

Bayesian Inference and Computation

Dr Rowland Seymour

Semester 2, 2023

Contents

Practicalities

0.1 Module Aims

The module aims to give you an overview of the Bayesian paradigm. By the end of the course, you should

1. Be able to conceptualise the Bayesian approach for statistics
2. Be able to derive posterior and posterior predictive distributions for univariate and multivariate models
3. Identify suitable prior distributions and understand how the choice of prior distribution may affect the final result
4. Understand the principles of Markov chain Monte Carlo and be able to construct MCMC algorithms

0.2 Module Structure

The module is split between theory and programming. Each week (excluding week 6) will have two lectures and two computer labs.

0.3 Assessment

The module is 55% coursework and 45% exam. The exam will last 1h 30m and take place during the summer exam period. More details about the coursework will be announced during the semester.

0.4 Getting Help

There are lots of ways of getting help throughout the module. You can visit my office hour (Watson 317) on or email me at r.g.seymour@bham.ac.uk. Each week, there will also be a problem class.

0.5 Recommended Books and Videos

No books are required for this course and the whole material is contained in these notes. However, you may find it useful to use other resources in your studies. I recommend the following:

1. A First Course in Bayesian Statistical Methods - Peter D. Hoff. This is a short book that covers the basics of Bayesian inference and computation. To the point and well written, it's a useful place to look topics up.
2. Bayesian Data Analysis - Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. This is a thorough book explaining everything you'd need to know to carry out Bayesian data analysis. It's a fairly look and in-depth book, but the authors are authoritative and give good advice throughout. Example code on the website is in R, Python and Stan.
3. Statistical Rethinking - Richard McElrath. This book provides a friendly intuitive understanding of Bayesian inference and computation. Aimed at social and natural scientists, it has less theory than the other two books but is perhaps more approachable. A set of video lectures for this book can be found on YouTube.

0.6 Common Distributions

For many Bayesian inference problems, it is useful to be able to identify probability density functions (for continuous random variables) and probability mass functions (for discrete random variables) up to proportionality. Some common density/mass functions are given below.

Normal distribution

$$\pi(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\} \quad x \in \mathbb{R},$$

where $\mu \in \mathbb{R}$ and $\sigma > 0$.

Beta distribution

$$\pi(x \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad x \in [0, 1],$$

where $\alpha, \beta > 0$ and $B(\alpha, \beta)$ is the beta function.

Gamma distribution

$$\pi(x \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \quad x > 0,$$

where $\alpha, \beta > 0$ and $\Gamma(\alpha)$ is the gamma function.

Exponential distribution

$$f(x \mid \lambda) = \lambda e^{-\lambda x} \quad x > 0,$$

where $\lambda > 0$.

Poisson distribution

$$\pi(x = k \mid \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad k \in \{1, 2, \dots\},$$

where $\lambda > 0$.

Binomial distribution

$$\pi(x = k \mid N, p) = \binom{N}{k} p^k (1-p)^{N-k} \quad k \in \{1, \dots, N\}$$

where $p \in [0, 1]$.

Chapter 1

Fundamentals of Bayesian Inference

Bayesian inference is built on a different way of thinking about parameters of probability distributions using methods you have learnt so far. In the past 30 years or so, Bayesian inference has become much more popular. This is partly due to increased computational power becoming available. In this first chapter, we are going to set out to answer:

1. What are the fundamental principles of Bayesian inference?
2. What makes Bayesian inference different from other methods?

1.1 Statistical Inference

The purpose of statistical inference is to “draw conclusions, from numerical data, about quantities that are not observed” (Bayesian Data Analysis, chapter 1). Generally speaking, there are two kinds of inference: inference for quantities that are not possible to observe and inference for quantities that are unobserved or hasn’t happened yet. We can apply our inference methods to a huge range of applications, from the effectiveness of drugs in a clinical trial, to estimating the number of victims of violence against women and girls.

In this module, we are going to look at a different way of carrying out statistical inference, one that doesn’t depend on long run events. Instead, we’re going to introduce the definition of probability that allows us to interpret the subjective chance that an event occurs.

1.2 Frequentist Theory

Frequentist probability is built upon the theory on long run events. Probabilities must be interpretable as frequencies over multiple repetitions of the experiment that is being analysed, and are calculated from the sampling distributions of measured quantities.

Definition 1.1. The long run relative frequency of an event is the **probability** of that event.

Example 1.1. If a frequentist wanted to assign a probability to rolling a 6 on a particular dice, then they would roll the dice a large number of times and compute the relative frequency.

Definition 1.2. The **sampling distribution** of a statistic is the distribution based on a long run of samples of a fixed size from the population.

These two definitions form the basis of frequentist statistics and give rise to higher-level objects. However, due to the long run condition, these objects are often misunderstood.

Definition 1.3. Let X be a random sample from a probability distribution θ . A **95% confidence interval** for θ is an interval $(u(X), v(X))$ such that

$$\mathbb{P}(u(X) < \theta < v(X)) = 0.95$$

This means that if you had an infinite number of samples for X and the corresponding infinite number of confidence intervals, then 95% of them would contain the true value of θ . It does *not* mean that there is a 0.95 probability a particular interval contains the true value of θ .

Example 1.2. Suppose that a ornithologist believes that the sex of an offspring of a certain species of bird is equally likely to be male or female and is independent of the sex of any siblings. They selects a random sample of 100 broods with 4 offspring and counts the number of male offspring in each. The results are summarised in the following table.

```
offspring.df <- data.frame("males" = c(0, 1, 2, 3, 4),
                           "observed" = c(5, 20, 50, 20, 5))

knitr::kable(
  offspring.df, booktabs = TRUE,
  caption = 'The number of males in each of the broods.'
)
```

If X denotes the number of male offspring then the scientist's null hypothesis H_0 , is that $X \sim \text{Bin}(4, 1/2)$. This can be tested by calculating the so called χ^2 statistic - which measures how well the observations conform to the supposed distribution. This is computed by

Table 1.1: The number of males in each of the broods.

males	observed
0	5
1	20
2	50
3	20
4	5

```

expected <- 100*dbinom(offspring.df$males, 4, 0.5)
observed <- offspring.df$observed

## Chi sq = sum of (expected - observed)^2/expected
chi.sq <- sum((expected - observed)^2/expected)

```

Under H_0 , over repeated sampling the distribution of the χ^2 statistic follows a χ^2 distribution with 4 degrees of freedom. Therefore, the probability of seeing something as extreme as this (the p-value) is

```
1 - pchisq(chi.sq, 4)
```

```
## [1] 0.1545873
```

In other words, there is no significant evidence in the data to reject H_0 at the 10% level.

Consider now a second scientist who is more open-minded than their colleague. They hypothesise only that $X \sim \text{Bin}(4, p)$ where p is unknown. To test their H_0 they first calculate the maximum likelihood estimator of p , $\hat{p} = 0.5$, compute the values of $E[X]$ using this estimate and calculate χ^2 to be 6.67 like their colleague.

However, because they are estimating p as part of the process of computing χ^2 over repeated sampling the distribution of their χ^2 is (approximately) a χ^2 distribution on 3 degrees of freedom. Their p-value is therefore

```
1 - pchisq(chi.sq, 3)
```

```
## [1] 0.08331631
```

We now have the somewhat confusing scenario whereby the second scientist finds that the evidence against their null hypothesis is stronger than that found by the first scientist against their null hypothesis, despite the second hypothesis being weaker (i.e. a logical consequence of the first). Thus, whatever a p-value represents, it must never be interpreted as the probability that the null hypothesis is true conditional on the observed data. In this example it simply tells us

how extreme the observed values of χ^2 are when compared to their distribution over many repetitions of the experiment.

1.3 Bayesian Probability

The Bayesian paradigm is built around a different definition of probability. We are simply building our view of statistics upon a different set of axioms.

Definition 1.4. The subjective belief of an event is the **probability** of that event.

This definition means we can assign probabilities to events that frequentists do not recognise as valid.

Example 1.3. Consider the following:

1. The probability that I vote for the labour party at the next election
2. A photo taken from the James Watt telescope contains a new planet.
3. The real identify of Banksy is Robin Gunningham.

These are not events that can be repeated in the long run.

1.4 Conditional Probability and Exchangability

Before we derive Bayes' theorem, we recap some important definitions in probability.

Definition 1.5. Given two events A and B , the **conditional probability** that event A occurs given the event B has already occurred is

$$\pi(A | B) = \frac{\pi(A \cap B)}{\pi(B)},$$

when $\pi(B) > 0$.

Definition 1.6. Two events A and B are **independent** given event C if and only if

$$\pi(A \cap B | C) = \pi(A | C)\pi(B | C).$$

Definition 1.7. Let $\pi(y_1, \dots, y_N)$ be the joint density of Y_1, \dots, Y_N . If $\pi(y_1, \dots, y_N) = \pi(y_{\pi_1}, \dots, y_{\pi_N})$ for a permutations π of $\{1, \dots, N\}$, then Y_1, \dots, Y_N are **exchangeable**.

Exchangability means that the labels of the random variables don't contain any information about the outcomes. This is an important idea in many areas of probability and statistics, and we often model exchangeable events as iid.

Example 1.4. If $Y_i \sim \text{Bin}(n, p)$ are independent and identically distributed for $i = 1, 2, 3$, then $\pi(Y_1, Y_2, Y_3) = \pi(Y_3, Y_1, Y_2)$.

Example 1.5. Let (X, Y) follow a bivariate normal distribution with mean $\mathbf{0}$, variances $\sigma_x = \sigma_y = 1$ and a correlation parameter $\rho \in [-1, 1]$. (X, Y) are exchangeable, but only independent if $\rho = 0$.

Proposition 1.1. *If $\theta \sim \pi(\theta)$ and (Y_1, \dots, Y_N) from a sample space \mathcal{Y} are conditionally iid given some parameter θ , then marginally Y_1, \dots, Y_N are exchangeable.*

Proof. Suppose (Y_1, \dots, Y_N) are conditionally iid given some parameter θ . Then for any permutation π of $\{1, \dots, N\}$ and observations $\{y_1, \dots, y_N\}$

$$\begin{aligned}
 \pi(y_1, \dots, y_N) &= \int \pi(y_1, \dots, y_N \mid \theta) \pi(\theta) d\theta && \text{(definition of marginal distribution)} \\
 &= \int \left\{ \prod_{i=1}^N \pi(y_i \mid \theta) \right\} \pi(\theta) d\theta && \text{(definition of conditionally iid)} \\
 &= \int \left\{ \prod_{i=1}^N \pi(y_{\pi_i} \mid \theta) \right\} \pi(\theta) d\theta && \text{(product is commutative)} \\
 &= \pi(y_{\pi_1}, \dots, y_{\pi_N}) && \text{(definition of marginal distribution)}
 \end{aligned} \tag{1.1}$$

□

This tells us that if we have some conditionally iid random variables and a subjective prior belief about some parameter θ , then we have exchangeability. This is nice to have, but the implication in the other direction is much more interesting and powerful.

Theorem 1.1 (de Finetti). *If a sequence of random variables (Y_1, \dots, Y_N) from a sample space \mathcal{Y} is exchangeable, then its joint distribution can be written as*

$$\pi(y_1, \dots, y_N) = \int \left\{ \prod_{i=1}^N \pi(y_i \mid \theta) \right\} \pi(\theta) d\theta$$

for some parameter θ , some distribution on θ , and some sampling model $\pi(y_i \mid \theta)$.

This is a kind of existence theorem for Bayesian inference. It says that if we have exchangeable random variables, then a parameter θ must exist and a subjective probability distribution $\pi(\theta)$ must also exist. The argument against Bayesian inference is that it doesn't guarantee a *good* subjective probability distribution $\pi(\theta)$ exists.

1.5 Bayes' Theorem

Now we have an understanding of conditional probability and exchangeability, we can put these two together to understand Bayes' Theorem. Bayes' theorem is concerned with the distribution of the parameter θ given some observed data

y . It tries to answer the question: what does the data tell us about the model parameters?

Theorem 1.2 (Bayes). *The distribution of the model parameter θ given the data y is*

$$\pi(\theta | y) = \frac{\pi(y | \theta)\pi(\theta)}{\pi(y)}$$

Proof.

$$\pi(\theta | y) = \frac{\pi(\theta, y)}{\pi(y)} \tag{1.2}$$

$$\implies \pi(\theta, y) = \pi(\theta | y)\pi(y) \tag{1.3}$$

Analogously, using $\pi(y | \theta)$ we can derive

$$\pi(\theta, y) = \pi(y | \theta)\pi(\theta)$$

Putting these two terms equal to each other and dividing by $\pi(y)$ gives

$$\pi(\theta | y) = \frac{\pi(y | \theta)\pi(\theta)}{\pi(y)}$$

□

There are four terms in Bayes' theorem:

1. The **posterior distribution** $\pi(\theta | y)$. This tells us our belief about the model parameter θ given the data we have observed y .
2. The **likelihood function** $\pi(y | \theta)$. The likelihood function is common to both frequentist and Bayesian methods. By the likelihood principle, the likelihood function contains all the information the data can tell us about the model parameter θ .
3. The **prior distribution** $\pi(\theta)$. This is the distribution that describes our prior beliefs about the value of θ . The form of θ should be decided before we see the data. It may be a vague distribution (e.g. $\theta \sim N(0, 10^2)$) or a specific distribution based on prior information from experts (e.g. $\theta \sim N(5.5, 1.3^2)$).
4. The **evidence of the data** $\pi(y)$. This is sometimes called the average probability of the data or the marginal likelihood. In practice, we do not need to derive this term as it can be back computed to ensure the posterior distribution sums/integrates to one.

A consequence of point four is that posterior distributions are usually derived proportionally, and (up to proportionality) Bayes' theorem

$$\pi(\theta | y) \propto \pi(y | \theta)\pi(\theta).$$

Some history of Thomas Bayes. Thomas Bayes was an English theologian born in 1702. His “Essay towards solving a problem in the doctrine of chances” was published posthumously. It introduces theorems on conditional probability and the idea of prior probability. He discusses an experiment where the data can be modelled using the Binomial distribution and he guesses (places a prior distribution) on the probability of success.

Richard Price sent Bayes' work to the Royal Society two years after Bayes had died. In his commentary on Bayes' work, he suggested that the Bayesian way of thinking proves the existence of God, stating: The purpose I mean is, to show what reason we have for believing that there are in the constitution of things fixed laws according to which things happen, and that, therefore, the frame of the world must be the effect of the wisdom and power of an intelligent cause; and thus to confirm the argument taken from final causes for the existence of the Deity.

It's not clear how Bayesian Thomas Bayes actually was, as his work was mainly about specific forms of probability theory and not his interpretation of it. The Bayesian way of thinking was really popularised by Laplace, who wrote about deductive probability in the early 19th century.

Example 1.6. We finish this chapter with a very simple example. The advantage of the example being so simple is that we can obtain plots in R that show what's going on.

Suppose we have a model $y \sim N(\theta, 1)$ and we want to estimate θ . To do this we need to derive the posterior distribution. By Bayes' theorem,

$$\pi(\theta | y) \propto \pi(y | \theta)\pi(\theta).$$

We know the form of $\pi(y | \theta) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(y-\theta)^2}$, but how should we describe our prior beliefs about θ ? Here are three options:

1. We can be very vague about θ – we genuinely don't know about its value. We assign a uniform prior distribution to θ that takes values between -1,000 and +1,000, i.e. $\theta \sim u[-1000, 1000]$. Up to proportionality $\pi(\theta) \propto 1$ for $\theta \in [-1000, 1000]$.
2. After thinking hard about the problem, or talking to an expert, we decide that the only thing we know about θ is that it can't be negative. We adjust our prior distribution from 1. to be $\theta \sim u[0, 1000]$. Up to proportionality $\pi(\theta) \propto 1$ for $\theta \in [0, 1000]$.
3. We decide to talk to a series of experts about θ asking for their views on likely values of θ . Averaging the experts' opinions gives $\theta \sim N(3, 0.7^2)$. This is a method known as prior elicitation.

We now go and observe some data. After a lot of time and effort, we collect one data point – $y = 0$.

Now we have all the ingredients to construct the posterior distribution. We multiply the likelihood function evaluated at $y = 0$ by each of the three prior distributions. This gives us the posterior distributions (up to proportionality).

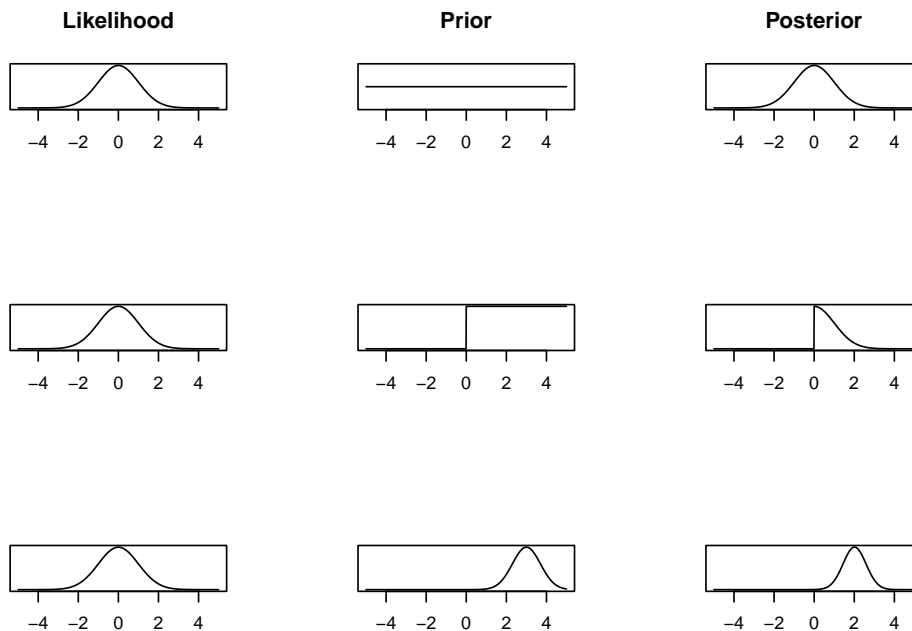
```
#The likelihood function is the normal PDF
#To illustrate this, we evaluate this from [-5, 5].
x <- seq(-5, 5, 0.01)
likelihood <- dnorm(x, mean = 0, sd = 1)

#The first prior distribution we try is a
#uniform [-1000, 1000] distribution. This is a
#vague prior distribution.
uniform.prior <- rep(1, length(x))
posterior1 <- likelihood*uniform.prior

#The second prior distribution we try is a uniform
#[0, 1000] distribution, i.e. theta is non-negative.
step.prior <- ifelse(x >= 0, 1, 0)
posterior2 <- likelihood*step.prior

#The third prior distribution we try is a
#specific normal prior distribution. It
#has mean 3 and variance 0.7.
normal.prior <- dnorm(x, mean = 3, sd = 0.7)
posterior3 <- likelihood*normal.prior

#Now we plot the likelihoods, prior and posterior distributions.
#Each row corresponds to a different prior distribution. Each
#column corresponds to a part in Bayes' theorem.
par(mfrow = c(3, 3))
plot(x, likelihood, type = 'l', xlab = "", ylab = "", yaxt = "n", main = "Likelihood")
plot(x, uniform.prior, type = 'l', yaxt = "n", xlab = "", ylab = "", main = "Prior")
plot(x, posterior1, type = 'l', yaxt = "n", xlab = "", ylab = "", main = "Posterior")
plot(x, likelihood, type = 'l', xlab = "", ylab = "", yaxt = "n")
plot(x, step.prior, type = 'l', yaxt = "n", xlab = "", ylab = "")
plot(x, posterior2, type = 'l', yaxt = "n", xlab = "", ylab = "")
plot(x, likelihood, type = 'l', xlab = "", ylab = "", yaxt = "n")
plot(x, normal.prior, type = 'l', yaxt = "n", xlab = "", ylab = "")
plot(x, posterior3, type = 'l', yaxt = "n", xlab = "", ylab = "")
```

1. The posterior distribution is proportional to the likelihood function. The prior distribution closely matches frequentist inference. Both the MLE and posterior mean are 0.
2. We get a lopsided posterior distribution, that is proportional to the likelihood function for positive values of θ , but is 0 for negative values of θ .
3. We get some sort of average of the likelihood function and the prior distribution. Had we collected more data, the posterior distribution would have been weighted toward the information from the likelihood function more.

