Bayesian Inference and Computation

Dr Rowland Seymour

Semester 2, 2024

Contents

Practicalities

0.1 Module Aims

Bayesian inference is a set of methods where the probability of an event occurring can be updated as more information becomes available. It is fundamentally different from frequentist methods, which are based on long running relative frequencies. This module gives an introduction to the Bayesian approach to statistical analysis and the theory that underpins it.

Students will be able to explain the distinctive features of Bayesian methodology, understand and appreciate the role of prior distributions and compute posterior distributions. It will cover the derivation of posterior distributions, the construction of prior distributions, and inference for missing data. Extensions are considered to models with more than a single parameter and how these can be used to analyse data. Computational methods have greatly advanced the use of Bayesian methods and this module covers, and allows students to apply, procedures for the sampling and analysis of intractable Bayesian problems.

By the end of the course, students should be able to:

1. Demonstrate a full and rigorous understanding of all definitions associated with Bayesian inference and understand the differences between the Bayesian and frequentist approaches to inference
2. Demonstrate a sound understanding of the fundamental concepts of Bayesian inference and computational sampling methods
3. Understand how to make inferences assuming various population distributions while taking into account expert opinion and the implications of weak prior knowledge and large samples
4. Demonstrate an understanding of the principles of Markov Chain Monte Carlo and be able to programme an MCMC algorithm
5. Engage in Bayesian data analysis in diverse situations drawn from physics, biological, engineering and other mathematical contexts.

0.2 Module Structure

The module is split between theory and computation Each week (excluding week 6) will have three lectures, one computer lab and one problem class.

0.3 Assessment

Assessment for this module is 50% via an exam and 50% via coursework assignments during the semester. The exam will last 1h 30m and take place during the summer exam period. There will be three coursework assignment – assignment 1 will be worth 10% of the final mark, with assignments 2 and three counting for 20% each. More details about the assignments will be made available 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.

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 long 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 that 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

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

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

Beta distribution

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

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

Gamma distribution

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

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

Exponential distribution

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

where $\lambda > 0$.

Poisson distribution

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

where $\lambda > 0$.

Binomial distribution

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

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 than 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:

1. Inference for quantities that are unobserved or haven't happened yet. Examples of this might be the size of a payout an insurance company has to make, or a patients outcome in a clinical trial had they been received a certain treatment.
2. Inference for quantities that are not possible to observe. This is usual because they are part of modelling process, like parameters in a linear model.

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.

The sampling distribution is an important concept in frequentist theory as it describes the randomness in the process. From a frequentist standpoint, we have a model containing some parameter θ and some data y. All the evidence in the data y about θ is contained in the likelihood function $\pi(y \mid \theta)$. The parameter θ is fixed and the likelihood function describes the probability of observing the data y given the parameter θ.

The most common way to estimate the value of θ is using maximum likelihood estimation. Although other methods do exist (e.g. method of moments, or generalised maximum likelihood estimation).

Definition 1.3. The maximum likelihood estimate of θ, $\hat{\theta}$, is the value such that $\hat{\theta} = \max_\theta \pi(y \mid \theta)$.

Uncertainty around the maximum likelihood estimate is based on the theory of long running events that underpin frequentist theory.

Definition 1.4. Let X be a random sample from a probability distribution θ. A $100(1-\alpha)$ **confidence interval** for θ is an interval $(u(Y), v(Y))$ such that

$$\mathbb{P}(u(Y) < \theta < v(Y)) = 1 - \alpha$$

This means that if you had an infinite number of samples for Y and the corresponding infinite number of confidence intervals, then $100(1-\alpha)\%$ of them would contain the true value of θ. It does *not* mean that there is a $100(1-\alpha)$ probability a particular interval contains the true value of θ.

Given that we want to understand the properties of θ given the data we have observed y, then you might think it makes sense to investigate the distribution $\pi(\theta \mid y)$. This distribution says what are the likely values of θ given the information we have observed from the data y. We will talk about Bayes' theorem in more detail later on in this chapter, but, for now, we will use it to write down this distribution

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

This is where frequentist theory cannot help us, particularly the term $\pi(\theta)$. Randomness can only come from the data, so how can we assign a probability distribution to a constant θ? The term $\pi(\theta)$ is meaningless under this philosophy. Instead, we turn to a different philosophy where we can assign a probability distribution to θ.

1.3 Bayesian Probability

The Bayesian paradigm is built around a different definition of probability. This allows us to generate probability distirbutuions for parameters values.

