10,\theta), \quad \theta \sim \mathsf{Be}(2,2);\\ \mathcal{M}_2 &: X|\theta \sim \mathsf{Bin}(20,\theta), \quad \theta \sim \mathsf{Be}(2,2).\end{aligned}$$

I strongly believe that the first model is correct: $\pi(\mathcal{M}_1) = 0.9$. So my prior odds for $\mathcal{M}_1$ are 9. I then observe $x = 10$.

$$\begin{aligned}\pi(x=10|\mathcal{M}_1) &= \int_0^1 \pi(x=10|\theta,\mathcal{M}_1)\pi(\theta|\mathcal{M}_1)d\theta\\ &= \binom{10}{10}\frac{1}{B(2,2)}\int_0^1 \theta^{11}(1-\theta)d\theta\\ &= \frac{B(12,2)}{B(2,2)} \text{ (The previous integral was of a Be(12,2) density.)}\end{aligned}$$

$$= \frac{11! \times 1!}{13!}\frac{3!}{1! \times 1!} = \frac{1}{26}.$$

Similarly, $\pi(x = 10|\mathcal{M}_2) = 11/161$. So my posterior odds for $\mathcal{M}_1$ are

$$\frac{\pi(\mathcal{M}_1|x)}{\pi(\mathcal{M}_2|x)} = 9 \times \frac{161}{11 \times 26} = 5.07 \text{ to 2 d.p.}$$

or my posterior probability for the first model is

$$\pi(\mathcal{M}_1|x) = \frac{5.07}{1 + 5.07} = 0.84.$$

If I had believed that both models were equally likely *a priori*, my posterior odds for the first model would equal the Bayes factor, 0.56, and my posterior probability for the first model would be 0.36.

11.4 Bayesian model averaging

Bayesian model averaging (BMA) takes us back to the equation at the start of this chapter for the M-closed scenario:

$$\pi(\mathbf{x}^*|\mathbf{x}) = \sum_{i=1}^{k} \pi\left(\mathbf{x}^*|\mathbf{x}, \mathcal{M}_i\right) \pi\left(\mathcal{M}_i|\mathbf{x}\right),$$

where we make the distinction between data we have seen, $\mathbf{x}$, and data we might see in the future $\mathbf{x}^*$.

Example 11.4.1

Returning to the previous example and assuming that both models were thought to be equally likely *a priori*, we have

$$\pi\left(\mathcal{M}_1|\mathbf{x}\right) = 0.36 \text{ and } \pi\left(\mathcal{M}_2|\mathbf{x}\right) = 0.64.$$

Under $\mathcal{M}_1$, the predictive probability mass function is

$$\pi(x^*|x) = \binom{10}{x^*}\frac{\mathrm{B}(x^* + 12, 12 - x^*)}{\mathrm{B}(12, 2)}.$$

Similarly, under $\mathcal{M}_2$, the predictive probability mass function is

$$\pi(x^*|x) = \binom{20}{x^*}\frac{\mathrm{B}(x^*+12, 32-x^*)}{\mathrm{B}(12,12)}.$$

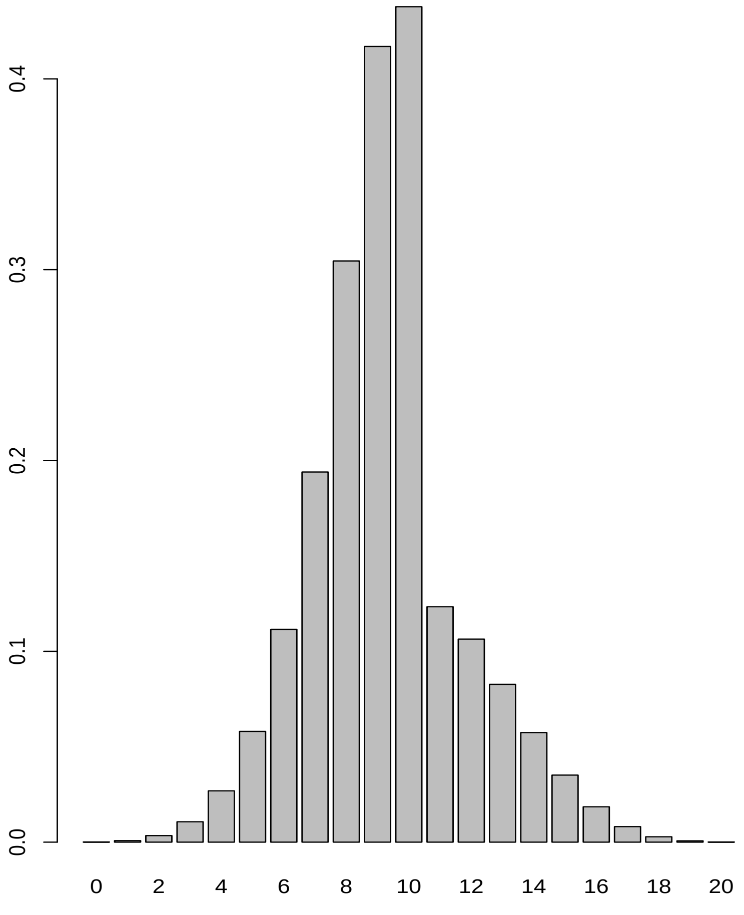
These lead us to combined predictive probability mass function using BMA of