1.6 Exercises

Exercise 1.1. Consider a standard pack of 52 playing cards. You pick a card at random, what is the probability you pick:

1. A Queen, given you have picked a picture card (King, Queen, Jack)?
2. The five of clubs, given you have picked a black card?
3. A black card, given you have not picked the five of clubs?

Solution. A pack of playing cards is equally divided into four suits: Hearts (red), Diamonds (red), Clubs (black), and Spades (black). Each suit has 13 cards numbered 2 - 10, Jack, Queen, King (all three picture cards), and Ace.

1.

$$\begin{aligned}
 P(\text{Queen} \mid \text{Picture}) &= \frac{P(\text{Queen and Picture})}{P(\text{Picture})} \\
 &= \frac{4/52}{12/52} \\
 &= \frac{1}{3}.
 \end{aligned}$$

2.

$$\begin{aligned}
 P(5 \text{ Clubs} \mid \text{Black}) &= \frac{P(5 \text{ Clubs and Black})}{P(\text{Black})} \\
 &= \frac{1/52}{1/2} \\
 &= \frac{1}{26}.
 \end{aligned}$$

3.

$$\begin{aligned}
 P(\text{Black} \mid \text{Not 5 Clubs}) &= \frac{P(\text{Black} \mid \text{Not 5 Clubs})}{P(\text{Not 5 Clubs})} \\
 &= \frac{25/52}{51/52} \\
 &= \frac{25}{51}.
 \end{aligned}$$

Exercise 1.2. Decide if each of the following events can be assigned probabilities by frequentists:

1. The Bermuda triangle exists.
2. Getting a 6 when rolling a dice.
3. Someone will test positive for Covid-19 after contracting the disease.
4. The sun will rise tomorrow.

Solution. 1. No, this can't be assigned a probability.

2. Yes, you can repeatedly roll of dice.

3. Yes, this can be assigned a probability. You can repeatedly test someone for the disease, hence there is a long-run frequency of the test returning a positive result.

4. It depends what you mean by tomorrow. Suppose today is 1st January 2023, if tomorrow means 2nd January 2023, then no. 2nd January 2023 will on occur once and there is no long-run frequency. If, however, you define tomorrow by the day after today, then yes. There have been many (The Earth has been going round the Sun for ~4.5 billion years, so approximately 4.5×365 tomorrows), so it can be assigned a probability.

Exercise 1.3. An urn contains three coins. Two of the coins are fair, but one of the coins has heads on both sides.

1. You pick a coin out of the urn without looking and flip it. What's the probability you get heads?
2. You pick a coin out of the urn without looking and flip it and get heads. What's the probability it's the two-headed coin?

Solution. Label the coins 1, 2, 3, where $P(\text{Heads} \mid \text{Coin 1}) = P(\text{Heads} \mid \text{Coin 2}) = \frac{1}{2}$ and $P(\text{Heads} \mid \text{Coin 3}) = 1$.

1. Using the law of total probability

$$\begin{aligned} P(\text{Heads}) &= P(\text{Heads} \mid \text{Coin 1})P(\text{Coin 1}) + \\ &\quad P(\text{Heads} \mid \text{Coin 2})P(\text{Coin 2}) + P(\text{Heads} \mid \text{Coin 3})P(\text{Coin 3}) \\ &= \frac{1}{2} \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} \\ &= \frac{2}{3}. \end{aligned}$$

2. Using Bayes' theorem

$$\begin{aligned} P(\text{Coin 1} \mid \text{Heads}) &= \frac{P(\text{Heads} \mid \text{Coin 1})P(\text{Coin 1})}{P(\text{Heads})} \\ &= \frac{1 \cdot \frac{1}{3}}{\frac{2}{3}} \\ &= \frac{1}{2}. \end{aligned}$$

Exercise 1.4. You see a sponsored post online with the word *bitcoin* in. You want to work out the probability the post is spam.

1. Using the law of total probability, show the probability the post is spam, given it contains the word probability is

$$\pi(\text{spam} \mid \text{bitcoin}) = \frac{\pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam})}{\pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam}) + \pi(\text{bitcoin} \mid \text{not spam})\pi(\text{not spam})}$$

2. Most spam filters take a naive approach and set

$$\pi(\text{spam}) = \pi(\text{not spam}) = \frac{1}{2}.$$

If a post is known to be spam, there's an 80% chance it contains the word bitcoin. If a post is not spam, then there's a 1% chance it contains the word bitcoin. Calculate the probability the post is spam given it contains bitcoin.

3. Suppose you take a much more pessimistic view, and assume that 80% of all sponsored posts are spam. Recalculate the probability the post is spam given it contains bitcoin.

Solution. By Bayes' theorem, we have

$$\pi(\text{spam} \mid \text{bitcoin}) = \frac{\pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam})}{\pi(\text{bitcoin})}.$$

1. By the law of total probability,

$$\pi(\text{bitcoin}) = \pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam}) + \pi(\text{bitcoin} \mid \text{not spam})\pi(\text{not spam}).$$

Thus,

$$\pi(\text{spam} \mid \text{bitcoin}) = \frac{\pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam})}{\pi(\text{bitcoin} \mid \text{spam})\pi(\text{spam}) + \pi(\text{bitcoin} \mid \text{not spam})\pi(\text{not spam})}$$

2. From the question, we have $\pi(\text{spam}) = \pi(\text{not spam}) = \frac{1}{2}$, $\pi(\text{bitcoin} \mid \text{spam}) = 0.8$ and $\pi(\text{bitcoin} \mid \text{not spam}) = 0.01$. Plugging these into the probability gives

$$\pi(\text{spam} \mid \text{bitcoin}) = \frac{80}{81} \approx 98.7\%.$$

3. This time $\pi(\text{spam}) = 0.8$ and $\pi(\text{not spam}) = 0.2$, which yields

$$\pi(\text{spam} \mid \text{bitcoin}) = \frac{320}{321} \approx 99.7\%.$$

Exercise 1.5. You are working on a project investigating pollution related illnesses in the West Midlands. I have sampled the proportion of people with pollution related illnesses in five areas of the West Midlands, y_1, \dots, y_5 , with nothing to distinguish the data. This exercise is about the last data point y_5 .

1. Should you model these data points exchangeably?
2. I now tell you the first four of these rates (0.72, 1.00, 0.85, 0.78 per 100,000). Should you continue to model these data points exchangeably?
3. Now, suppose instead of telling you these four rates, I had told you the five areas I have information about are Birmingham City Centre, Smethwick, Edgbaston, Dorridge, and Sutton Coldfield. Should you continue to model these data points exchangeably?
4. Now suppose I give you the data in part 2 and say that the missing data point y_5 is Birmingham City Centre. Should you continue to model these data points exchangeably?

Solution. This is based on Gelman et. al (2013, p. 105).

1. Yes, you have no information to distinguish the data points so exchangeability seems a reasonable assumption.
2. Yes, you still have no information to distinguish between any of the data points.
3. Yes, the joint distribution of these variables doesn't depend on the labels. However, you may start to formulate prior beliefs. Birmingham City Centre and Smethwick are likely to have higher rates of air pollution than leafy Dorridge.

4. No, these can no longer be modelled exchangeably. You have reason to believe that Birmingham City Centre is likely to have a substantially higher value than the rest. That is you have information about $\pi(y_5 > \max(y_1, y_2, y_3, y_4) \mid y_1, y_2, y_3, y_4)$, so exchangeability is no longer a suitable assumption.

Chapter 2

Programming in R

2.1 Random Numbers, For Loops and R

This first computer lab is about getting used to R. The first step is to download R and Rstudio.

- Download R
- Download RStudio IDE

The easiest way to learn R is by using it to solve problems. The lab contains four exercises and three ways of approaching the exercise (easy, medium and hard). If you're new to R, use the easy approach and copy and paste the code straight into R – you'll need to fill in a few blanks though. If you've used R before, or a similar programming language, stick to the medium and hard approaches. This is also an exercise in using Google. Googling around a problem or for specific commands can allow you to quickly find examples (most likely on Stack Overflow) with code you can use.

There are three aims of this lab:

1. Getting used to programming in R.
2. Generating random numbers in R.
3. Creating for loops in R.

Example 2.1. Computationally verify that the Poisson distribution with rate $\lambda = 100$ can be approximated by a normal distribution with mean and variance 100.

To do this, we can generate lots of samples from a `Poisson(100)` distribution and plot them on top of the density function of the normal distribution with mean and variance 100.

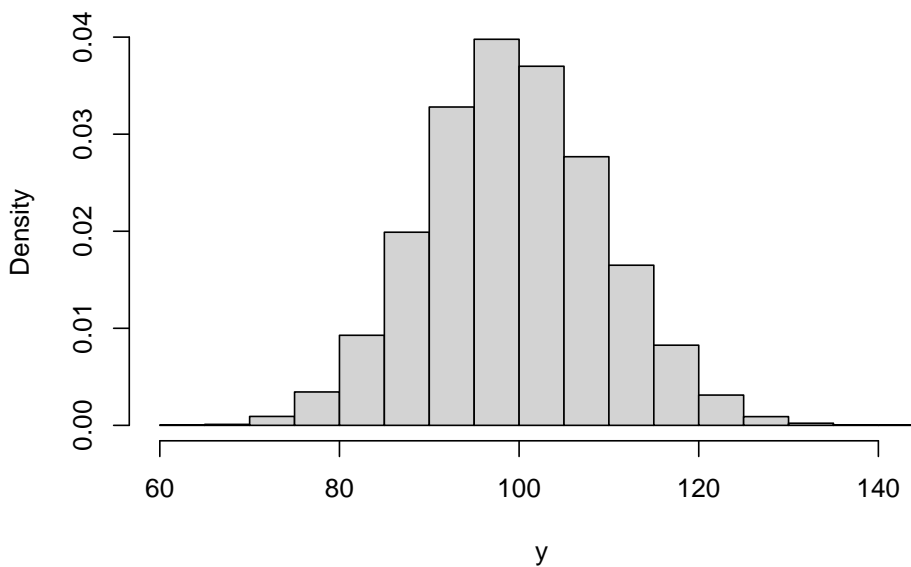
The function `rpois` allows us to generate samples from a Poisson distribution.

We store 10,000 samples in a vector `y` by calling

```
y <- rpois(n = 10000, lambda = 100)
```

We can generate a histogram of `y` using the `hist` command. Setting `freq = FALSE`, makes R plot a density histogram instead of a frequency histogram. Typing `?hist` will give you more information about this

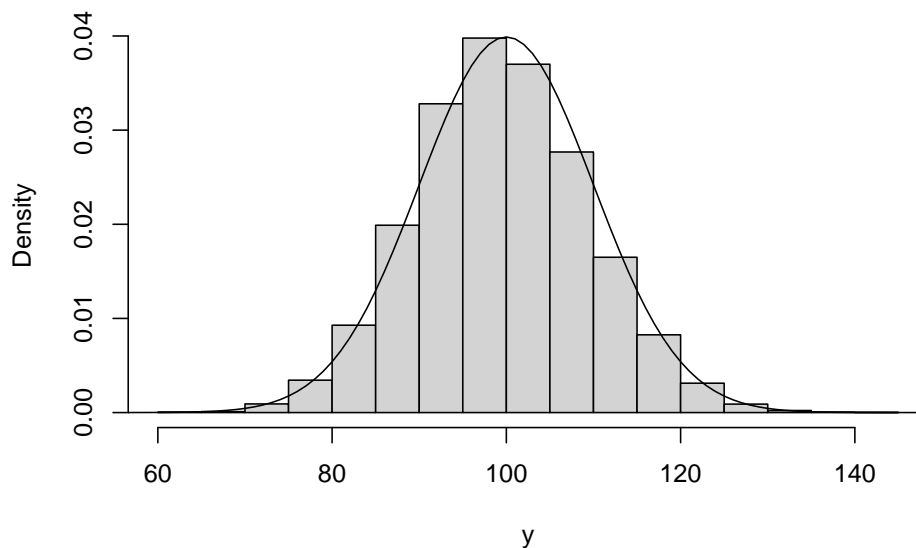
```
hist(y, freq = FALSE, xlab = "y", main = "")
```



The last thing to do is to plot the normal density on top. There are a couple of ways of doing this. The way below generates a uniform grid of points and then evaluates the density at each point. Finally, it adds a line graph of these densities on top.

```
x <- seq(from = 50, to = 150, by = 1)           #create uniform grid on [50, 150]
density <- dnorm(x, mean = 100, sd = sqrt(100)) #compute density

#plot together
hist(y, freq = FALSE, xlab = "y", main = "")
lines(x, density)
```

The two match up well, showing the normal distribution is a suitable approximation here.

Over the next two sessions, you will need to solve the following four problems in R. You can type `?` before any function in R (e.g. `?rnorm`) to bring up R's helpage on the function. Googling can also bring up lots of information, possible solutions and support.

Exercise 2.1. The changes in the Birmingham stock exchange each day can be modelled using a normal distribution. The price on day i , X_i is given by

$$X_i = \alpha X_{i-1}, \quad \alpha \sim N(1.001, 0.005^2).$$

The index begins at $X_0 = 100$. Investigate the distribution of the value of the stock market on days 50 and 100.

Hard. Use a simulation method to generate the relevant distributions.

Medium. Simulate the value for α for each of the 100 days and use the `cumprod` command to plot a trajectory. Use a for loop to repeat this 100 times and investigate the distribution of the value of the stock market on days 50 and 100.

Easy. Fill in the blanks in the following code.

```
# Plot one -----
x <- rnorm(n = , mean = , sd = ) #Simulate daily change for 100 days
plot(, type = 'l') #multiply each day by the previous days

# Plot 100 realisations -----
market.index <- matrix(NA, 100, 100) #Initialise a matrix to store trajectories
for(i in 1:100){
  x <- rnorm(n = , mean = , sd = )
```

```

    market.index[, i] <-
}

#Plot all trajectories
matplot(prices, type = 'l')

#Get distribution of days 50 and 100
hist()
hist()
quantile(, )
quantile(, )

```

Exercise 2.2. You are an avid lottery player and play the lottery twice a week, every week for 50 years (a total of 5,200 times). The lottery has 50 balls labeled 1, ..., 50 and you play the same 6 numbers each time. Six out of the 50 balls are chosen uniformly at random and the prize money is shown in the table below.

Numbers Matched	Prize Amount
0-2	£0
3	£30
4	£140
5	£1,750
6	£1,000,000

It costs you £2 to play each time. Simulate one set of 5,200 draws. How much do you win? What is your total profit/loss?

Hard. Use a for loop and sequence of if else statements to generate your prize winnings.

Medium. Use a for loop to generate the lottery numbers and prize winnings for each draw. Use the `sample` function to generate a set of lottery numbers and check they match against your numbers using the `%in%` function. Finally, use if else statements to check how much you have won each time.

Easy. Fill in the blanks in the following code.

```

my.numbers <-

#For loop to generate lottery numbers and prize winnings
prize <- numeric(5200)
for(i in 1:5200){

    #Generate lottery numbers
    draw <- sample(, )

    #Check how many match my numbers

```

```

numbers.matched <- #use %in% function

#Compute prize winings
if(numbers.matched < 3)
  prize[i] <- 0
else if()
  prize[i] <- 30
else if()
  prize[i] <- 140
else if()
  prize[i] <- 1750
else
  prize[i] <- 1000000
}

```

```

#Summarise prize winnings
table(prize)
hist(prize)
sum(prize) - 2*5200

```

Exercise 2.3. Estimate π .

Hard. Use a rejection sampling algorithm.

Medium. Generate lots of points (x, y) on the unit square $[0, 1]^2$. Check each point to see if it lies within the unit circle. Use the proportion of points that lie within the unit circle to estimate π .

Easy. Fill in the blanks in the following code.

```

#Sample on unit square
N <- 10000      #number of points
x <-           #sample N points uniformly at random on [0, 1]
y <-           #sample N points uniformly at random on [0, 1]

#Estimate pi
r.sq           <- x^2 + y^2           #check how far from origin
number.inside.circle <-              #count how many points inside unit circle
pi.estimate    <-

#Plot points
par(pty = "s")      #make sure plot is square
plot(x, y, cex = 0.1) #plot points
theta <- seq(0, pi/2, 0.01) #plot unit circle
lines(x = cos(theta), y = sin(theta), col = "red")

```

Extra. Use a for loop to repeat this for $N = \{1, \dots, 10000\}$. Record the estimate for π for each value and the relative error.

Exercise 2.4. A linear congruential generator (LCG) is a simple algorithm for generating random integers. Given a starting value X_0 , it generates a sequence of integers according to

$$X_{i+1} = aX_i + c \pmod{m}.$$

Software that generates numbers using an LCG Setting $a = 3$, $c = 2$, $m = 7$ and $X_0 = 0$, generate 20 samples from this generator.

1. Investigate the ‘randomness’ of this generator by creating the delay plot, where X_{i-1} is plotted against X_i
2. One way to improve the quality of these generators is to shuffle the sequence generated. Generate two sequences X and Y from two different LCGs, and report the shuffled sequence $Z_j = X_{Y_j}$. For the sequence Y use the values $a = 5$, $c = 1$, $m = 8$ and $Y_0 = 2$.
3. As the past two exercises show, LCGs are notoriously poor. In the 1960s and 70s, RANDU was a widely used LCG developed by IBM. According to Wikipedia > IBM’s RANDU is widely considered to be one of the most ill-conceived random number generators ever designed, and was described as “truly horrible” by Donald Knuth.

The RANDU LCG uses $a = 2^{16} + 3$, $c = 0$, $m = 2^{31}$ and $Y_0 = 1$. Generate a sequence of 10,000 pseudorandom variables from the RANDU LCG and create the delay plot.

The delay plot seems to show little relationship between X_i and X_{i+1} . The third order delay plot is a 3d-plot with coordinate (X_i, X_{i+1}, X_{i+2}) and this plot shows a different picture. Create this plot using the code

```
#install.packages("scatterplot3d") #you may need to install this package
scatterplot3d::scatterplot3d(X[1:9998], X[2:9999], X[3:10000], angle=154,
                             xlab = expression(X[i]), ylab = expression(X[i+1]), zlab =
```

This is what makes the RANDU LCG so poor. Write down X_{i+1} and X_{i+2} in terms of X_i . Show that $X_{i+2} = \alpha X_{i+1} + \beta X_i$.

Hard. Use a for loop to construct sequences from the LCGs X and Y .

Medium. Create a for loop to generate the value for the sequence X_i for $i = 1, \dots, 20$. Modular arithmetic can be performed using the `%` function. Create a new for loop to construct the sequence Y . To shuffle the sequence X using Y , you will need to subset X by Y in R.

Easy. Fill in the blanks in the code below

```
# 1. Shuffling -----
X <- numeric(21) #initialise vector to store X

#Set values for LCG
a <-
```

```

c <-
m <-
X[1]<-

#Run Generator
for(i in 2:21){
  X[i] <-
}
X

#Delay plot
plot( , , xlab = expression(X[i-1]), ylab = expression(X[i]), type = 'l')

# 2. Shuffling -----
Y <- numeric(21) #initialise vector to store Y

#Set values for LCG
a <-
c <-
m <-
Y[1]<-

#Run Generator
for(i in 2:50){
  Y[i] <-
}

#report sequence
Y
X[Y]

#Plot delay plot
plot(x = ,y = , xlab = expression(Z[i-1]), ylab = expression(Z[i]), type = 'l')

```

2.2 Functions in R

The purpose of this lab is to learn how to write functions in R. Functions are wrappers that allow you to easily repeat commands, as well as customise specific pieces of code.

2.2.1 Built in commands

R has many built in commands and you used these in Computer Lab I. An example is the `runif` command from the second exercise. This function generates random numbers from an interval. The code chunk below shows it in action:

```
u <- runif(n = 10, min = -1, max = 1)
u

## [1] 0.07787615 0.69962469 0.16857827 -0.16481413 0.29819436 0.27621316
## [7] 0.76670538 -0.56352839 -0.95361879 -0.35903449
```

The function has three **arguments**: *n* the number of samples to be generated, *min* the lower limit of the interval, *max* the upper limit of the interval. In the code chunk 10 random numbers were generated from the interval $[-1, 1]$. In R, you don't need to label the arguments, so this will sample the same number of samples from the same interval

```
u <- runif(10, -1, 1)
```

Although it helps to label the arguments in many cases.

2.2.2 User defined functions

In many cases, we will need to repeat the same piece of code over and over again, or we will need to run it again with different values. In this case, we can write our own function. In R, there are two ways to type your own function. The first is to write a full function definition. The basic template is

```
name.of.function <- function(arguments){

  #do something
  #produce result

  return(result)

}
```

The second way is an in-line function, which is sometimes useful for short functions. The template is

```
name.of.function <- function(arguments) #do something
```

In this module, we're going to use the full function way of writing functions.

Example 2.2. In this example, we're going to write a function to evaluate the normal density function. The density function is given by

$$\pi(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\{-\frac{1}{2\sigma^2}(x-\mu)^2\}}.$$

We will need our function to take three arguments, the value at which the density function needs to be evaluated, and the mean and standard deviation of the distribution.

```
normal.density <- function(x, mu, sigma){

  fraction.term <- 1/sqrt(2*pi*sigma^2)
  exponent.term <- -1/(2*sigma^2)*(x-mu)^2

  result <- fraction.term*exp(exponent.term)
  return(result)

}
```

We have split up the density into two parts to make it easier to code up and read. R has its own inbuilt normal density function `dnorm` and we can compare our function against R's. Although R's is faster and more reliable, we should get the same results.

```
normal.density(x = 0.5, mu = 1, sigma = 0.5)
```

```
## [1] 0.4839414
```

```
dnorm(x = 0.5, mean = 1, sd = 0.5)
```

```
## [1] 0.4839414
```

Why might R's function be faster and more reliable than ours?

Exercise 2.5. Consider the stock exchange problem in Exercise 2.1. Write a function that simulates 100 days of the stock exchange. Use the `replicate` function to call this function 10,000 times.

Exercise 2.6. In a game, players can buy loot boxes. Each box contains one item chosen at random from fifty loot items in the game. By simulating lots of players buying loot boxes, what is the expected number of boxes a player needs to buy in order to collect all fifty items.

1. Write a function that buys a new box and, if the player doesn't own it already, adds it to the set of items the player doesn't own already.
2. Write another function using a while loop that simulates a player buying boxes until they own all the items.
3. Use a for loop to simulate 10,000 players buying boxes until they own all the items.
4. It turns out there are three types of items in the loot boxes. 25 items are common, 20 are rare, and 5 are epic. Players are twice as likely to get a common item than a rare item, and four times as likely to get a rare item

than an epic item. Modify the function from part 1 to account for this and repeat the simulation.

2.3 Good Coding Practices

In the past two labs, we've written code to solve different problems. In this lab, we're going to take a step back and think about what good R code does and doesn't look like.

2.3.1 Code Style

Code should be both efficient and easy to read. In most cases it's better to write code that's easy to read and less efficient, than highly efficient code that's difficult to read. Some basic principles to make code easy to read are:

1. Write short function names, e.g. `buy.loot.box` is better than `player.buys.one.loot.boox` or `blb`.
2. Document and comment code. In R comments start with `#`.
3. Multiple short functions are better than long functions that do multiple things.
4. Be consistent.

Example 2.3. Review the tidyverse Style Guide.

Example 2.4. Review Google's R Style Guide.

One way to ensure code style is consistent and bug free is to carry out code reviews. These are common both in academia and industry. A code review is where someone else goes through your code line-by-line ensuring it conforms to the company style and doesn't have any bugs.

Exercise 2.7. The following code is for Exercises 2.1 about the stock exchange. Restyle the code so it is easy to read.

```
# Plot one -----
rnorm(100,1.001,0.5) -> x
plot(100*cumprod(x),type='l')
# Plot 100 realisations -----
X <- matrix(NA, 100, 100)
for(i in 1:100){
  x <- rnorm(100, 1.001, 0.005)
  X[,i] <- 100*cumprod(x)
}
matplot(prices,type='l')
hist(prices[50,]);hist(prices[100,]);quantile(prices[50,], c(0.25, 0.5, 0.75));quantil
```

Exercise 2.8. In pairs or groups, carry out a code review for one of your solutions to an exercise from a previous lab. Remember to

1. Make sure the coding style is consistent.
2. Identify any bugs.
3. Be respectful and constructive in your feedback.

Chapter 3

Bayesian Inference

Whereas Chapter 1 dealt with the fundamentals of Bayesian inference and definitions, Chapter 2 is much more practical. We're going to be deriving posterior distributions and proving when it does and doesn't work.

3.1 The Binomial Distribution

The first example we're going to go through is with the binomial distribution.

Example 3.1. A social media company wants to determine how many of its users are bots. A software engineering collects a random sampler of 200 accounts and finds that eight are bots. She uses a Bayesian method to estimate the probability of an account being a bot. She labels the accounts with a 1 if they are a bot and 0 if there is a real person. The set of account labels is given by $y = \{y_1, \dots, y_{200}\}$ and the probability an account is a bot is θ . By Bayes' theorem

$$\pi(\theta \mid y) \propto \pi(y \mid \theta)\pi(\theta).$$

Likelihood function $\pi(y \mid \theta)$. We observe a 200 trials each with a probability of success or failure. The binomial distribution seems the most suitable way of modelling this. Therefore

$$\pi(y \mid \theta) = \binom{200}{3} \theta^3 (1 - \theta)^{197}$$

We are going to assume the accounts are independent.

Prior distribution $\pi(\theta)$. We now need to describe our prior beliefs about θ . We have no reason to suggest θ takes any specific value, so we use a uniform prior distribution $\theta \sim U[0, 1]$, where $\pi(\theta) = 1$ for $\theta \in [0, 1]$.

Posterior distribution $\pi(\theta | y)$. We can now derive the posterior distribution up to proportionality

$$\pi(\theta | y) \propto \theta^3(1 - \theta)^{197}.$$

This functional dependence on θ identifies the $\pi(\theta | y)$ as a Beta distribution. The PDF for the beta distribution with shape parameters α and β is

$$\pi(x | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1 - x)^{\beta-1}.$$

The posterior distribution is therefore $\theta | y \sim \text{Beta}(4, 198)$.

3.2 Reporting Concluisions from Bayesian Inference

In the previous example, we derived the posterior distribution $\theta | y \sim \text{Beta}(4, 198)$. But often, we want to share more descriptive information about our beliefs given the observed data. In this example, the posterior mean given the data is $\frac{4}{198} = \frac{2}{99}$. That is to say given the data, we expect that for every 99 accounts, two to be bots. The posterior mode for θ is $\frac{3}{200}$ or 1.5%.

It is important to share the uncertainty about our beliefs. In a frequentist framework, this would be via a confidence interval. The Bayesian analogue is a credible interval.

Definition 3.1. A **credible interval** is a central interval of posterior probability which corresponds, in the case of a $100(1 - \alpha)\%$ interval, to the range of values that capture $100(1 - \alpha)\%$ of the posterior probability.

Example 3.2. The 95% credible interval for the Binomial example is given by

```
cred.int.95 <- qbeta(c(0.025, 0.975), 4, 198)
round(cred.int.95, 3)
```

```
## [1] 0.005 0.043
```

This says that we believe there is a 95% chance that the probability of an account being a bot lies between 0.005 and 0.043. This is a much more intuitive definition to the confidence interval, which says if we ran the experiment an infinite number of times and computed an infinite number of confidence intervals, 95% of them would contain the true value of θ .

3.3 The Exponential Distribution

Example 3.3. An insurance company want to estimate the time until a claim is made on a specific policy. They describe the rate at which claims come in by λ . The company provides a sample of 10 months at which a claim was made

$y = \{14, 10, 6, 7, 13, 9, 12, 7, 9, 8\}$. By Bayes' theorem, the posterior distribution for λ is

$$\pi(\lambda \mid y) \propto \pi(\lambda \mid y)\pi(\lambda)$$

Likelihood function $\pi(\lambda \mid y)$. The exponential distribution is a good way of modelling lifetimes or the length of time until an event happens. Assuming all the claims are independent, the likelihood function is given by

$$\begin{aligned}\pi(y \mid \lambda) &= \prod_{i=1}^{10} \lambda e^{-\lambda y_i} \\ &= \lambda^{10} e^{-\lambda \sum_{i=1}^{10} y_i} \\ &= \lambda^{10} e^{-95\lambda}\end{aligned}$$

Prior distribution $\pi(\lambda)$. As we are modelling a rate parameter, we know it must be positive and continuous. We decide to use an exponential prior distribution for λ , but leave the choice of the rate parameter up to the insurance professionals at the insurance company. The prior distribution is given by $\pi(\lambda) \sim \text{Exp}(\gamma)$.

Posterior distribution $\pi(\lambda \mid y)$. We now have all the ingredients to derive the posterior distribution. It is given by

$$\begin{aligned}\pi(\lambda \mid y) &\propto \lambda^{10} e^{-95\lambda} \times \lambda e^{-\gamma\lambda} \\ &\propto \lambda^{11} e^{-(95+\gamma)\lambda}\end{aligned}$$

The functional form tells us that the posterior distribution is a Gamma distribution. The PDF of a gamma random variable with shape α and rate β is

$$\pi(x \mid \alpha, \beta) = \frac{\alpha^\beta}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}.$$

The distribution of the rate of the claims given the observed data is $\lambda \mid y \sim \text{Gamma}(10, 95 + \gamma)$.

The posterior mean months until a claim is $\frac{10}{95+\gamma}$. We can see the effect of the choice of rate parameter in this mean. Small values of γ yield vague prior distribution, which plays a minimal role in the posterior distribution. Large values of γ result in prior distributions that contribute a lot to the posterior distribution. The plots below show the prior and posterior distributions for $\gamma = 0.01$ and $\gamma = 50$.

```
plot.distributions <- function(chi){
  #evaluate at selected values of theta
  theta <- seq(0.001, 0.3, 0.001)

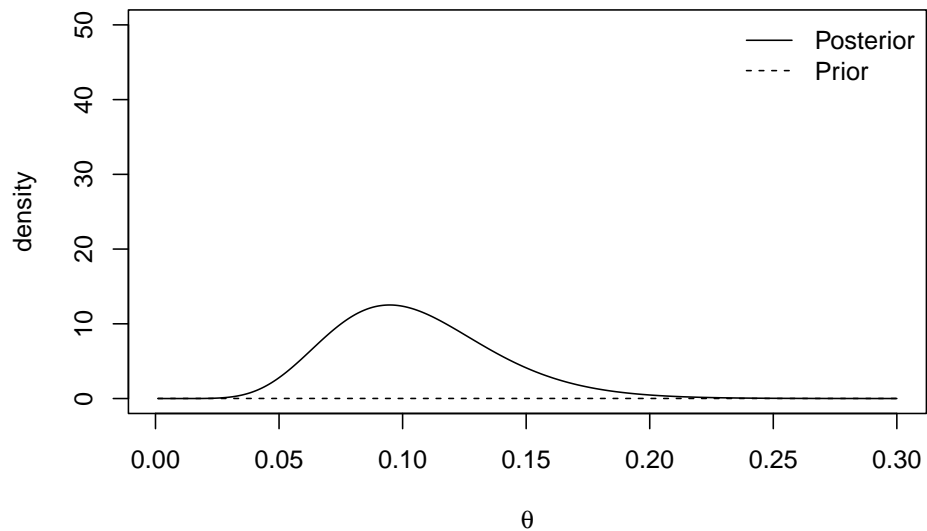
  #evaluate prior density
```

```
prior <- dexp(theta, rate = chi)

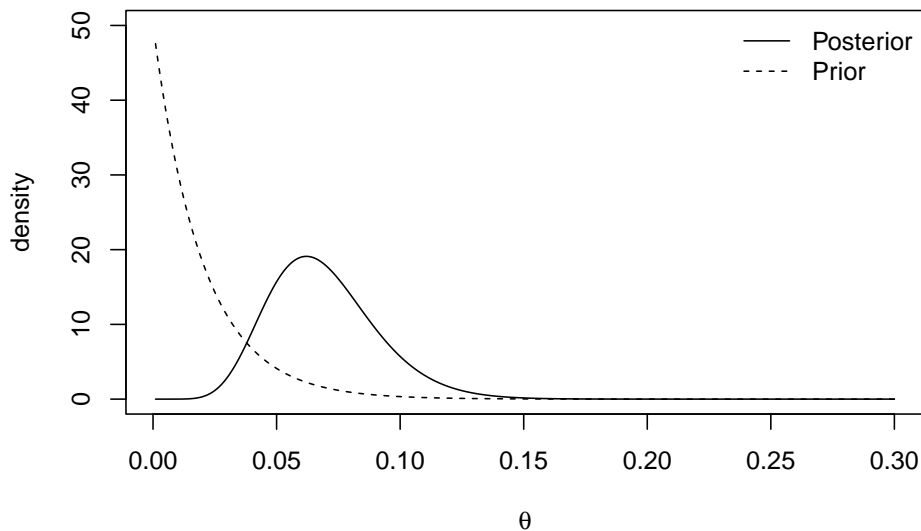
#evaluate posterior density
posterior <- dgamma(theta, shape = 10, rate = 95 + chi)

#plot
plot(theta, posterior, type= 'l',
      ylim = c(0, 50), xlab = expression(theta), ylab = "density")
lines(theta, prior, lty = 2)
legend('topright', lty = c(1, 2), legend = c("Posterior", "Prior"),
      bty = "n")
}
```

```
plot.distributions(0.01)
```



```
plot.distributions(50)
```



The insurance managers recommend that because this is a new premium, a vague prior distribution be used and $\gamma = 0.01$. The posterior mean is $\frac{10}{95.01} \approx 0.105$ and the 95% credible interval is

```
round(qgamma(c(0.025, 0.975), 10, 95.01), 3)
```

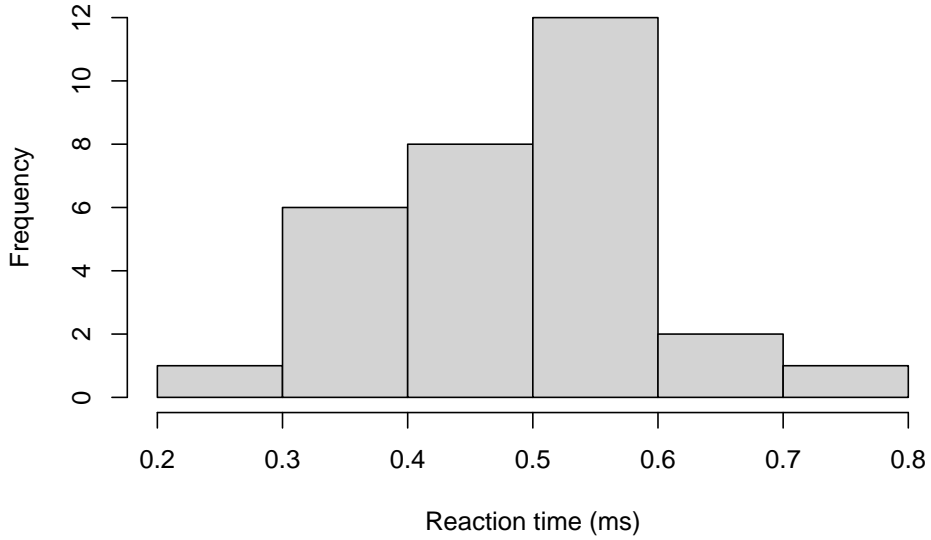
```
## [1] 0.05 0.18
```

3.4 The Normal Distribution

The Normal distribution is incredibly useful for modelling a wide range of natural phenomena and in its own right. We're now going to derive posterior distributions for the normal distribution. As we're going to see, the concepts behind deriving posterior distributions are the same as in the previous two examples. However, the algebraic accounting is a lot more taxing.

Example 3.4. Reaction times can be modeled with a normal distribution. Suppose we have a data set of the reaction times of 30 lorry drivers when they see an obstacle. The reaction times were collected in a test environment on a rolling road. The time until each lorry driver reacts (in milliseconds) is

```
y <- c(0.34, 0.47, 0.58, 0.27, 0.74, 0.44, 0.46, 0.65, 0.36, 0.55, 0.58, 0.55, 0.53, 0.56, 0.54,
hist(y, main = "", xlab = "Reaction time (ms)"))
```



```
mean(y)
```

```
## [1] 0.4896667
```

Suppose that, somehow, we know the population standard deviation is $0.01ms$ and we wish to estimate the population mean μ . By Bayes' theorem, the posterior distribution is

$$\pi(\mu | y, \sigma^2) \propto \pi(y | \mu, \sigma^2) \pi(\mu)$$

Likelihood function. We assume the each driver's reaction time is independently and identically distributed such that

$$y_i \sim N(\mu, \sigma^2)$$

The likelihood function is therefore given by the product of the 30 normal density functions as follows

$$\begin{aligned} \pi(y | \mu, \sigma^2) &= \prod_{i=1}^{30} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - \mu)^2}{\sigma^2} \right\} \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp \left\{ -\sum_{i=1}^{30} \frac{(y_i - \mu)^2}{\sigma^2} \right\} \end{aligned}$$

Prior distribution We suppose we have no prior beliefs about the values that μ can take. We assign a normal prior distribution to $\mu \sim N(\mu_0, \sigma_0^2)$ despite it being a time. We will set $\mu = 0$ and $\sigma_0^2 = 1000$ to signify our vague prior beliefs, but, for ease, we will use the symbolic values during the derivation of the posterior distribution. We have

$$\pi(\mu) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp \left\{ -\frac{1}{2\sigma_0^2} (\mu - \mu_0)^2 \right\}.$$

Posterior distribution. To derive the posterior distribution, up to proportionality, we multiply the prior distribution by the likelihood function. As the fractions out the front of both terms do not depend on μ , we can ignore these.

$$\begin{aligned}\pi(\mu \mid y, \sigma^2) &\propto \exp \left\{ -\sum_{i=1}^{30} \frac{(y_i - \mu)^2}{\sigma^2} \right\} \exp \left\{ \frac{1}{2\sigma_0^2} (\mu - \mu_0)^2 \right\} \\ &= \exp \left\{ -\sum_{i=1}^{30} \frac{(y_i - \mu)^2}{\sigma^2} - \frac{1}{2\sigma_0^2} (\mu - \mu_0)^2 \right\} \\ &= \exp \left\{ -\frac{\sum_{i=1}^{30} y_i^2}{2\sigma^2} + \frac{\mu \sum_{i=1}^{30} y_i}{\sigma^2} - \frac{30\mu^2}{2\sigma^2} - \frac{\mu^2}{2\sigma_0^2} + \frac{\mu\mu_0}{\sigma_0^2} - \frac{\mu_0^2}{2\sigma_0^2} \right\}.\end{aligned}$$

We can drop the first and last term as they do not depend on μ . With some arranging, the equation becomes

$$\pi(\mu \mid y, \sigma^2) \propto \exp \left\{ -\mu^2 \left(\frac{30}{2\sigma^2} + \frac{1}{2\sigma_0^2} \right) + \mu \left(\frac{\sum_{i=1}^{30} y_i}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right) \right\}$$

Defining $\mu_1 = \left(\frac{\sum_{i=1}^{30} y_i}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right)$ and $\sigma_1^2 = \left(\frac{30}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1}$ tidies this up and gives

$$\pi(\mu \mid y, \sigma^2) \propto \exp \left\{ -\frac{\mu^2}{2\sigma_1^2} + \mu\mu_1 \right\}.$$

Our last step to turning this into a distribution is completing the square. Consider the exponent term, completing the square becomes

$$-2\sigma_1^2\mu^2 + \mu\mu_1 = -\frac{1}{2\sigma_1^2} \left(\mu - \frac{\mu_1}{\sigma_1^2} \right)^2.$$

Therefore, the posterior distribution, up to proportionality, is given by

$$\pi(\mu \mid y, \sigma^2) \propto \exp \left\{ -\frac{1}{2\sigma_1^2} \left(\mu - \frac{\mu_1}{\sigma_1^2} \right)^2 \right\},$$

and so the posterior distribution of μ is $\mu \mid y, \sigma^2 \sim N(\mu_1, \sigma_1^2)$.