$$\pi(x^*|x) = 0.36\binom{10}{x^*}\frac{\mathrm{B}(x^*+12, 12-x^*)}{\mathrm{B}(12,2)} + 0.64\binom{20}{x^*}\frac{\mathrm{B}(x^*+12, 32-x^*)}{\mathrm{B}(12,12)}$$

for $x^* \leq 10$ and

$$\pi(x^*|x) = 0.64\binom{20}{x^*}\frac{\mathrm{B}(x^*+12, 32-x^*)}{\mathrm{B}(12,12)}$$

for $10 < x^* \leq 20$.



This entire method depends on our ability to derive the model evidence and, subsequently, the posterior probabilities for the models.

Example 11.4.2

$X|\theta \sim \mathsf{N}\left(\theta, \frac{1}{6}\right)$ with $\theta \sim \mathsf{Exp}(3)$, and we observe $x_1 = 1$ and $x_2 = 6$:

$$\pi(\theta|\underline{x}) \quad \propto \quad e^{-3\theta}e^{-3(1-\theta)^2}e^{-3(6-\theta)^2}.$$

We can find the posterior mode $\frac{13}{4}$ (by differentiation), but, to get the evidence, we need to integrate. We have:

$$\begin{aligned}\pi(\underline{x} = \{1, 6\}) &= \int_0^\infty \pi(x = 1|\theta)\pi(x = 6|\theta)\pi(\theta)\, d\theta \\ &\propto \int_0^\infty e^{-3\theta}e^{-3(1-\theta)^2}e^{-3(6-\theta)^2}\, d\theta.\end{aligned}$$

We can approximate the posterior probabilities for the individual models using approximate Bayesian computation (ABC and see Chapter 6). Very roughly speaking, we do the following very many times until we have an adequately-sized sample from our posterior:

1. randomly choose a model using the prior model probabilities,

2. randomly choose parameter values based upon the chosen model,

3. generate simulated data using the chosen model's data generating process with the chosen model parameters,

4. check to see if the simulated data are "close" to the observed data,

5. if they are close, add one to the count of model acceptances and store the parameters; if not, discard all.

After doing this, we can estimate the posterior probabilities for each model by dividing the number of times we have accepted a set of parameters for that model dived by the total number of acceptances.

Example 11.4.3

Once again, consider the two models:

$$\begin{aligned}\mathcal{M}_1 &: \; X|\theta \sim \mathsf{Bin}(10, \theta), \quad \theta \sim \mathsf{Be}(2, 2); \\ \mathcal{M}_2 &: \; X|\theta \sim \mathsf{Bin}(20, \theta), \quad \theta \sim \mathsf{Be}(2, 2).\end{aligned}$$

I have that $\pi(\mathcal{M}_1) = 0.5$. We observe $x = 10$.

Here are the first few iterations of the ABC algorithm with **bold** denoting acceptance.

Iteration number	Model selected	θ selected	Simulated data
1	$\mathcal{M}_1$	0.52	8
2	$\mathcal{M}_1$	0.78	5
3	$\boldsymbol{\mathcal{M}_2}$	**0.45**	**10**
4	$\mathcal{M}_1$	0.32	3
5	$\mathcal{M}_2$	0.14	8

Repeating this process for 10,000 iterations, we get simulated data of $x^* = 10$ on 556 occasions and 210 of those are when we have used $\mathcal{M}_1$. Therefore, we estimate

$$\pi\left(\mathcal{M}_1|\mathbf{x}\right) = 210/556 = 0.38 \text{ to 2 d.p..}$$

This becomes less trivial when the data are continuous and more abundant. We may need to allow the simulated data to be in a neighbourhood of the observed data rather than being exactly correct and also need to use summary statistics rather than trying to match each data point.