It may help to consider the meaning of μ_1 and σ_1^2 . The variance of the posterior distribution can be thought of as the weighted average of the population and sample precision, where the weight is the number of data points collected. The interpretation of the posterior mean can be seen more easily by writing is as

$$\mu = \sigma_1^2 \left(\frac{30\bar{y}}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right).$$

The posterior mean is partially defined through the weighted average of the population and prior means, where the weighting depends on the number of data points collected and how precise the distributions are.

Now we have derived the posterior distribution, we can explore it using R.

```
#data
N <- 30

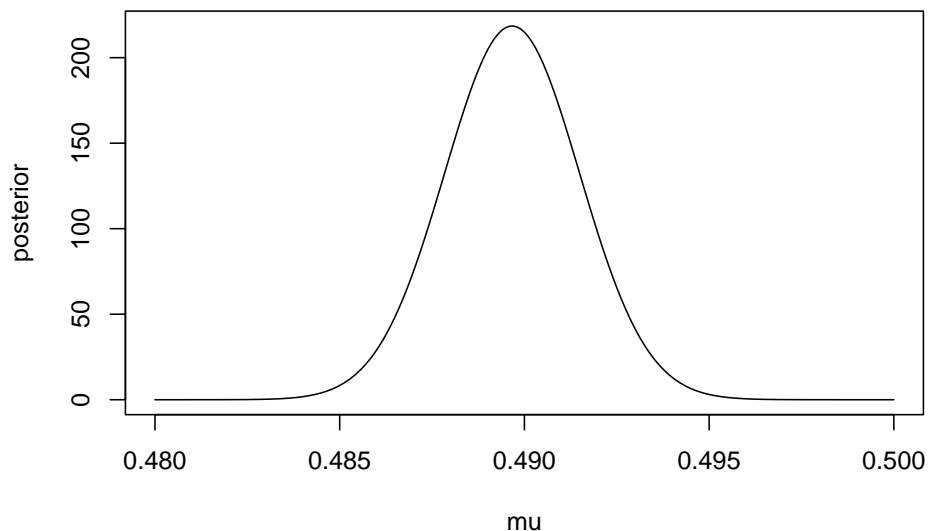
#prior
sigma0 <- 1000
mu0 <- 0

#posterior
sigma1.sq <- (1/(sigma0^2) + N/(0.01^2))^-1
mu1 <- sigma1.sq*(sum(y)/(0.01^2) + mu0/(sigma0^2))

c(mu1, sigma1.sq) #output mean and variance

## [1] 4.896667e-01 3.333333e-06

#Create plot
mu <- seq(0.48, 0.5, 0.0001)
posterior <- dnorm(mu, mean = mu1, sd = sqrt(sigma1.sq))
plot(mu, posterior, type = 'l')
```



The 95% credible interval for the population's mean reaction time is

```
qnorm(c(0.025, 0.975), mu1, sqrt(sigma1.sq))
```

```
## [1] 0.4860883 0.4932451
```

One issue in this example is the choice of the prior distribution for μ . Why are we putting a prior distribution that places weight on negative values, when we are modelling reaction times? We could argue that the resulting posterior distribution places negligible weight on invalid times. The real reason is analytical

ease. The resulting posterior distribution has a nice closed form, the normal distribution. When the prior distribution induces the same function form in the posterior distribution, this is known as conjugacy.

If the prior distribution $\pi(\theta)$ has the same distributional family as the posterior distribution $\pi(\theta \mid y)$, then the prior distribution is a **conjugate prior distribution**.

3.5 Heirachical Models

In many modelling problems, there will be multiple parameters each related to one another. These parameters may be directly related to the model, or they may be parameters we introduce through prior distributions. We can form a hierarchy of these parameters, from closest to further from the data, to construct our model.

Example 3.5. Let's consider @{exm:exponential} again. We had data y from an Exponential distribution with rate λ . We placed an Exponential prior distribution with rate γ on λ and the posterior distribution was $\lambda \mid y \sim \text{Gamma}(10, 95 + \gamma)$.

In that example, we discussed how the choice of γ can affect the posterior distribution and conclusions presented to the company. One option is to place a prior distribution on γ – a hyperprior distribution. The heirachy formed is

$$\begin{array}{ll} y \mid \lambda \sim \text{Exp}(\lambda) & \text{(likelihood)} \\ \lambda \mid \gamma \sim \text{Exp}(\gamma) & \text{(prior distribution)} \\ \gamma \mid \nu \sim \text{Exp}(\nu) & \text{(hyperprior distribution)} \end{array}$$

.

By Bayes' theorem, we can write the posterior distribution as

$$\pi(\lambda, \gamma \mid y) \propto \pi(y \mid \lambda) \pi(\lambda \mid \gamma) \pi(\gamma) \propto \lambda^{11} e^{-\lambda(95+\gamma)} \nu e^{-\nu\gamma}.$$

To derive the full conditional distributions, we only consider the terms that depends on the parameters we are interested in. The full conditional distribution for λ is

$$\pi(\lambda \mid y, \gamma) \propto \lambda^{11} e^{-\lambda(95+\gamma)}.$$

This is unchanged and shows that $\lambda \mid y, \gamma \sim \text{Gamma}(10, 95 + \gamma)$. The full conditional distribution for γ is

$$\pi(\gamma \mid y, \lambda) \propto e^{-\nu\gamma}.$$

Therefore the full conditional distribution of γ is $\gamma \mid y, \lambda \sim \text{Exp}(\lambda + \nu)$. In the next chapter, we will look at how to sample from these distributions.

Another example is when dealing with a distribution with multiple parameters. In the previous section, we derived the posterior distribution for the mean of a normal distribution when the variance is known. We'll now repeat this exercise, but deriving the posterior distribution for both the mean and variance.

Example 3.6. Consider the reaction time data y from Example @{exm:normal}. Again, we assume the $y_i \sim N(\mu, \sigma^2)$, except that this time, we will place a prior distribution on both μ and σ^2 .

3.6 Prediction

In many cases, although we are interested in estimating the model parameters, what we're really interested in is predicting new values, whose distribution is determined by the model parameters.

Definition 3.2. Suppose we observe some data y and derive the posterior distribution $\pi(\theta | y)$. The quantity we are interested in is some future observation z , we would like to the distribution of z given the observed data y , $\pi(z | y)$. This distribution, known as the **posterior predictive distribution** of z , must be exhibited as a mixture distribution over the possible values of θ . We must write

$$\pi(z | y) = \int \pi(z | \theta) \pi(\theta | y) d\theta.$$

Example 3.7. Students have to submit coursework for a particular statistical modules. However, each semester a number of students miss the deadline and hand in their coursework late. Last year, three out of 20 students handed their coursework in late. This year, the course has thirty students in. How many students can we expect to hand in their coursework late?

We can model the number of students handing in late X , using a Binomial distribution, $Y \sim \text{Bin}(n, \theta)$. As in Example ??, we assign a uniform prior distribution to $\theta \sim U[0, 1]$. Given then observed data, we can derive $\theta | y \sim \text{Beta}(4, 28)$ (See problem sheets for derivation).

Now we can derive the posterior predictive distribution of Z , the number of students who hand in late. We model Z using a Binomial distribution, $Z \sim \text{Bin}(30, \theta)$. The distribution of Z given the observed data is

$$\begin{aligned}
\pi(z | y) &= \int_0^1 \pi(z | \theta) \pi(\theta | y) d\theta \\
&= \int_0^1 \binom{30}{z} \theta^z (1 - \theta)^{30-z} \frac{\Gamma(32)}{\Gamma(4)\Gamma(28)} \theta^3 (1 - \theta)^{27} d\theta \\
&= \binom{30}{z} \frac{\Gamma(32)}{\Gamma(4)\Gamma(28)} \int_0^1 \theta^{z+3} (1 - \theta)^{57-z} d\theta
\end{aligned}$$

This integral is difficult to evaluate immediately. But by multiplying (and dividing outside the integral) by a constant, we can turn it into the density function of a $\text{Beta}(5 + z, 58 - z)$ random variable. This integrates to 1.

$$\begin{aligned}
\pi(z | y) &= \binom{30}{z} \frac{\Gamma(32)}{\Gamma(4)\Gamma(28)} \frac{\Gamma(z+4)\Gamma(58-z)}{\Gamma(62)} \int_0^1 \frac{\Gamma(62)}{\Gamma(z+4)\Gamma(58-z)} \theta^{z+3} (1 - \theta)^{57-z} d\theta \\
&= \binom{30}{z} \frac{\Gamma(32)\Gamma(z+4)\Gamma(58-z)}{\Gamma(4)\Gamma(28)\Gamma(62)} \quad \text{for } 0 \leq z \leq 30.
\end{aligned}$$

This code implements the distribution

```

beta.binom.posterior.predictive.distribution <- function(z){

  numerator <- gamma(32)*gamma(z + 4)*gamma(58-z)
  denominator <- gamma(4)*gamma(28)*gamma(62)

  output <- choose(30, z)*numerator/denominator
  return(output)

}

```

We can check it's correct, by seeing if it sums to one

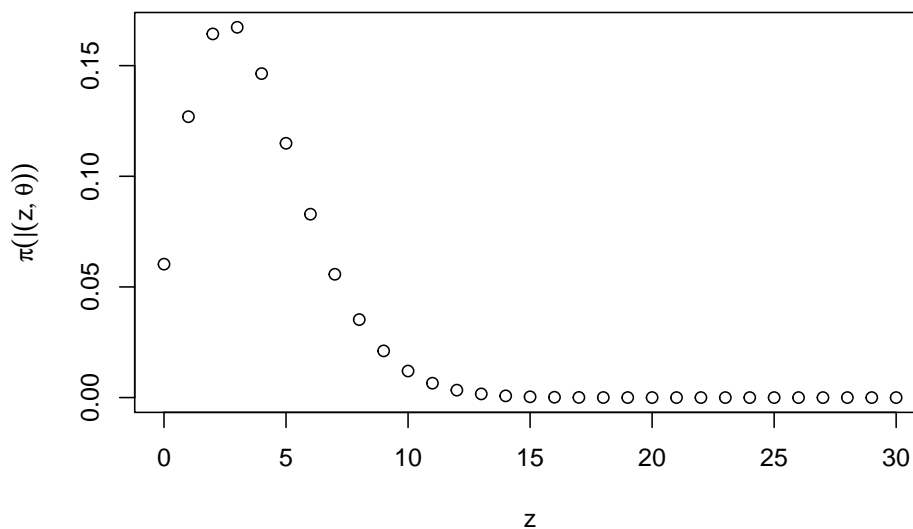
```

z <- 0:30
ppd <- beta.binom.posterior.predictive.distribution(z)
sum(ppd)

```

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

```
plot(z, ppd, xlab = "z", ylab = expression(pi(z | theta)))
```



The expected number of students who hand in late is 3.75 and there's a 95% chance that up to 8 hand in late.

```
z*%ppd #expectation
```

```
##      [,1]
```

```
## [1,] 3.75
```

```
cbind(z, cumsum(ppd)) #CDF
```

```
##      z
## [1,] 0 0.06029453
## [2,] 1 0.18723037
## [3,] 2 0.35156696
## [4,] 3 0.51889148
## [5,] 4 0.66530044
## [6,] 5 0.78021765
## [7,] 6 0.86309065
## [8,] 7 0.91880359
## [9,] 8 0.95404202
## [10,] 9 0.97513714
## [11,] 10 0.98713498
## [12,] 11 0.99363285
## [13,] 12 0.99698773
## [14,] 13 0.99863936
## [15,] 14 0.99941423
## [16,] 15 0.99976022
## [17,] 16 0.99990696
## [18,] 17 0.99996591
## [19,] 18 0.99998826
## [20,] 19 0.99999622
```

```
## [21,] 20 0.99999887
## [22,] 21 0.99999969
## [23,] 22 0.99999992
## [24,] 23 0.99999998
## [25,] 24 1.00000000
## [26,] 25 1.00000000
## [27,] 26 1.00000000
## [28,] 27 1.00000000
## [29,] 28 1.00000000
## [30,] 29 1.00000000
## [31,] 30 1.00000000
```

3.7 Non-informative Prior Distributions

We have seen in a few examples how the choice of prior distribution can affect that posterior distributions and the resulting conclusions. As the choice of prior distribution is subjective, it is the main criticism of Bayesian inference. A possible way around this is to use a prior distribution that reflects a lack of information about θ .

Definition 3.3. A **non-informative prior distribution** is a prior distribution that places equal weight on the every possible value of θ .

Example 3.8. In Example ??, we assigned a uniform prior distribution to the parameter θ .

Theorem 3.1 (Jeffrey). *Given some observed data $y = \{y_1, \dots, y_N\}$, an invariant prior distribution is*

$$\pi(\theta) \propto \sqrt{I_\theta(y)},$$

where $I_\theta(y)$ is the Fisher information for θ contained in y .

Jeffrey argues that if there are two ways of parameterising a model, e.g. via θ and ψ , then the priors on these parameters should be equivalent. In other words, the prior distribution should be invariant under sensible (one-to-one) transformations.

Proof. Recall that the distribution of $\psi = h(\theta)$, for some one-to-one function h , is invariant to the distribution of θ if

$$\pi(\psi) = \pi(\theta) \left| \frac{d\theta}{d\psi} \right|.$$

Transforming the Fisher information for ψ shows

$$\begin{aligned} I_\psi(y) &= -\mathbb{E} \left(\frac{d^2 \log \pi(y | \psi)}{d\psi^2} \right) \\ &= \mathbb{E} \left(\frac{d^2 \log \pi(y | \theta = h^{-1}(\psi))}{d\theta^2} \right) \left(\frac{d\theta}{d\psi} \right)^2 \\ &= I_\theta(y) \left(\frac{d\theta}{d\psi} \right)^2. \end{aligned}$$

Thus $\sqrt{I_\psi(y)} = \sqrt{I_\theta(y)} \left| \frac{d\theta}{d\psi} \right|$ and $\sqrt{I_\psi(y)}$ and $\sqrt{I_\theta(y)}$ are invariant prior distributions. \square

Example 3.9. In example ??, we modelled the number of crisps that were the incorrect weight by $\theta \sim \text{Bin}(n, \theta)$. To construct Jeffrey's prior distribution for θ , we must first derive the Fisher's information matrix.

$$\begin{aligned} \pi(y | \theta) &= \binom{n}{y} \theta^y (1 - \theta)^{n-y} \\ \Rightarrow \log \pi(y | \theta) &= \log \binom{n}{y} + y \log \theta + (n - y) \log(1 - \theta) \\ \Rightarrow \frac{\partial \log \pi(y | \theta)}{\partial \theta} &= \frac{y}{\theta} - \frac{n - y}{1 - \theta} \\ \Rightarrow \frac{\partial^2 \log \pi(y | \theta)}{\partial \theta^2} &= -\frac{y}{\theta^2} + \frac{n - y}{(1 - \theta)^2} \\ \Rightarrow \mathbb{E} \left(\frac{\partial \log \pi(y | \theta)}{\partial \theta} \right) &= -\frac{\mathbb{E}(y)}{\theta^2} + \frac{n - \mathbb{E}(y)}{(1 - \theta)^2} \\ \Rightarrow \mathbb{E} \left(\frac{\partial \log \pi(y | \theta)}{\partial \theta} \right) &= -\frac{n\theta}{\theta^2} + \frac{n - n\theta}{(1 - \theta)^2} \\ \Rightarrow \mathbb{E} \left(\frac{\partial \log \pi(y | \theta)}{\partial \theta} \right) &= -\frac{n}{\theta} + \frac{n}{1 - \theta} \\ \Rightarrow \mathbb{E} \left(\frac{\partial \log \pi(y | \theta)}{\partial \theta} \right) &= -\frac{n}{\theta(1 - \theta)} \\ \Rightarrow I_\theta(y) &\propto \frac{1}{\theta(1 - \theta)}. \end{aligned}$$

§ Hence Jeffrey's prior is $\pi(\theta) \propto \theta^{-\frac{1}{2}}(1 - \theta)^{-\frac{1}{2}}$. This functional dependency on θ shows that $\theta \sim \text{Beta}(\frac{1}{2}, \frac{1}{2})$.

3.8 Bernstein-von-Mises Theorem

So far, we have considered Bayesian methods in contrast to frequentist ones. The Bernstein-von-Mises theorem is a key theorem linking the two inference

methods.

Theorem 3.2 (Bernstein-von-Mises). *For a well-specified model $\pi(y \mid \theta)$ with a fixed number of parameters, and for a smooth prior distribution $\pi(\theta)$ that is non-zero around the MLE $\hat{\theta}$, then*

$$\left\| \pi(\theta \mid y) - N \left(\hat{\theta}, \frac{I(\hat{\theta})^{-1}}{n} \right) \right\|_{TV} \rightarrow 0,$$

where $\|p - q\|_{TV}$ is the total variation distance between distributions p and q :

$$\|p - q\|_{TV} = \frac{1}{2} \int |\pi(x) - q(x)| dx.$$

The Bernstein-von-Mises theorem says that as the number of data points approaches infinity, the posterior distribution tends to a Normal distribution centered around the MLE and variance dependent on the Fisher information.

3.9 Exercises

Exercise 3.1. Suppose $y_1, \dots, y_N \sim \text{Geom}(p)$. 1. Derive the likelihood function and then the maximum likelihood estimator for p . 2. By letting $p \sim \text{Beta}(\alpha, \beta)$, derive the posterior distribution for p given the data. 3. Compare the maximum likelihood estimate with with expectation of the posterior distribution. What values of α and β result in a posterior expectation that is equal to the maximum likelihood estimate? Why is this inadvisable?

Exercise 3.2. When someone is infected with a disease, it's common to model the time they are infectious for t with a Gamma distribution. Suppose you observe 100 measles infections and $\sum_{i=1}^{100} t_i = 870$ days. Based on advice from clinicians, you suppose $t \sim \Gamma(5, \theta)$. Using an $\theta \sim \text{Exp}(0.01)$ prior distribution, derive the posterior distribution. What is the 95% credible interval for θ ?

Exercise 3.3. The density function for the Pareto distribution with scale α and shape β is

$$\pi(x \mid 1, b) = \frac{b}{x^{b+1}}$$

Suppose the data is given by $y = \{y_1, \dots, y_N\}$. Place a Gamma prior distribution on b such that $b \sim \Gamma(\alpha, \beta)$. Derive the posterior distribution for b given the data.

Exercise 3.4. You are given the data $y_1, \dots, y_N \sim \text{Exp}(\lambda)$. Your prior belief is that $\lambda \in (0, 1)$. Show that the posterior distribution $\pi(\lambda \mid y)$ has no closed form when the prior distribution for $\lambda \sim \text{Beta}(\alpha, \beta)$.

Solution. The likelihood function is the product of N exponential density functions

$$\begin{aligned}\pi(y \mid \lambda) &= \prod_{i=1}^N \lambda \exp \{-\lambda y_i\} \\ &= \lambda^N \exp \left\{ -\lambda \sum_{i=1}^N y_i \right\}\end{aligned}$$

The prior distribution is $\pi(\lambda) = \frac{1}{B(\alpha, \beta)} \lambda^{\alpha-1} (1-\lambda)^{\beta-1}$. The posterior distribution is therefore proportional to

$$\pi(\lambda \mid y) \propto \lambda^{\alpha+N-1} (1-\lambda)^{\beta-1} \exp \{-\lambda y_i\}.$$

Hence, the posterior distribution has no closed form.

Exercise 3.5. Suppose that $y_i \sim \text{Binom}(n, p)$ are iid for $i \in \{1, \dots, N\}$.

1. By placing a Beta prior distribution on p such that $p \sim \text{Beta}(\alpha, \beta)$ derive the posterior distribution.
2. Suppose that y_{N+1} is then observed and is also independently drawn from the same distribution. Derive the posterior distribution by updating the previous distribution (i.e. your posterior distribution for the previous part becomes your prior distribution).
3. Show that you obtain the same distribution if you observe all $N+1$ data points at the start of the process.

Exercise 3.6. Let $y_1, \dots, y_N \sim \text{Pois}(\lambda)$.

1. Use Bayes' theorem to derive the posterior distribution of λ given the observed data.
2. Place a Gamma prior distribution on $\lambda \sim \Gamma(\alpha, \beta)$ and derive the form the posterior distribution takes.
3. Discuss the effects of α and β on the posterior distribution.
4. Derive the posterior predictive distribution for a new observation \tilde{y} . *Hint: The mass function for the negative binomial distribution with failures r and probability of success p is*

$$\pi(x = k \mid r, p) = \binom{k+r-1}{k} (1-p)^k p^r.$$

Exercise 3.7. Consider the data from @{exm:normal}. Suppose the police pull over a lorry driver after they failed to stop at a red light. During a test at the side of the road, the lorry driver's reaction time is tested and comes out as 77ms.

1. By constructing a credible interval from the posterior predictive distribution, determine if the driver's reaction time is suspicious.
2. Why is this not a good comparison to make?

Exercise 3.8. A distribution is said to belong to the exponential family of distributions if its density function has the form

$$\pi(y_i | \theta) = f(y_i)g(\theta) \exp\{\nu(\theta)T(x)\}.$$

1. Derive the likelihood function with the data $\{y_1, \dots, y_N\}$.
2. Show that the following prior distribution induces conjugacy:

$$\pi(\theta) \propto g(\theta)^\alpha \exp\{\beta\nu(\theta)\}.$$

Note: In general, distributions that belong to the exponential family have conjugate prior distributions. This is because conjugacy involve manipulation of sufficient statistics.

Exercise 3.9. Let $y_1, \dots, y_N \sim N(\mu, \sigma^2)$, where σ^2 is known. Derive an invariant prior distribution (Jeffrey's prior distribution) for μ .

Exercise 3.10. Consider Example @(exm:exponential) featuring the exponential distribution.

1. Construct an invariant prior distribution for this distribution.
2. Derive the posterior distribution using the invariant prior distribution.
3. Using integration, discuss the validity of the prior distribution.

Chapter 4

Sampling

4.1 Uniform random numbers

What we won't be doing in this module is generating true uniform random numbers. This is incredibly difficult and usually requires lots of expensive hardware. This is because computers aren't good at being random, they require algorithmic instructions. True random number generation often uses physical methods, such as the radioactive decay of atoms, or atmospheric noise.

Throughout this module, we will be using R's built in random number generation. This is a pseudo random number generator, that has excellent random properties, but will eventually repeat. A basic random number generation method is sampling from a uniform distribution on the unit interval

```
runif(1, 0, 1)
```

```
## [1] 0.08573886
```

4.2 Inverse transform sampling

Suppose we want to sample from a non-uniform one-dimensional distribution. The inverse transform theorem allows us to do this using the distribution's inverse function.

Definition 4.1. The **inverse function** of a distribution F , denoted F^{-1} , is defined for all $u \in (0, 1)$ by

$$F^{-1}(u) = \inf\{x \in \mathbb{R} : F(x) > u\}.$$

Theorem 4.1. Let $F : \mathbb{R} \rightarrow [0, 1]$, $U \sim U[0, 1]$ and $Y = F^{-1}(U)$. Then Y has distribution F .

Proof. We have

$$\pi(Y \leq a) = \pi(F^{-1}(U) \leq a) = \pi(\inf\{x \in \mathbb{R} : F(x) > u\} \leq a).$$

Since $\inf\{x \in \mathbb{R} : F(x) > u\} \leq a$ can only hold if $F(a) \geq U$, we have

$$\pi(Y \leq a) = \pi(F(a) \geq U)$$

As $U \sim U[0, 1]$, we have $\pi(F(a) \geq U) = F(a)$. □

This theorem says that if we have a random variable $U \sim U[0, 1]$ and we want to get $Y \sim F$, then we can use $F^{-1}(U)$. Viewing this theorem graphically can provide a much more intuitive understanding.

Example 4.1. We would like to sample from an exponential distribution with rate 0.5. The density function is given by

$$\pi(y \mid \lambda = 0.5) = 0.5e^{0.5y}.$$

The distribution function can be derived by

$$\begin{aligned} F(y \mid \lambda = 0.5) &= \int_0^y 0.5e^{0.5t} dt \\ &= 1 - e^{0.5y}. \end{aligned}$$

Finally, the inverse function is given by

$$F^{-1}(y \mid \lambda = 0.5) = -2\log(1 - y).$$

Therefore, if $U \sim U[0, 1]$, $-2\log(1 - U) \sim \text{Exp}(0.5)$.

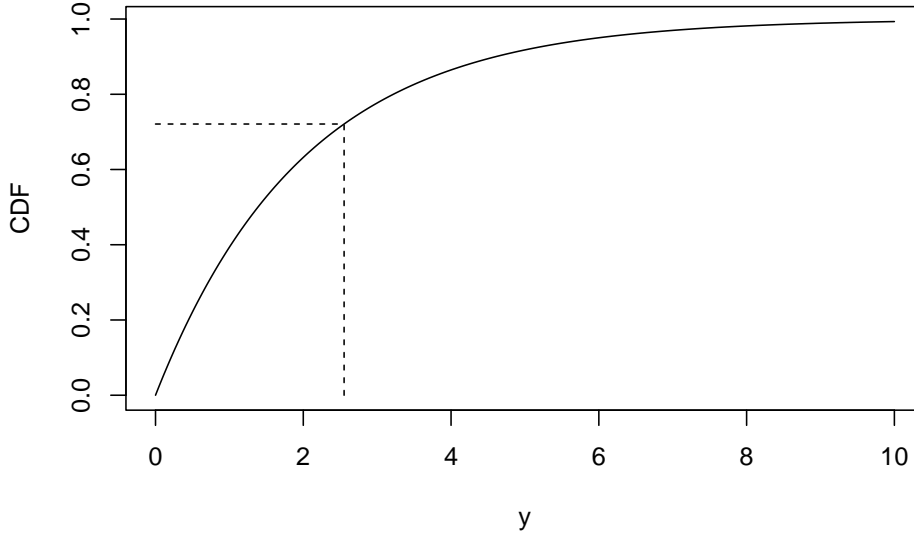
The R code below generates a plot to show this. We can plot the CDF for most one parameter distributions straightforwardly. We can think of this theorem as allowing us to sample a point on the y-axis and then computing the quantile this corresponds to.

```
set.seed(12345) # to reproduce
y <- seq(0, 10, 0.01) #Show on the interval [0, 5]
f <- 1 - exp(-0.5*y) #Construct the cumulative density function (CDF)
plot(y, f, type='l', xlab="y", ylab="CDF")

#Sample u
u <- runif(1)

#Get the corresponding y value
f.inv <- -2*log(1-u)

#plot
segments(x0 = 0, y0 = u, x1 = f.inv, y1 = u, lty = 2)
segments(x0 = f.inv, y0 = 0, x1 = f.inv, y1 = u, lty = 2)
```



4.3 Rejection sampling

We now have a way of sampling methods where we can analytically derive the inverse distribution function. We can use this to sample from more complex densities, or simple densities more efficiently. Rejection sampling by sampling according to a density we can sample from and then rejecting or accepting that sample based on the density we're actually interested in.

Suppose we want to sample from a density p , but can only generate samples from a density q . If there exists some constant $c > 0$, such that $\frac{\pi(y)}{q(y)} \leq c$ for all y , then we can generate samples from p by

1. Sampling $Y \sim Q$
2. Sampling $U \sim U[0, 1]$
3. Computing $k = \frac{\pi(u)}{cq(y)}$
4. Accepting y if $U < k$ and rejecting otherwise.

This says draw sample a point y according to the density q . Draw a vertical line at y from the x-axis to $cq(y)$. Sample uniformly on this line. If the uniformly random sample is below q , then accept it. Otherwise, reject it. The theory behind this is as follows. Suppose we sample some point y according to this algorithm and we want to work out its density f , then

$$f(y) \propto q(y)\pi(U < k) = q(y)\frac{\pi(u)}{cq(y)} = \frac{\pi(u)}{c}.$$

Therefore, $f = p$.

Example 4.2. Suppose we want to sample from a distribution that has the density

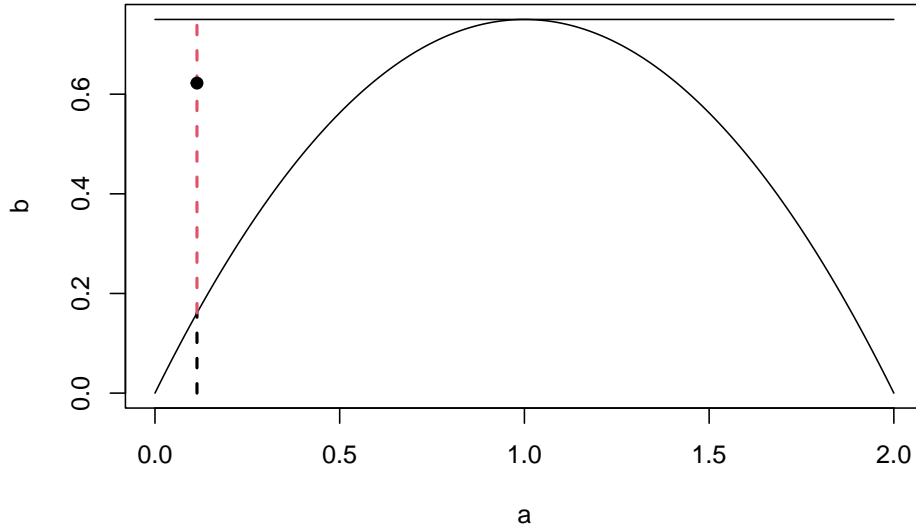
$$\pi(y) = \begin{cases} \frac{3}{4}y(2-y), & y \in [0, 2] \\ 0, & \text{otherwise} \end{cases}.$$

This has a maximum at $\frac{3}{4}$. We choose $p \sim U[0, 1]$ and $c = \frac{3}{4}$. The R code below shows a pictorial version of how one sample is generated.

```
set.seed(1234)  #to reproduce
M <- 3/4        #set M
y <- runif(1)   #sample Y ~ Q
p <- 3/4*y*(2-y) #compute pi(y)
k <- p/(M*1)    #compute k
u <- runif(1)   #sample U ~ U[0, 1]
ifelse(u < k, 'accept', 'reject') #Accept if u < k

## [1] "reject"

#Create nice plot
a <- seq(0, 2, 0.01)
b <- 3/4*a*(2-a)
c <- M*rep(1, length(a))
plot(a, b, ylim = c(0, M), type = 'l')
lines(a, c)
segments(x0 = y, y0 = 0, x1 = y, y1 = 3/4*y*(2-y), lty = 2, lwd = 2)
segments(x0 = y, y0 = 3/4*y*(2-y), x1 = y, y1 = M, lty = 2, col = 2, lwd = 2)
points(x = y, y = u, pch = 19)
```



The plot also shows how the choices of M and q can make the sampling more or less efficient. In our example, the rejection space is large, meaning many of our proposed samples will be rejected. Here, we could have chosen a better q

to minimise this space.

4.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a set of algorithms that sample from a posterior distribution. They combine the idea of rejection sampling, with the theory of Markov chains. Before we set out the theory of Markov chains, we'll go through an example to show how MCMC works.

Example 4.3. (Adapted from Statistical Rethinking §9) Consider an eccentric King whose kingdom consists of a ring of 10 islands. Directly north is island one, the smallest island. Going clockwise around the archipelago, next is island two, which is twice the size of island one, then island three, which is three times as large as island one. Finally, island 10 is next to island one and ten times as large.

The King wanted to visit all of his islands, but spending time on each one according to its size. That is he should spend the most time on island ten and the least on island one. Being climate conscious, he also decided that flying from one side of the archipelago to the other was not allowed. Instead, he would only sail from one island to either of its neighbors. So from island one, he could reach islands two and ten.

He decided to travel according to these rules:

1. At the end of each week, he decides to stay on the same island or move to a neighboring island according to a coin toss. If it's heads he proposes moving clockwise, and tails anti-clockwise. The island he is considering moving to is called the proposal island.
2. To decide if he is going to move to the proposal island, the King counts out a number of shells equal to the number of size of the island. So if island five is the proposal island, he counts out five shells. He then counts out a number of stones equal to the size of the current island.
3. If the number of seashells is greater than the number of stones, he moves to the proposed island. If the number of seashells is less than the number of stones, he takes a different strategy. He discards the number of stones equal to the number of seashells. So if there are six stones and five seashells, he ends up with $6-5=1$ stone. He then places the stones and seashells into a bag and chooses one at random. If he picks a seashell, he moves to the proposed island, if he picks a shell, he stays put.

This is a complex way of moving around, but it produces the required result; the time he spends on each island is proportionate to the size of the island. The code below shows an example of this over 10,000 weeks.

```
weeks <- 10000
island <- numeric(weeks)
```

```

current <- 10
for(i in 1:weeks){

  ## record current position
  island[i] <- current

  #Flip a coin to move to a propose a new island
  proposed <- current + sample(c(1, -1), size = 1)

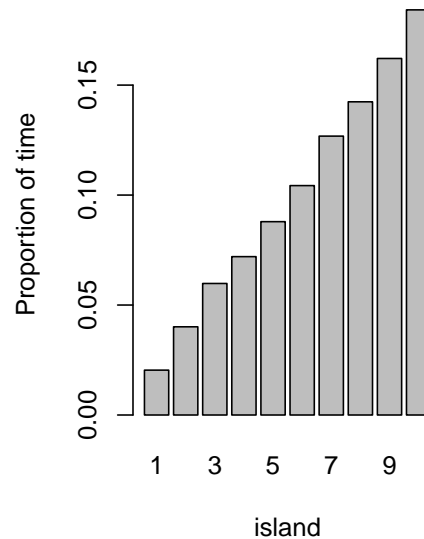
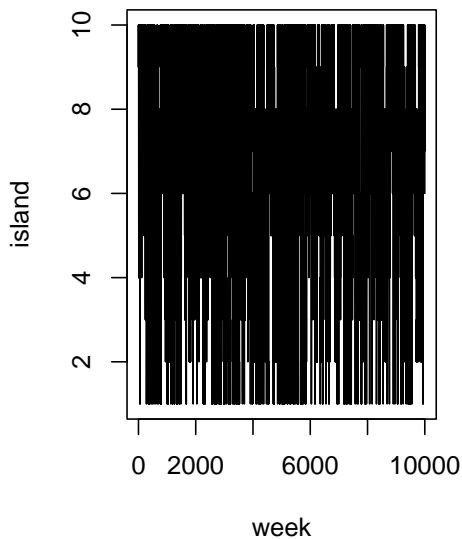
  #Ensure he loops round the island
  if(proposed < 1)
    proposed <- 10
  if(proposed > 10)
    proposed <- 1

  #Decide to move
  p <- proposed/current
  u <- runif(1)
  if(u < p)
    current <- proposed

}

#Plot results
par(mfrow = c(1, 2))
plot(island, type = 'l', xlab = "week", ylab = "island")
barplot(table(island)/weeks, xlab = "island", ylab = "Proportion of time")

```



We can recognise several different statistical principles in this example. The King decides to move islands dependent on where he is currently, not based on where he has been previously (Markov principle). He proposes an island to move to and accepts or rejects this decision based on some distribution (rejection principle). We are now going to describe some of the properties of Markov chains, including the Markov principle.

4.5 Properties of Markov Chains

Definition 4.2. A sequence of random variables $\{Y_1, Y_2, \dots\}$ is a **Markov chain** if $p(Y_{n+1} | Y_n, \dots, Y_1) = p(Y_{n+1} | Y_n)$. That is that distribution of the next state Y_{n+1} only depends on the current state Y_n and not any previous states.

Definition 4.3. The probability of transitioning from state i to state j in a Markov chain is given by p_{ij} . The **transition matrix** for a Markov chain with N states is the $N \times N$ matrix $P = p_{ij}$, where the $\{i, j\}^{th}$ entry is probability is moving from state i to state j .

These two properties make Markov chains nice to work with, especially the Markov property (Definition ??). Two other important definitions are

Definition 4.4. The **period** of a state i is given by $d_i = \gcd\{n > 0; p_{ii} > 0\}$. A state is **aperiodic** if $d_i = 1$. An **aperiodic chain** is a chain where all states are a periodic.

Definition 4.5. A Markov chain is **irreducible** if there exists an $n \in \mathbb{N}$ such that $p(Y_n = i | Y_0 = j)$ for all pairs i and j . In other words, it is possible to move from any state to any other state in a finite number of steps.

We can use these definitions to start working with distributions. Suppose, the state we start at is drawn from some distribution $Y_1 \sim q$. Then the distributions of the second state Y_2 depends on the distribution of Y_1 and the transition probabilities

$$p(Y_2 = j) = \sum_i q_i p_{ij}.$$

If we denote the distribution of $Y_2 \sim q^{(2)}$, then we can write it in terms of the transition matrix $q' = qP$. Now suppose we would like the distribution of $Y_3 \sim q^{(3)}$, thanks to the Markov property, this is the distribution for Y_2 multiplied by the transition matrix, so $Y_3 \sim qP^2$. Inductively, $P_k \sim qP^{k-1}$. To use Markov chains to sample from distributions, we need to identify the Eigenvalues of the transition matrix.