11.5 Predictive performance and cross validation

At the beginning of this chapter, the idea of using predictive performance to rate and select models was mentioned. In an ideal world, we would build our models and then use them to make separate predictions of some unseen data.

Example 11.5.1

Consider two models:

$$\begin{aligned}
\mathcal{M}_1 \quad &: \quad X|\mu \sim \mathsf{N}(\mu, 5^2), \quad \mu \sim \mathsf{N}(5,1);\\
\mathcal{M}_2 \quad &: \quad X|\mu \sim \mathsf{N}(\mu, 10^2), \quad \mu \sim \mathsf{N}(3,1).
\end{aligned}$$

We observe 50 x and get

$$\sum_{i=1}^{50} x_i = 191.34.$$

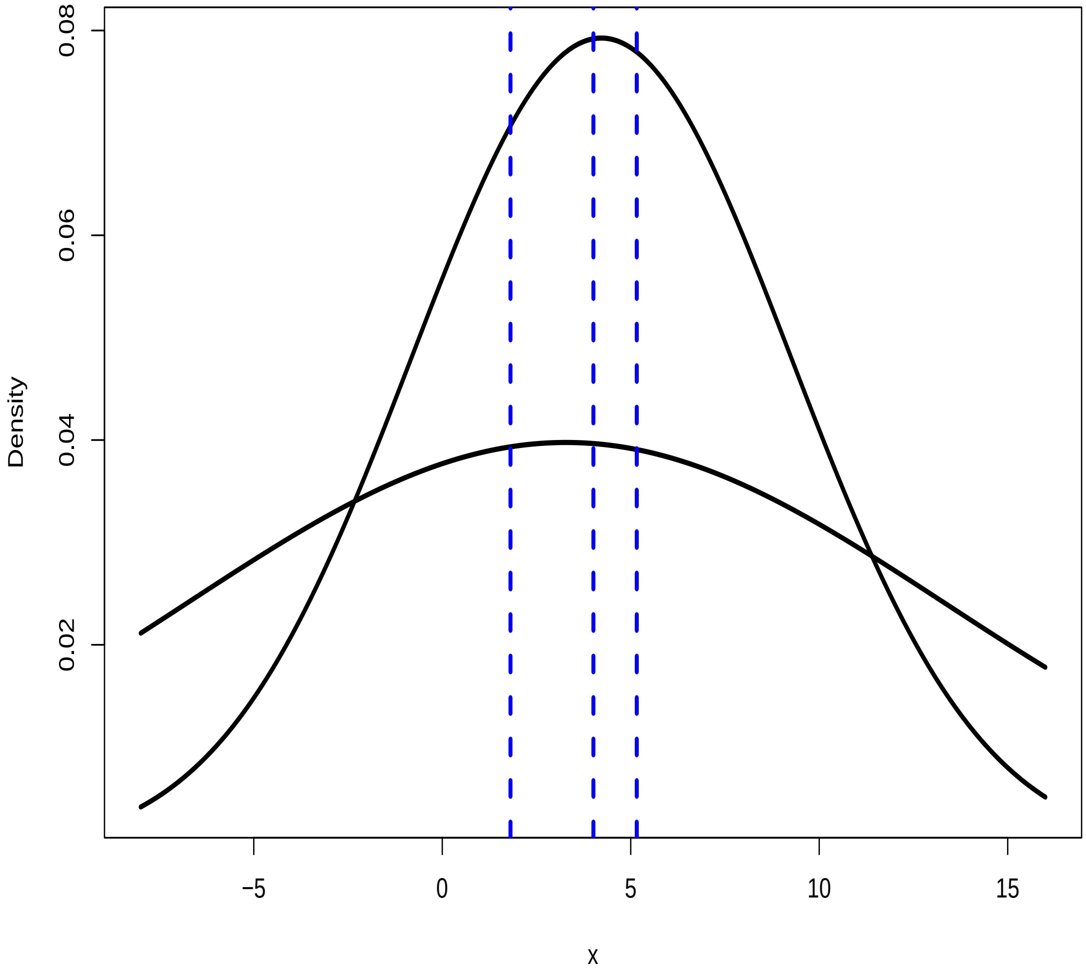
By conjugacy, we get the following posteriors for each model:

$$\mathcal{M}_1 : \quad \mu|\underline{x} \sim \mathsf{N}(4.22, 1/3); \qquad \mathcal{M}_2 : \quad \mu|\underline{x} \sim \mathsf{N}(3.28, 2/3).$$