Proposition 4.1. A transition matrix P always has at least one eigenvalue equal to one.

Proof. The columns of P sum to 1 as they are probability distributions. Therefore, $\mathbf{1}$ is an eigenvector. \square

Definition 4.6. If a transition matrix P has a unique Eigenvalue that takes the value 1, there is a unique distribution π such that

$$\pi P = \pi.$$

This distribution π , is known as the **stationary distribution**.

This important concept underpins MCMC methods. It says that no matter where we start our chain, we'll eventually end up sampling states according to the distribution π . It make take a long time to reach the stationary distribution, but it will eventually get there.

In order to check whether our Markov chain will converge to a stationary distributions, we need to check:

1. the Markov chain is aperiodic,
2. the Markov chain is irreducible, and
3. that there exists a unique distribution π such that $\pi P = \pi$.

Example 4.4. In Example ??, the King wanted to visit the islands according to how large they are. We can think of the islands as the states and the stationary distribution as $p(Y = i) \propto i$. The eccentric method the King used allowed him to construct a transition matrix for an aperiodic Markov chain. He also never visited islands regularly using this method.

When designing a Markov chain, it is usually straightforward to design one that meets conditions one and two. Condition three is more difficult to prove, but for some chains it is possible to show they satisfy detailed balance.

Definition 4.7. The Markov chain P satisfies **detailed balance** with respect to the distribution π if

$$\pi_i p_{ij} = \pi_j p_{ji}.$$

Theorem 4.2 (Detailed Balance). *Let P be a transition matrix that satisfies detailed balance with respect to the distribution π . Then $\pi P = \pi$.*

Proof. The j^{th} row of πP is

$$\begin{aligned} \sum_i \pi_i p_{ij} &= \sum_i \pi_j p_{ji} \quad (\text{detailed balance}) \\ &= \pi_j \sum_i p_{ji} \\ &= \pi_j. \quad (\text{probaility sums to 1}) \end{aligned}$$

Hence $\pi P = \pi$. □

The section has shown us that we can use a Markov chain to simulate from a distribution π . All we need is for the Markov chain to be irreducible, aperiodic,

and for the transition matrix to satisfy $\pi P = \pi$. This provides the foundation theory for MCMC and allows us to sample from a posterior distribution π . What it doesn't tell us is how to design the Markov chain, and that is what the next sections deal with.

4.6 Gibbs Sampler

We're now going to look at two algorithms to draw samples from posterior distributions. The first is the Gibbs sampler.

When we can sample directly from conditional posterior distributions, we can use a Gibbs sampler. Suppose we have a distribution with parameters $\{\theta_1, \dots, \theta_N\}$, a Gibbs sampler works as follows:

1. Set initial values $\{\theta_1^{(0)}, \dots, \theta_N^{(0)}\}$
2. Set $i = 1$.
3. Draw a value for $\theta_1^{(i)}$ from $\pi(\theta_1 | \theta_2^{(i-1)}, \dots, \theta_N^{(i-1)})$.
4. Draw a value for $\theta_2^{(i)}$ from $\pi(\theta_2 | \theta_1^{(i-1)}, \theta_3^{(i-1)}, \dots, \theta_N^{(i-1)})$.
5. Repeat steps 3 and 4 for parameters $\{\theta_3^{(i)}, \dots, \theta_N^{(i)}\}$.
6. Repeat steps 3, 4, and 5, for $i = 2, \dots, M$.

In code, this might look like

```
M #number of iterations
N #number of parameters
theta.store <- matrix(NA, N, M)
theta <- numeric(N)

for(j in 1:M){
  for(j in 1:N){
    theta[i] <- #sample from conditional with theta[-i]
  }
  theta.store[, j] <- theta.current #store current values
}
```

The sequence $\{\theta_0^{(0)}, \dots, \theta_N^{(0)}\}, \{\theta_0^{(1)}, \dots, \theta_N^{(1)}\}, \dots, \{\theta_0^{(M)}, \dots, \theta_N^{(M)}\}$ form a Markov chain. They also form a series of samples from the posterior distribution. However, they are not iid.

Example 4.5. Recall the lab in the previous chapter, where we derived the posterior distributions for the normal distribution with unknown mean and variance. The posterior distributions are given by

$$\mu | y, \sigma^2 \sim N(\mu_1, \sigma_1^2)$$

where $\mu_1 = \left(\frac{\sum_{i=1}^{30} y_i}{\sigma^2} + \frac{\mu_0}{\sigma_0^2} \right)$ and $\sigma_1^2 = \left(\frac{30}{\sigma^2} + \frac{1}{\sigma_0^2} \right)^{-1}$, and

$$\sigma^2 \mid y, \mu \sim \text{inv-Gamma} \left(\alpha + \frac{N}{2}, \beta + \frac{\sum_{i=1}^N (y_i - \mu)^2}{2} \right)$$

In this example, we can use a Gibbs sampler to sample from the posterior distribution. The algorithm is given by

1. Set $\mu^{(0)} = 0$ and $\sigma^{2(0)}$
2. Set $i = 1$.
3. Sample $\mu^{(i)} \mid y, \sigma^{2(i-1)} \sim N(\mu_1^{(i-1)}, \sigma_1^{2(i-1)})$
4. Sample $\sigma^{2(i)} \mid y, \mu^{(i)} \sim \text{inv-Gamma} \left(\alpha + \frac{N}{2}, \beta + \frac{\sum_{i=1}^N (y_i - \mu^{(i)})^2}{2} \right)$
5. Repeat steps 3 and 4 for $i = 2, \dots, 1000$.

4.7 Metropolis-Hastings

The second algorithm we are going to look at is the Metropolis-Hastings algorithm. This is a useful algorithm if we cannot sample directly from the posterior distribution and if the conditional distributions do not have a closed form.

Suppose we want to sample from the posterior distribution $\pi(\theta \mid y)$. The Metropolis-Hastings works as follows:

1. Set the initial value $\theta^{(0)}$.
2. Set $i = 1$.
3. Propose a new value of θ' from some distribution q
4. Accept θ' with probability

$$p_{\text{acc}} = \min \left\{ \frac{\pi(\theta' \mid y) q(\theta \mid \theta')}{\pi(\theta \mid y) q(\theta' \mid \theta)}, 1 \right\}.$$

5. Repeat steps 3 to 4 for $i = 2, \dots, M$.

There are two parts to the acceptance probability in step 4. The first is the posterior ratio, similar to saying the likelihood of θ' given the observed data over the likelihood of θ given the data. The second is the proposal ratio. It is similar to saying the likelihood of proposing θ given the current value θ' , over the likelihood of proposing θ' given the current value θ .

In practice, we don't need to evaluate the full posterior distribution. Recall

$$\pi(\theta \mid y) = \frac{\pi(y \mid \theta) \pi(\theta)}{\pi(y)}$$

As the the denominator doesn't depend on θ , it cancels in the ration. The ratio becomes

$$\frac{\pi(\theta' | y)}{\pi(\theta | y)} = \frac{\pi(y | \theta')\pi(\theta')}{\pi(y | \theta)\pi(\theta)}.$$

This is the likelihood ratio multiplied by the prior ratio.

Proposition 4.2. *The Markov chain generated by the Metropolis-Hastings algorithm satisfies detailed balance with respect to the posterior distribution.*

Proof. Denote the current state θ and the proposed state θ' . We would like to show

$$\pi(\theta | y)\pi(\theta' | \theta) = \pi(\theta' | y)\pi(\theta | \theta').$$

The density of θ' given the proposed state θ is the proposal density multiplied by the acceptance probability. It is given by

$$\begin{aligned}\pi(\theta' | \theta) &= q(\theta' | \theta)p_{acc} \\ &= q(\theta' | \theta) \min \left\{ \frac{\pi(\theta' | y) q(\theta | \theta')}{\pi(\theta' | y) q(\theta' | \theta)}, 1 \right\} \\ &= \min \left\{ \frac{\pi(\theta' | y)}{\pi(\theta' | y)} q(\theta | \theta'), q(\theta' | \theta) \right\}.\end{aligned}$$

The left hand side of the detailed balance equation becomes

$$\pi(\theta | y)\pi(\theta' | \theta) = \min\{\pi(\theta' | y)q(\theta | \theta'), \pi(\theta | y)q(\theta' | \theta)\}.$$

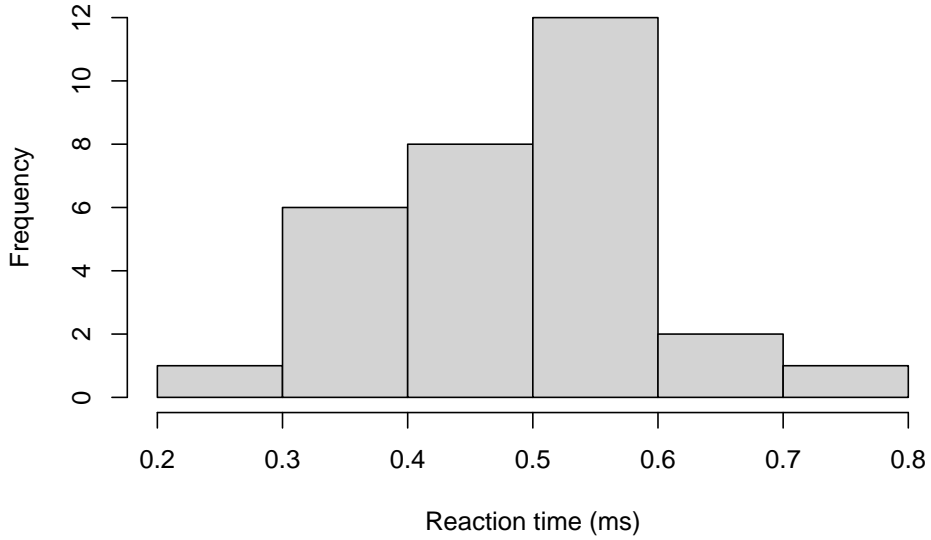
Analogously, we can show the right hand side is

$$\pi(\theta' | y)\pi(\theta | \theta') = \min\{\pi(\theta' | y)q(\theta | \theta'), \pi(\theta | y)q(\theta' | \theta)\}.$$

Hence, $\pi(\theta | y)\pi(\theta' | \theta) = \pi(\theta' | y)\pi(\theta | \theta')$ and the Markov chain satisfies detailed balance with respect to the posterior disquisition. \square

Example 4.6. Lets think again about the reaction time example in the previous chapter. The time until each lorry driver reacts (in milliseconds) is

```
y <- c(0.34, 0.47, 0.58, 0.27, 0.74, 0.44, 0.46, 0.65, 0.36, 0.55, 0.58, 0.55, 0.53, 0.56, 0.54,
hist(y, main = "", xlab = "Reaction time (ms)"))
```



Assuming $y_i \sim N(\mu, \sigma^2)$, by Bayes' theorem, the posterior distribution is

$$\pi(\mu \mid y, \sigma^2) \propto \pi(y \mid \mu, \sigma^2)\pi(\mu).$$

One of the issues, was the we assigned a normal prior distribution to the population mean parameter μ . The advantage was that we could derive a posterior distribution with closed form. The disadvantage was the prior distribution assigned positive probability to impossible values of μ .

Now we have a tool to sample from posterior distributions that don't have a closed form. We can instead assign an exponential prior distribution, a distribution which only has non-negative support. Letting $\mu \sim \text{Exp}(0.01)$ sets a vague prior distribution on μ . The posterior distribution is therefore

$$\pi(\mu \mid y, \sigma^2) \propto \exp \left\{ -0.01\mu - \sum_{i=1}^{30} \frac{(y_i - \mu)^2}{\sigma^2} \right\}$$

We can use the Metropolis-Hasting algorithm to sample from this posterior distribution. But how should we propose new value of μ ? A common method is a Metropolis-Hastings Random Walk proposal distribution. The proposal distribution is symmetric and centered on μ . The two most common methods are $\mu' \mid \mu \sim U[\mu - \varepsilon, \mu + \varepsilon]$ and $\mu' \mid \mu \sim N(\mu, \tau^2)$. We choose the uniform proposal distribution, with

$$q(\mu' \mid \mu) = \frac{1}{2\varepsilon}.$$

The acceptance probability is therefore

$$p_{\text{acc}} = \min \left\{ \frac{\exp \left\{ -0.01\mu' - \sum_{i=1}^{30} \frac{(y_i - \mu')^2}{\sigma^2} \right\}}{\exp \left\{ -0.01\mu - \sum_{i=1}^{30} \frac{(y_i - \mu)^2}{\sigma^2} \right\}}, 1 \right\}$$

We can implement a sampler in R as follows:

```
#Set up elements for MCMC
set.seed(123) #to reproduce
n.iter  <- 10000
mu.store <- numeric(n.iter)

#Initial values
mu <- 1
sigma <- 0.1 #known

for(i in 1:n.iter){

  #Propose value for mu
  mu.proposed <- runif(1, mu - 0.01, mu + 0.01)

  if(mu.proposed > 0){ #If mu < 0 we can reject straight away

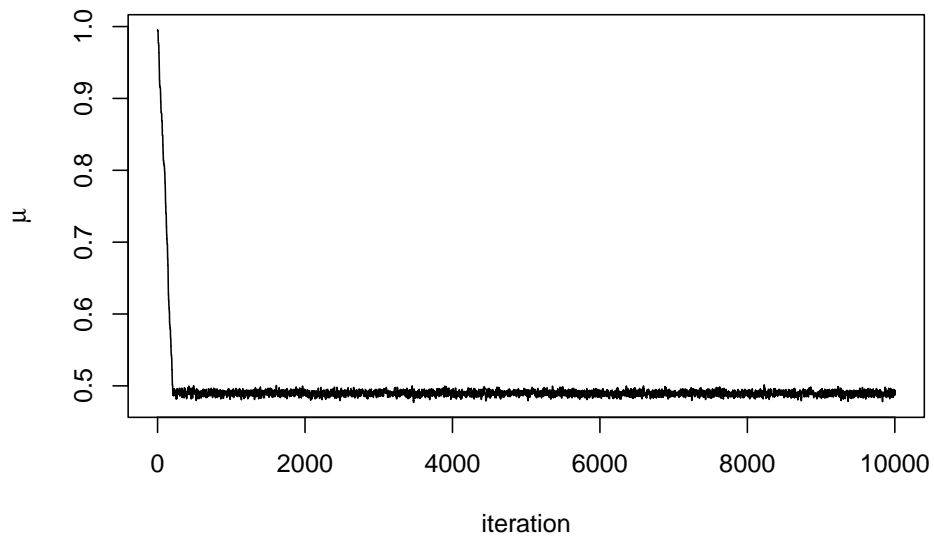
    #Compute (log) acceptance probability
    log.numerator  <- -0.01*mu.proposed - sum(y - mu.proposed)^2/(2*sigma^2)
    log.denominator <- -0.01*mu - sum(y - mu)^2/(2*sigma^2)

    log.p.acc <- log.numerator - log.denominator
    u <- runif(1)

    #Accept/Reject step
    if(log(u) < log.p.acc){
      mu <- mu.proposed
    }
  }

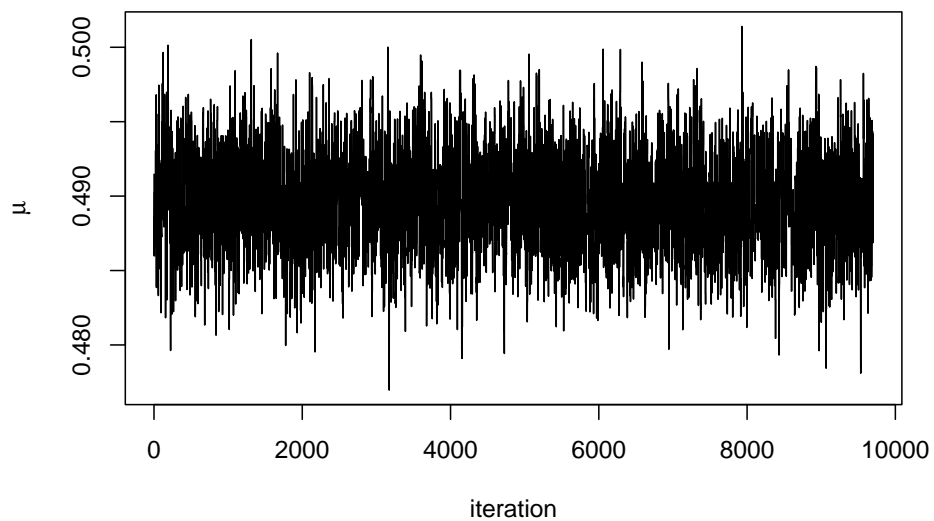
  #Store mu at each iteration
  mu.store[i] <- mu
}

plot(mu.store, type = 'l', xlab = "iteration", ylab = expression(mu))
```

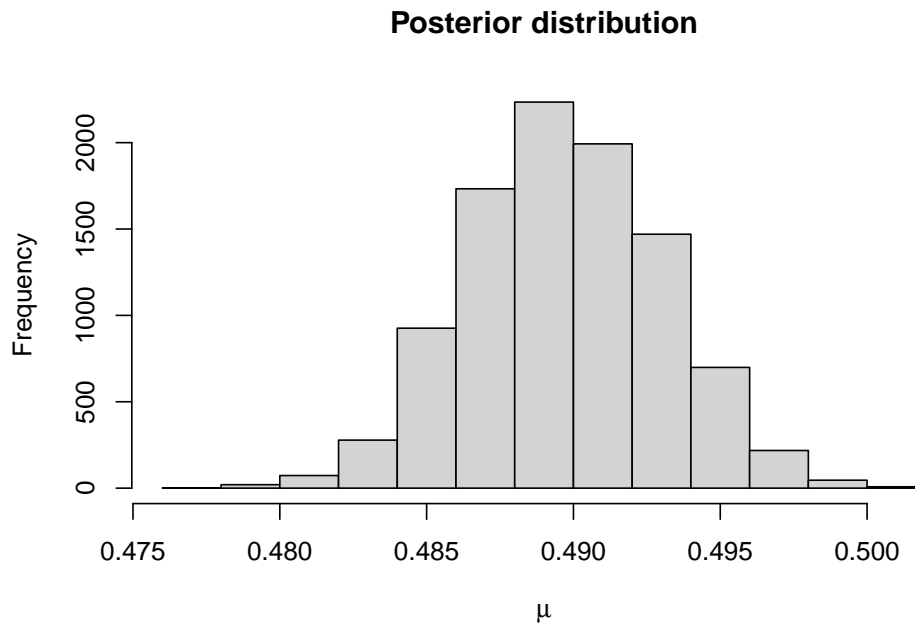


We can see that after about 300 iterations, the Markov chain has converged to its stationary distribution, the posterior distribution. We can see this more clearly by removing the first 300 iterations.

```
plot(mu.store[-c(1:300)], type = 'l', xlab = "iteration", ylab = expression(mu))
```



```
hist(mu.store[-c(1:300)], xlab = expression(mu), main = "Posterior distribution")
```



The 95% credible interval for μ using this prior distribution is

```
quantile(mu.store[-c(1:300)], c(0.025, 0.975))
```

```
##      2.5%      97.5%
## 0.4832146 0.4961484
```

Recall that using the normal prior distribution, it was

```
0.486 0.493
```

It seems that the posterior distribution is very similar when using these two prior distributions. This is because the data is very informative.

4.8 MCMC Diagnostics

When running an MCMC algorithm, it is always important to check that the Markov chain has converged and is mixing well.

A key way of doing this is by looking at the trace plot. This is the plot out MCMC iterations over time. The trace plot should look like it has converged to the stationary distribution and exploring the stationary distribution efficiently. What it shouldn't look like is a long series of small steps, or being stuck in one spot for a long time. There are two definitions that help us isolate an efficient Markov chain.

Definition 4.8. The **burn-in period** is the number of iterations the Markov chain takes to reach the stationary distribution.

Definition 4.9. The **thinning parameter** is the period of iterations of the Markov chain that are stored.

Example 4.7. In Example @{exm:norm}, we saw a Markov chain that mixes well. We took the burn-in period to be 3,000 iterations, which was how long it took to for the chain to converge.

In a Metropolis-Hasting random walk algorithm, the proposal distribution often has a large impact on how well the Markov chain mixes. The variance, or step size, of the proposal distribution can be tuned to ensure the chain mixes well.

The following two examples show poorly mixing Markov chains. The first is where the step size is too big and the chain frequently gets stuck for several hundred iterations.

```
set.seed(123) #to reproduce
n.iter  <- 10000
mu.store <- numeric(n.iter)

#Initial values
mu <- 1
sigma <- 0.1 #known

for(i in 1:n.iter){

  #Propose value for mu
  mu.proposed <- runif(1, mu - 0.1, mu + 0.1) #Step size too big

  if(mu.proposed > 0){ #If mu < 0 we can reject straight away

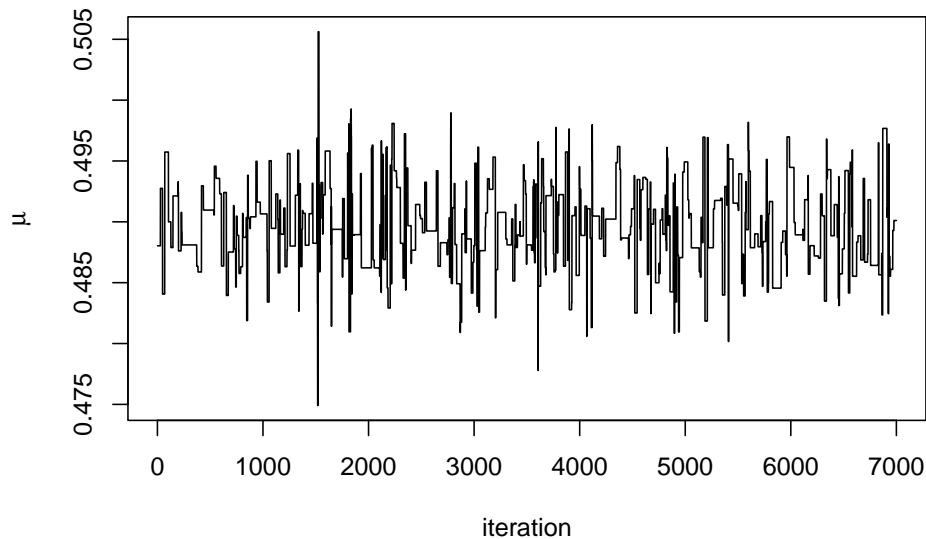
    #Compute (log) acceptance probability
    log.numerator  <- -0.01*mu.proposed - sum(y - mu.proposed)^2/(2*sigma^2)
    log.denominator <- -0.01*mu - sum(y - mu)^2/(2*sigma^2)

    log.p.acc <- log.numerator - log.denominator
    u <- runif(1)

    #Accept/Reject step
    if(log(u) < log.p.acc){
      mu <- mu.proposed
    }
  }

  #Store mu at each iteration
  mu.store[i] <- mu
}

plot(mu.store[-c(1:3000)], type = 'l', xlab = "iteration", ylab = expression(mu))
```



The next is where the step size is too small. It takes a long time for the chain to converge (~50% of the run time). When the chain does converge, it is inefficient at exploring the space.

```
set.seed(123) #to reproduce
n.iter  <- 10000
mu.store <- numeric(n.iter)

#Initial values
mu <- 1
sigma <- 0.1 #known

for(i in 1:n.iter){

  #Propose value for mu
  mu.proposed <- runif(1, mu - 0.0005, mu + 0.0005) #Step size too small

  if(mu.proposed > 0){ #If mu < 0 we can reject straight away

    #Compute (log) acceptance probability
    log.numerator  <- -0.01*mu.proposed - sum(y - mu.proposed)^2/(2*sigma^2)
    log.denominator <- -0.01*mu - sum(y - mu)^2/(2*sigma^2)

    log.p.acc <- log.numerator - log.denominator
    u <- runif(1)

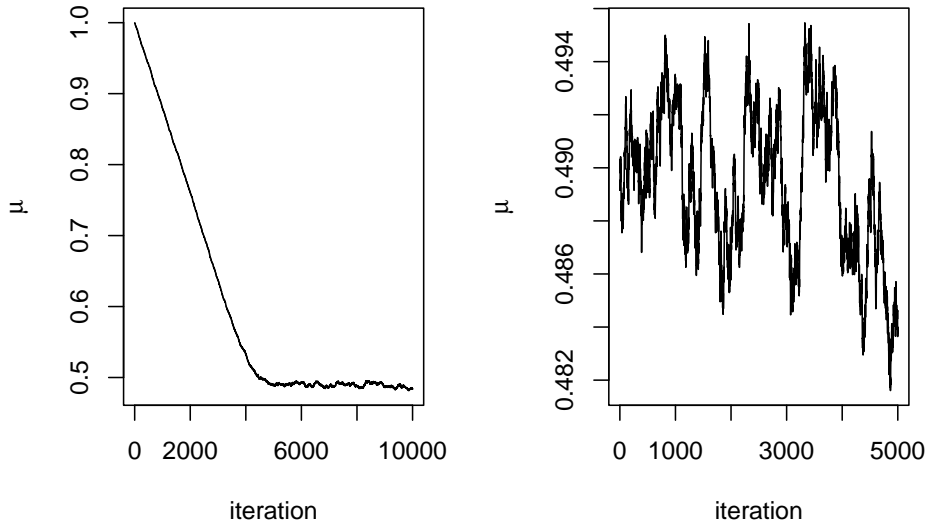
    #Accept/Reject step
    if(log(u) < log.p.acc){
      mu <- mu.proposed
    }
  }
}
```

```

    }
  }

  #Store mu at each iteration
  mu.store[i] <- mu
}
par(mfrow = c(1, 2))
plot(mu.store, type = 'l', xlab = "iteration", ylab = expression(mu))
plot(mu.store[-c(1:5000)], type = 'l', xlab = "iteration", ylab = expression(mu))

```



4.9 Exercises

Exercise 4.1. Let Y have the density function

$$\pi(y) = \alpha y^3 \quad \text{for } 0 \leq y \leq 4.$$

1. Compute the value of α that makes this a proper distribution.
2. Derive the distribution function $F(y)$.
3. Using the inverse transform theorem, derive a function g such that if $U \sim U[0, 1]$, then $g(U)$ has the same distribution as Y .

Solution. This exercise is about the inverse transform sampling method.

1. As $\int_0^4 \pi(y) dy = 1$, we have $\alpha = \frac{1}{64}$.
2. Introducing a dummy variable x , we have

$$\int_0^y \frac{1}{64} x^3 dx = \frac{1}{256} y^4.$$

The distribution function for Y is given by

$$F(y) = \begin{cases} 0 & \text{for } y < 0 \\ \frac{1}{256}y^4 & \text{for } 0 \leq y \leq 4 \\ 1 & \end{cases}$$

3. We have

$$F(y) = \frac{1}{256}y^4 \iff F^{-1}(y) = 4y^{\frac{1}{4}}.$$

By the inverse transform theorem, if $U \sim U[0, 1]$ then $4U^{\frac{1}{4}}$ has the same distribution as Y .

Exercise 4.2. Suppose Y has the density function

$$f(y) = \frac{1}{2\sqrt{y}}e^{-\sqrt{y}} \quad \text{for } y \geq 0.$$

1. Using integrating by substitution, derive the distribution function $F(y)$.
2. Using the inverse transform method, construct a method for sampling from this distribution.

Solution. Another exercise about the inverse transform method, but this time with a slightly more complex density function.

1. Introducing the dummy variable x , we need to evaluate

$$\int_0^y \frac{1}{2\sqrt{x}}e^{-\sqrt{x}} dx.$$

The easiest substitution to use is $u = \sqrt{x}$. This gives us $dx = 2du\sqrt{x}$. Substituting these into the integral gives A quicker, but less obvious way to compute the integral is to note that $\frac{d}{dx}e^{f(x)} = f'(x)e^{f(x)}$. The distribution function is therefore

$$F(y) = \begin{cases} 1 - e^{-\sqrt{y}} & \text{for } y \geq 0 \\ 0 & \text{otherwise} \end{cases}.$$

2. Using the inverse transform method, if $U \sim U[0, 1]$ then $F^{-1}(U)$ will have the same distribution as Y . The inverse of the distribution function is

$$F^{-1}(y) = (\log(1 - y))^2.$$

The method would be A. Sample $U \sim U[0, 1]$ B. Compute $(\log(1 - U))^2$

Exercise 4.3. The density function for the half-normal distribution with variance 1 is

$$\pi(x) = \frac{2}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\} \quad \text{for } x \geq 0.$$

1. Using an exponential distribution with rate λ as a proposal distribution, show that

$$\frac{\pi(x)}{q(x)} = \frac{2}{\lambda\sqrt{2\pi}} \exp\left\{\lambda x - \frac{1}{2}x^2\right\}.$$

2. What is M , the maximum value of this ratio? On the same plot, sketch $\pi(x)$ and $Mq(x)$.
3. Construct a rejection sampling algorithm to sample from the half normal distribution.

Solution. The normal distribution is difficult to sample from, especially as the inverse of its distribution is difficult to work with. One method to generate samples is using a rejection sampling method.

1. The density function of an exponential distribution with rate λ is $q(x | \lambda) = \lambda e^{-\lambda x}$. The ratio is given by

$$\begin{aligned} \frac{\pi(x)}{q(x)} &= \frac{\frac{2}{\sqrt{2\pi}} \exp\{-\frac{1}{2}x^2\}}{\lambda e^{-\lambda x}} \\ &= \frac{2}{\lambda\sqrt{2\pi}} \exp\left\{\lambda x - \frac{1}{2}x^2\right\}. \end{aligned}$$

2. To find the maximum value of this ratio, we first need to find the value of x that maximises the ratio. Differentiating gives

$$\frac{d}{dx} \frac{2}{\lambda\sqrt{2\pi}} \exp\left\{\lambda x - \frac{1}{2}x^2\right\} = \frac{2(\lambda - x)}{\lambda\sqrt{2\pi}} \exp\left\{\lambda x - \frac{1}{2}x^2\right\}$$

Setting this equal to 0 shows that $x = \lambda$ maximises the ratio. The maximum value of the ratio is thus

$$M = \frac{\pi(\lambda)}{q(\lambda)} = \frac{2}{\lambda\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\lambda^2\right\}$$

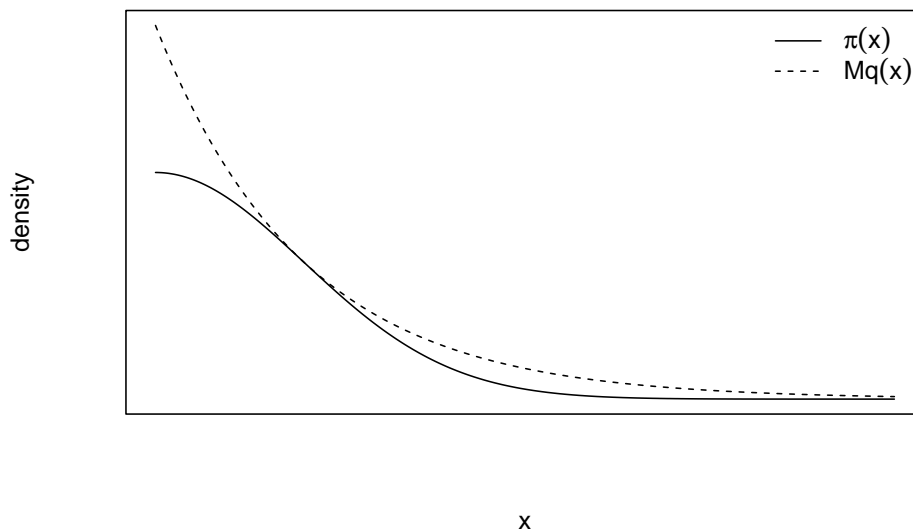
The plot below shows that not only is $\pi(x) \leq Mq(x)$, but is that M is the optimal value for a rejection sampling algorithm.

```
x <- seq(0, 5, 0.01)
lambda <- 1

#Densities
pi.x <- 2/(sqrt(2*pi))*exp(-0.5*x^2)
q.x <- dexp(x, lambda)

#M
M <- 2/(lambda*sqrt(2*pi))*exp(0.5*lambda^2)

#Plot
plot(x, pi.x, type = 'l', ylim = c(0, M), xaxt = "n", yaxt = "n", xlab = "x", ylab = "Density")
lines(x, M*q.x, lty = 2)
legend('topright', c(expression(pi(x)), expression(Mq(x))), lty = c(1, 2), bty = "n")
```

3. The rejection sampling algorithm for the half normal distribution works as follows: A. Sample x from the exponential distribution. B. Sample $u \sim U[0, 1]$. C. If $u < \frac{\pi(x)}{Mq(x)}$, then accept x as a sample from the half normal distribution.

Exercise 4.4. We are using a rejection sampling algorithm to sample from $f(y)$ using the proposal distribution $g(y)$. Given a proposed value y , and let A be the event of sample is accepted. The probability a sample is accepted, given it takes the value y is

$$A \mid y \sim \text{Bernoulli} \left(\frac{f(y)}{Mg(y)} \right).$$

1. Using the tower property, show $E(A) = \frac{1}{M}$.
2. Assuming each proposal is independent, what is the distribution of the number of samples generated until one is accepted?

Solution. We saw in the last question that we derived the optimal value of the constant M . This exercise is about computing the efficiency of a rejection sampling algorithm, which depends on M . The efficiency is the expected number of proposals generated until one is accepted.

1. Let A be the event a proposal is accepted, we would like We have $A \mid y \sim$

Bernoulli $\left(\frac{f(y)}{Mg(y)}\right)$. By the tower property

$$\begin{aligned} E(A) &= E(E(A \mid y)) \\ &= E\left(\frac{f(y)}{Mg(y)}\right) \\ &= \int \frac{f(y)}{Mg(y)} g(y) dy \\ &= \frac{1}{M} \int f(y) dy \\ &= \frac{1}{M}. \end{aligned}$$

2. The number of events until a success is modelled by the geometric distribution. The number of samples proposed before one is accepted is distributed geometrically with mean $\frac{1}{M}$. The smaller the value of M , the more efficient the sampler.

Exercise 4.5. Alice wants to construct a Markov Chain to sample numbers from one to six. She chooses an initial number at random. She flips a coin and decides to stay with that number if it's a head, and move to a different number if it's a tails. If she chooses to move number she flips the coin again. If it's heads, she moves to a number one larger than she is now, if it's tails she moves to a number one smaller than she is now. Once she reaches one or six, she stops.

1. Draw a state space diagram for this chain.
2. Is Alice's Markov chain aperiodic and irreducible?

Bob comes up with a similar Markov chain. Like Alice, he chooses an initial number at random and flips a coin to decide if he should keep that number or move to a different one. However, his next step differs from Alice's. If he chooses to move number he flips the coin again. If it's heads, he moves to a number two larger than he is now, if it's tails he moves to a number two smaller than he is now. He also treats the numbers like a circle, where if he is on six, he can move to four or two. Likewise, if he is on one, he can move to five or three.

1. Draw a state space diagram for this chain.
2. Is Bob's Markov chain aperiodic and irreducible?

Solution. The state space diagram for Alice and Bob's Markov chains are:

Neither Alice nor Bob have constructed Markov chains that are aperiodic and irreducible. In Alice's chain, states one and six are absorbing states – if Alice ends up with one of these numbers, she is never going to choose another number. This means states one and six have period one. As Alice's chain has absorbing states, it isn't irreducible either. For example, it isn't possible to get from state six to two in a finite (or any) number of steps. As Bob has introduced a mechanism to loop round the numbers, his chain is aperiodic. However, it isn't irreducible. Moving up and down two numbers means that he is restricted to

the even or odd numbers. If he starts on an even number, he is never going to reach an odd number.

Exercise 4.6. You observe N data points $\{y_1, \dots, y_N\} \sim N(\mu, \sigma^2)$. Place a normal prior distribution on the mean parameter μ and inverse-gamma prior distribution σ^2 to derive the posterior distributions $\pi(\mu, \sigma^2 | y)$.

Solution. Using the prior distributions

$$\mu \sim N(\mu_0, \sigma_0^2), \quad \sigma^2 \sim \text{inv} - \Gamma(\alpha, \beta)$$

gives the posterior distribution

The full conditional distributions are

$$\mu | \sigma^2, y \sim N(\mu_1, \sigma_1^2), \quad \sigma^2 | \mu, y \sim \text{inv} - \Gamma(\alpha_1, \beta_1).$$

A good MCMC algorithm could look like this:

1. Set $\mu^{(1)} = 0$, $\sigma^{2(0)} = 1$ and $i = 1$.
2. Sample $\mu^{(i)} | \sigma^{2(i-1)}, y \sim N(\mu_1, \sigma_1^2)$.
3. Sample $\sigma^{2(i)} | \mu^{(i)}, y \sim \text{inv} - \Gamma(\alpha_1, \beta_1)$.
4. Repeat steps 2 and 3 for a given number of iterations.
5. Check the trace plots to ensure the Markov chain has converged and mixed well.

Exercise 4.7. In the lectures, we looked at Metropolis-Hastings Random Walk algorithms. Another algorithm is a Metropolis-Hastings Independence Sampler. In an independence sampler, the proposal distribution does not depend on the current values, that is

$$q(y' | y) = q(y').$$

1. Let $y_1, \dots, y_N \sim N(\mu, \sigma^2)$, where σ^2 is known. Show the posterior distribution has no closed form when the prior distribution $\mu \sim \Gamma(\alpha, \beta)$ is used.
2. Suppose you are constructing an MCMC algorithm to sample from this distribution. Denote the current value of the Markov chain μ . Given a proposal $\mu' \sim N(0, 5^2)$ proposal distribution, derive the acceptance probability.

Solution. This question is about independence samplers. They are typically very good or very bad at sampling from the posterior distribution.

1. The likelihood function is given by

$$\pi(y | \mu, \sigma^2) \propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 \right\}.$$

The prior distribution is $\pi(\mu) \propto \mu^{\alpha-1} \exp\{-\beta\mu\}$. The posterior distribution is the product of these and given by

$$\pi(\mu | y, \sigma^2) \propto \mu^{\alpha-1} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 - \beta\mu \right\}.$$

2. The acceptance probability is

$$p_{acc} = \frac{\pi(\mu' | y, \sigma^2)}{\pi(\mu' | y, \sigma^2)} \frac{\pi(\mu)}{\pi(\mu')}.$$

Substituting the proportional densities in yields

$$p_{acc} = \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu')^2 + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mu)^2 - \frac{1}{50} \mu^2 + \frac{1}{50} \mu'^2 \right\}.$$

Chapter 5

Advanced computation

Now we have the tools of Bayesian inference and methods to sample from complex posterior distributions, we can start to look at more advanced methods and models. This chapter is split into two distinct parts, each showing a different method in Bayesian inference.