We can also derive predictive distributions for both models:

$$\begin{aligned}\mathcal{M}_1 &: \quad X^*|\underline{x} \sim \mathsf{N}(4.22, 1/3 + 5^2);\\ \mathcal{M}_2 &: \quad X^*|\underline{x} \sim \mathsf{N}(3.28, 2/3 + 10^2).\end{aligned}$$

If we now observe 1.81, 4.01 and 5.16, which model is best?



What if we cannot get new data and we want to make a determination based on the data that we have already got? It is wrong to try to predict the data points that have been used to fit the model. *Cross validation* is popular in both statistics and machine learning for evaluating model performance (and for model fitting). Here, we will consider leave-one-out cross validation where the model is fitted on all the data apart from one observation and the model performance is a measure of how well the model using the reduced dataset replicates the single removed data point.

Predictive distributions based upon cross validation schemes have been proposed to approximate Bayes factors. In fact, we have the *pseudo-Bayes factor*:

$$\prod_{i=1}^{n} \frac{\pi(x_i|\underline{x}_{-i}, \mathcal{M}_1)}{\pi(x_i|\underline{x}_{-i}, \mathcal{M}_2)}.$$

In the context of model weighting, we may choose to use the log point-wise predictive density to evaluate model performance (logarithms are used to improve numeric stability and point-wise refers to the leave-one-out scheme):

$$\log\left[\pi(x_i|\underline{x}_{-i}, \mathcal{M}_p)\right] = \log\left[\int \pi(\boldsymbol{\theta}|\underline{x}_{-i}, \mathcal{M}_p)\pi(x_i|\boldsymbol{\theta}, \mathcal{M}_p)d\boldsymbol{\theta}\right],$$

where $\boldsymbol{\theta}$ represents the unknown parameters and $\underline{x}_{-i}$ represents the full data set with the ith point removed. Clearly, we would like our performance metric to accommodate all of the individual data point predictions so we use the so-called expected log point-wise predictive density for model k, denoted here by elpd:

$$\widehat{\mathsf{elpd}}_p = \sum_{i=1}^{n} \log\left[\pi(x_i|\underline{x}_{-i}, \mathcal{M}_p)\right].$$

We then convert this to a model weight, called a pseudo-Bayesian-model-averaging weight:

$$\widehat{w}_p = \frac{\exp\left(\widehat{\mathsf{elpd}}_p\right)}{\sum_{j=1}^{k} \exp\left(\widehat{\mathsf{elpd}}_j\right)}.$$

Therefore, the higher a model's elpd, the more influence it will have in the pseudo-Bayesian-model-averaging prediction. Note that these weights operate under the assumption that all models in $\underline{\mathcal{M}}$ are equally likely.

Example 11.5.2

Let's return to an earlier example. Again, consider two models:

$$\begin{aligned}\mathcal{M}_1 &: X|\theta \sim \mathsf{Bin}(10,\theta), \quad \theta \sim \mathsf{Be}(2,2);\\ \mathcal{M}_2 &: X|\theta \sim \mathsf{Bin}(20,\theta), \quad \theta \sim \mathsf{Be}(2,2).\end{aligned}$$

This time we observe $x_1 = 10$, $x_2 = 8$ and $x_3 = 10$. If we condition on two of those observations at a time, we have

$$\begin{aligned}\pi(x_i|\underline{x}_{-i}, \mathcal{M}_1) &= \binom{10}{x_i}\frac{\mathrm{B}(2+\sum x_j, 32-\sum x_j)}{\mathrm{B}(2+\sum_{-i} x_j, 22-\sum_{-i} x_j)},\\ \pi(x_i|\underline{x}_{-i}, \mathcal{M}_2) &= \binom{20}{x_i}\frac{\mathrm{B}(2+\sum x_j, 62-\sum x_j)}{\mathrm{B}(2+\sum_{-i} x_j, 42-\sum_{-i} x_j)}.\end{aligned}$$

We can then compute the elpd for each model:

Observation left out	$\log \pi(x_i \vert \underline{x}_{-i}, \mathcal{M}_1)$	$\log \pi(x_i \vert \underline{x}_{-i}, \mathcal{M}_2)$
x_1	-1.53	-1.98
x_2	-1.97	-2.18
x_3	-1.53	-1.98

We can then get the unnormalised pseudo-Bayesian-model-averaging weights of

$$\begin{aligned} \exp\left(\widehat{\mathsf{elpd}}_1\right) &= 0.0065, \\ \exp\left(\widehat{\mathsf{elpd}}_2\right) &= 0.0021, \end{aligned}$$

which can be combined to give normalised pseudo-Bayesian-model-averaging weights of 0.75 for $\mathcal{M}_1$ and 0.25 for $\mathcal{M}_2$.

This contradicts what was calculated in previous examples where the data seemed to favour $\mathcal{M}_2$. The reason is that, in full posterior calculations, you are rewarded for getting the prior in the right area. With the predictive-based weights, you are rewarded for stacking up posterior predictive probability near to the data (in other words, the likelihood takes over).

Just like model evidence, the elpd is difficult to calculate. However, we can find the psuedo-Bayesian-model-averaging weights by generating some extra quantities in Stan and utilising functions in the loo package. We add a variable called log_lik to the Stan code for each model under consideration:

```
generated quantities {
  // we need a log-likelihood calculation for each observation
  vector[N_obs] log_lik;

  // we have a normal likelihood in this example
  for (n in 1:N_obs)
    log_lik[n] = normal_lpdf(x[n] | mu, sigma);
}
```

Here, we are getting evaluations of the posterior predictive distribution of the observed values. These are $\log\left[\pi(x_i|\underline{x}, \mathcal{M}_p)\right]$ rather then the $\log\left[\pi(x_i|\underline{x}_{-i}, \mathcal{M}_p)\right]$ that are needed in the elpd calculation. The loo package works directly with the sampled log-likelihood values to approximate the expected log pointwise predictive density (by applying a correction based on Pareto distributions that works better under certain circumstances):

```
library(loo)

# LOO for model 1
loo1 <- loo(fit)
# LOO for model 2
loo2 <- loo(fit_2)

# The weights
loo_model_weights(list(loo1, loo2),
                  method = "pseudobma",
                  BB = FALSE)
```

Example 11.5.3

Consider two models:

$$\begin{aligned}\mathcal{M}_1 &: X|\theta \sim \mathsf{N}\left(\theta, 0.16\right), \quad \theta \sim \mathsf{Exp}(3); \\ \mathcal{M}_2 &: X|\theta \sim \mathsf{N}\left(\theta, 0.17\right), \quad \theta \sim \mathsf{Exp}(3).\end{aligned}$$

We observe $\underline{x} = \{0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3\}$. It is clear already that both models are terrible. But, if we persist, we have the following Stan code:

```
data {
  int<lower=0> N;
  real x[N];
  real<lower=0> sigma2;
}
parameters {
  real<lower=0> theta;
}
model {
  // Prior
  theta ~ exponential(3);

  // Likelihood
  for (i in 1:N)
    x[i] ~ normal(theta, sqrt(sigma2));
}
generated quantities {
  vector[N] log_lik;
  for (i in 1:N)
    log_lik[i] = normal_lpdf(x[i] | theta, sqrt(sigma2));
}
```

The R code to then generate weights for each of the two models is then:

```
library(rstan)
library(loo)

# compile the model
model <- stan_model(file = "Model selection/expnorm.stan")

# generate a posterior sample for M1
fit1 <- sampling(model,
                 data = list(N = 20,
                             x = rep(0:3,5),
                             sigma2 = 0.16),
                 iter = 10000)

# generate a posterior sample for M2
fit2 <- sampling(model,
                 data = list(N = 20,
                             x = rep(0:3,5),
                             sigma2 = 0.17),
                 iter = 10000)

# calculate pseudo-BMA weights
loo1 <- loo(fit1)
loo2 <- loo(fit2)
loo_model_weights(list(loo1, loo2),
                  method = "pseudobma",
                  BB = FALSE)
```

This results in the following output:

```
Method: pseudo-BMA
------
       weight
model1 0.016
model2 0.984
```

which is not at all surprising and shows a key feature of modelling in the M-complete or M-open regimes: the weights will converge to one for the model that is closest to the true real-world model even if that model is wrong.