5.1 Data Augmentation

Real world data is often messy with data points missing, partially or completely unobserved. Two common ways to deal with are:

- Remove data points that are not completely observed. This throws away information and is likely to increase the overall uncertainty in the estimates.
- Replace data points that are not completely observed with estimates such as the sample mean. This is likely to underestimate the uncertainty as we are treating the observation as completely observed when it is not.

The Bayesian framework provides a natural way for dealing with missing, partially, or completely unobserved data. It allows us to treat the missing data points as random variables and infer the data points alongside the model parameters. This provides us with a method to quantify the uncertainty around our estimates of the missing data points.

In data augmentation, we distinguish between two likelihood functions.

Definition 5.1. The **observed data likelihood function** is the likelihood function of the observed data.

Definition 5.2. The **complete data likelihood function** is the likelihood function of the observed data and any missing or censored data had they been fully observed.

The difference between the two likelihood functions is that the complete data likelihood function is the functions had we observed everything we want to observe. However, as the complete data likelihood function contains data we didn't fully observe, we can't compute it. Instead we can only evaluate the observed data likelihood function.

In data augmentation, we start off with the observed data likelihood function and then augment this function by introducing variables that we want to have fully observed. This then gives us the complete data likelihood function.

5.1.1 Imputing censored observations

The first example we will look at is when data is censored. Instead of throwing away these observations, we will instead treat them as random variables and infer their values.

Example 5.1. A bank checks transactions for suspicious activities in batches of 1000. Denote the probability a transaction is suspicious by p and the number of suspicious transactions in a batch by Y .

The bank checks five batches and observes y_1, \dots, y_4 suspicious transactions in the first four batches. Due to a computer error, the number of suspicious transactions in the final batch is not properly recorded, but is known to be less than 6.

The observed data likelihood functions is

$$\pi(y_1, \dots, y_4, \tilde{y}_5 | p) = \left(\prod_{i=1}^4 \binom{1000}{y_i} p^{y_i} (1-p)^{1000-y_i} \right) \left(\sum_{j=1}^5 \binom{1000}{j} p^j (1-p)^{1000-j} \right).$$

Placing a uniform prior distribution on $p \sim U[0, 1]$ give the posterior distribution

$$\pi(p | y_1, \dots, y_4, \tilde{y}_5) = \left(\prod_{i=1}^4 \binom{1000}{y_i} p^{y_i} (1-p)^{1000-y_i} \right) \left(\sum_{j=1}^5 \binom{1000}{j} p^j (1-p)^{1000-j} \right).$$

Although we could sample from this distribution, it is not easy to work with. Instead, we can write down the complete data likelihood. This supposes we had observed y_5 and is

$$\pi(y_1, \dots, y_5 | p) = \prod_{i=1}^5 \binom{1000}{y_i} p^{y_i} (1-p)^{1000-y_i}$$

The posterior distribution is therefore

$$p | y_1, \dots, y_5 \sim \text{Beta} \left(\sum_{i=1}^5 y_i + 1, 1001 - \sum_{i=1}^5 y_i \right).$$

The full conditional distribution of y_5 given p , the other data points and $y_5 < 6$ is

$$\pi(y_5 = y \mid y_1, \dots, y_4, y_5 < 6, p) = \frac{\binom{1000}{y} p^y (1-p)^{1000-y}}{\sum_{j=1}^{1000} \binom{1000}{j} p^j (1-p)^j}, \quad y < 6$$

We can use a Gibbs sampler alternating between sampling p and y_5 .

5.1.2 Imputing latent variables

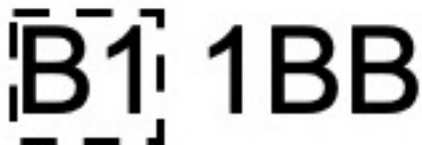
Often there are variables that cannot be observed, these may be hidden somehow or introduced to help with the modelling. Instead we can learn about this variable indirectly from the data.

A **latent variable** is a variable that cannot be observed.

A mixture model is an example of latent variables being useful.

Example 5.2. Royal Mail use image detection software to read postcodes on letters. A camera scans the front of an envelope and then records the barcode. This example is a very simplified version of how the system could work.

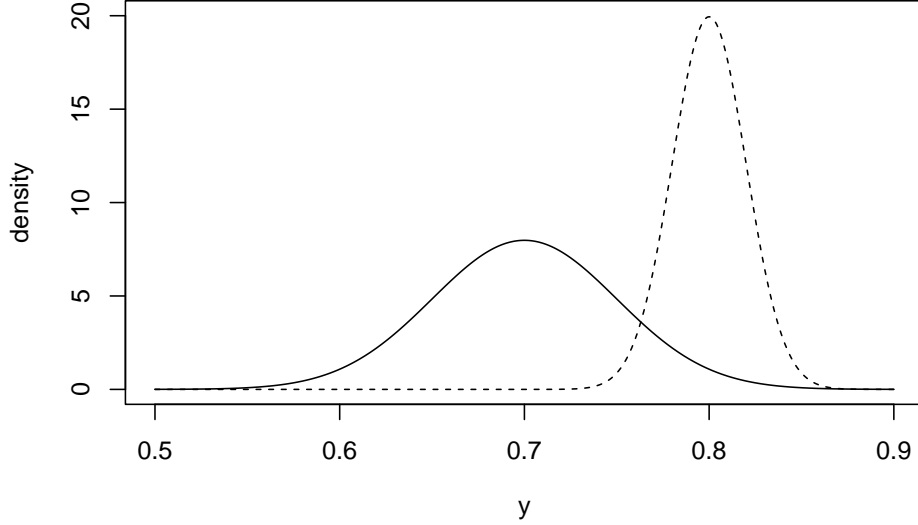
Suppose the machine is processing a bag of letters addressed to people in either B1 or B2 postcodes. The camera scans the first two characters of the postcode (B1 or B2) and records the proportion of the scanned image that is taken up by the characters. The picture below shows an example of what the scanned image looks like.



We introduce a latent variable $z_i \sim \text{Bernoulli}(p)$ that describes if the characters on the i^{th} image are B1 or B2. The observation y_i is the proportion of the i^{th} image that is taken up by the characters. We observe y_i , but want to estimate z_i . The difficulty is there lack of one-to-one correspondence between the values y_i can take and the value z_i . Due to the different handwriting and fonts used on envelopes, if the letter is going to B1 ($Z = 1$), then $Y_i \sim N(0.7, 0.05^2)$ and if it is going to B2 ($Z = 2$), then $Y_i \sim N(0.8, 0.02^2)$. The plot below shows the two densities and the overlap between them.

```
a <- seq(0.5, 0.9, 0.001)
x <- dnorm(a, 0.7, 0.05)
y <- dnorm(a, 0.8, 0.02)
```

```
plot(a, x, type = 'l', ylim = c(0, 20), xlab = expression(y), ylab = "density")
lines(a, y, lty = 2)
```



As the variables z are latent, the observed data likelihood function is

$$\pi(y | p) = \prod_{i=1}^N [p\pi(y_i | \mu = 0.7, \sigma^2 = 0.05^2) + (1 - p)\pi(y_i | \mu = 0.8, \sigma^2 = 0.02^2)].$$

Instead, it's easier to work with the complete data likelihood function, supposing we had observed the variables z . This is given by

$$\begin{aligned} \pi(y, z | p) &= \binom{N_1 + N_2}{N_1} p^{N_1} (1 - p)^{N_2} \prod_{i; z_i=1} \pi(y_i | \mu = 0.7, \sigma^2 = 0.05^2) \\ &\quad \times \prod_{i; z_i=2} \pi(y_i | \mu = 0.8, \sigma^2 = 0.02^2), \end{aligned}$$

where N_1 and N_2 are the number of letters for B1 and B2 respectively. This form makes it much easier to derive the posterior distributions and estimate the parameter values.

We place a uniform prior distribution on the parameter p , which gives the posterior distribution

$$p | y, z \sim \text{Beta}(N_1 + 1, N_2 + 1).$$

The distribution of z_i given the parameter p and the observation y_i can be derived using Bayes' theorem

$$p_i^* = \pi(z = 1 | p, y_1) = \frac{p\pi(y_i | \mu = 0.7, \sigma^2 = 0.05^2)}{p\pi(y_i | \mu = 0.7, \sigma^2 = 0.05^2) + (1 - p)\pi(y_i | \mu = 0.8, \sigma^2 = 0.02^2)}.$$

The full conditional distribution is therefore $z_i \mid y, p \sim \text{Bernoulli}(p_i^*)$.

An MCMC algorithm for this would repeat the following two steps:

1. Sample $p \mid y, z \sim \text{Beta}(N_1 + 1, N_2 + 1)$.
2. Sample $z_i \mid y, p \sim \text{Bernoulli}(p_i^*)$ for each i .

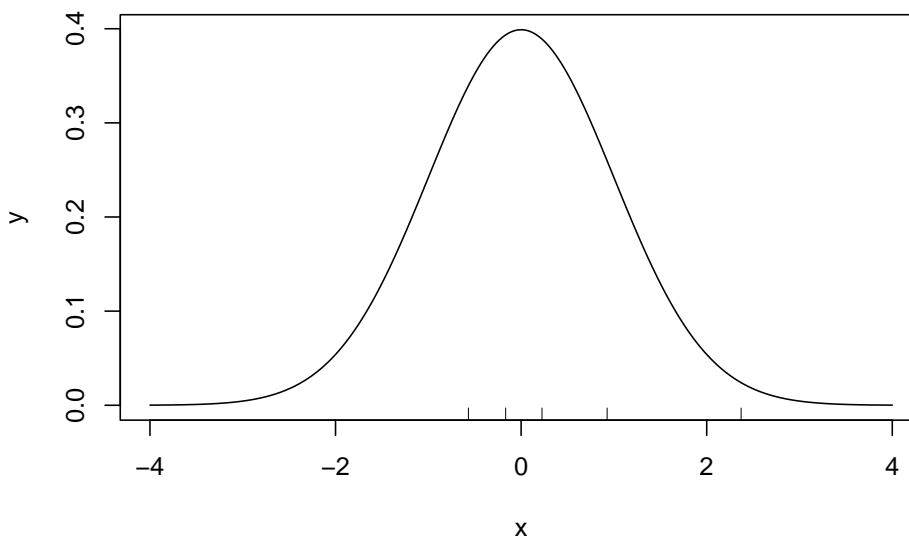
5.2 Gaussian Processes

So far in the module, we have considered prior distribution on parameters. These parameters have taken values (mostly real) or real-valued vectors. In this section, we're going to extend this idea further to place prior distributions on functions. That is, we're going to describe a prior distribution, that when sampled, gives us functions. The method we're going to use is called a Gaussian Process (GP).

Before, we define a GP, we're going to build an intuitive definition of it. Recall the normal distribution with mean μ and variance σ^2 , $N(\mu, \sigma^2)$. It assigns probabilities to values on the real line – when we sample from it, we get real values. The plot below shows the density function for a $N(0, 1)$ distribution and five samples.

```
#Plot N(0, 1)
x <- seq(-4, 4, 0.01)
y <- dnorm(x)
plot(x, y, type = 'l')

#Add samples
samples <- rnorm(5)
rug(samples)
```

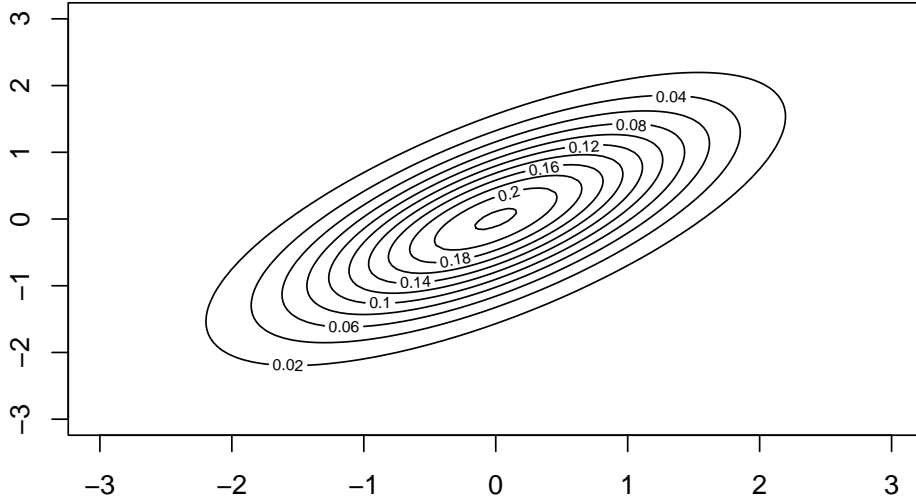


The multivariate normal distribution extends this to a vector space, \mathbb{R}^N . Instead of having a mean and variance value, the distribution is defined through a mean vector and covariance matrix. The mean vector describes the expected value of each component of the vector and the covariance matrix describes the relationship between each pair of components in the vector. When we draw samples, we get vectors. The plot below shows the density of the multivariate normal distribution with $N = 2$, zero mean, $\sigma_x^2 = \sigma_y^2 = 1$ and $\rho = 0.7$.

```
#Create Grid
x <- seq(-3,3,length.out=100)
y <- seq(-3,3,length.out=100)

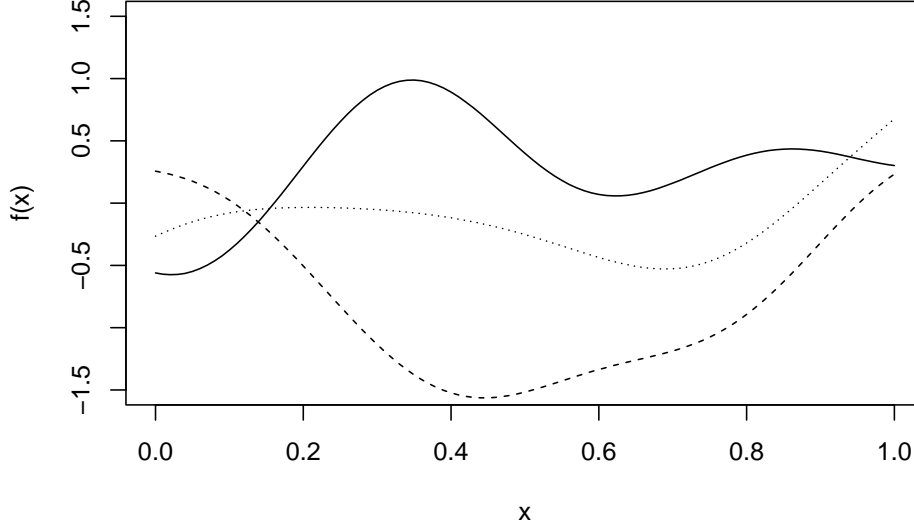
#Evaluate density at grid
z <- matrix(0,nrow=100,ncol=100)
mu <- c(0,0)
sigma <- matrix(c(1, 0.7, 0.7, 1),nrow=2)
for (i in 1:100) {
  for (j in 1:100) {
    z[i,j] <- mvtnorm::dmvnorm(c(x[i],y[j]),
                                mean=mu,sigma=sigma)
  }
}

#Generate contour plot
contour(x, y ,z)
```



A GP takes this one step further and puts a prior distribution on a function space. It is specified by a mean function, $\mu(\cdot)$ and covariance function $k(\cdot, \cdot)$. The mean function describes the expected value of each point the function can be evaluated at, and the covariance function describes the relationship between

each point on the function. The plot below shows three samples from a GP distribution with mean function the zero function $\mu(x) = 0 \forall x$ and a covariance function that supports smooth functions.



Definition 5.3. A **Gaussian Process** is a collection of random variables, any finite number of which have a joint Gaussian distribution.

This says that if we think of a function as an infinite collection of points, then if any finite subset of those points following a Gaussian distribution, we have a Gaussian process. In reality, we set up the function so that it meets this definition. More formally,

Definition 5.4. A **GP distribution on a function** $f(x)$ is defined through its mean function $\mu(x) = \mathbb{E}(f(x))$ and covariance function $k(x, x') = \mathbb{E}((f(x) - \mu(x))(f(x') - \mu(x')))$. We write it as $f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$.

Before we go any further, it is worth proceeding with caution. Those with good memories, we recall Bernstein-von-Mises' theorem from Chapter 3.

Theorem 5.1 (Bernstein-von-Mises). *For a well-specified model $\pi(y | \theta)$ with a fixed number of parameters, and for a smooth prior distribution $\pi(\theta)$ that is non-zero around the MLE $\hat{\theta}$, then*

$$\left\| \pi(\theta | y) - N \left(\hat{\theta}, \frac{I(\hat{\theta})^{-1}}{n} \right) \right\|_{TV} \rightarrow 0.$$

Bernstein-von-Mises' theorem only holds when the model has a fixed (i.e. finite) number of parameters. A GP is defined on an infinite collection of points, and so this theorem does not hold. This is the first time in this module we have

encountered a distribution where Bernstein-von-Mises' theorem does not hold. Fortunately, various forms of Bernstein-von-Mises' theorems for GPs exist, with many coming about in the early 2010s. However, this is still an ongoing area of research.

5.2.1 Covariance Functions

One issue when using GPs is describing the covariance function. How do we decide how each pair of points (there being an infinite number of them)? There are lots of standard choices of covariance functions that we can choose from, each one making different assumptions about the function we are interested in.

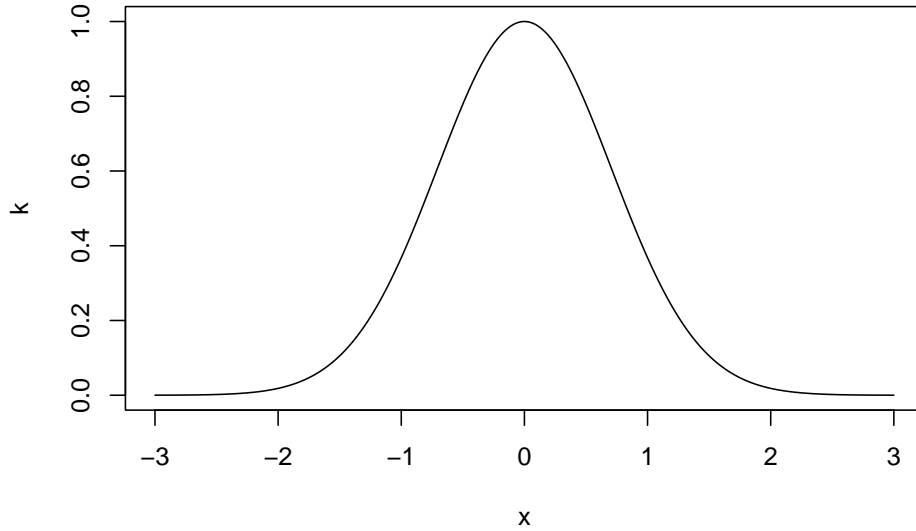
The most common covariance function is the squared exponential functions. It is used to model functions that are 'nice', i.e. they are smooth, continuous and infinitely differentiable.

Definition 5.5. The **squared exponential covariance function** takes the form

$$k(x, x') = \alpha^2 \exp\left\{-\frac{1}{l}(x - x')^2\right\},$$

where α^2 is the signal variance and l the length scale parameter.

For now, consider $\alpha = l = 1$. What is the covariance between the function evaluated at 0 and the function evaluated at x ? The plot below shows the covariance.



The covariance is highest when the x is near to 0, i.e. the points are immediately next to each other. If the value of x is ± 2 , the covariance is 0. As we are dealing with a joint normal distribution, a covariance of 0 implies independence. So with this covariance function, the value of $f(x)$ is independent of $f(0)$ if $|x|$ is larger than about two. The parameter l is called the length scale parameter