Definition 1.5. 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 1.2. 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 1.6. Given two events A and B, the **conditional probability** that event A occurs given the event B has already occurred is

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

when $\pi(B) > 0$.

Definition 1.7. Two events A and B are **independent** given event C if and only if

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

Definition 1.8. 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 1.3. If $Y_i \sim Bin(n, p)$ 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 1.4. 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 exchangable, 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 exchangable.*

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 $\{y_1, \dots, y_N\}$

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

□

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 varibales, 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*

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

Proof.

$$\begin{aligned} \pi(\theta \mid y) &= \frac{\pi(\theta, y)}{\pi(y)} && (1.2) \\ \implies \pi(\theta, y) &= \pi(\theta \mid y)\pi(y) && (1.3) \end{aligned}$$

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

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

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

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

□

There are four terms in Bayes' theorem:

1. The **posterior distribution** $\pi(\theta \mid y)$. This tells us our belief about the model parameter θ given the data we have observed y.
2. The **likelihood function** $\pi(y \mid \theta)$. 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

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

> **Some history of Thomas Bayes**. Thomas Bayes was an English theologean born in 1702. His "Essay towards solving a problem in the doctrine of chances" was published posthumously. It introduces theroems 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 existance of God, stating: The purpose I mean is, to show what reason we have for believing that there are in the constitution of things fixt 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 intepretation of it. The Bayesian way of thinking was really popularised by Laplace, who wrote about deductive probability in the early 19th century.

Example 1.5. 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,

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

We know the form of $\pi(y \mid \theta) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(y-\theta)^2}$, 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: $y = 0$.

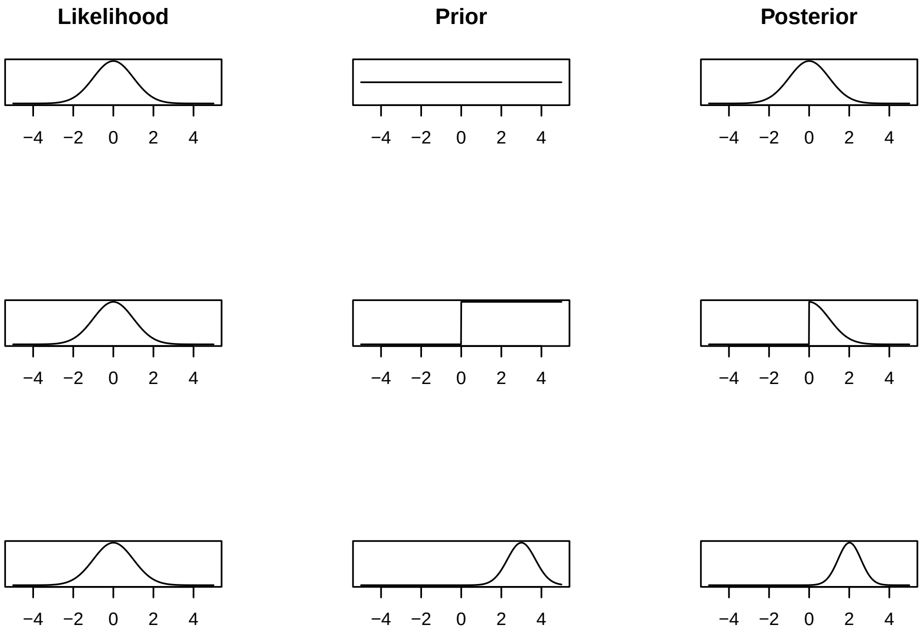
Now we have all the ingredients to construct the posterior distribution. We multiply the likelihood function evaluated at $y = 0$ by each of the three prior distributions. This gives us the posterior distributions. These are

1. For the uniform prior distribution, the posterior distribution is $\pi(\theta \mid y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\theta^2\right)$ for $\theta \in [-1000, 1000]$.

2. For the uniform prior distribution, the posterior distribution is $\pi(\theta \mid y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\theta^2\right)$ for $\theta \in [-1000, 1000]$.
3. For the normal prior distribution, as we are only interested in the posterior distribution up to proportionality, we can write it as $\pi(\theta \mid y) \propto \exp\left(-\frac{1}{2}\theta^2\right) \exp\left(-\frac{1}{2}\left(\frac{\theta-3}{0.7}\right)^2\right)$. Combining like terms, gives $\pi(\theta \mid y) \propto \exp\left(-\frac{1}{2}\left(\frac{1.7\theta^2-6\theta}{0.7^2}\right)\right)$ for $\theta \in \mathbb{R}$.

```r
#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
```

```r
#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.

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 of 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.

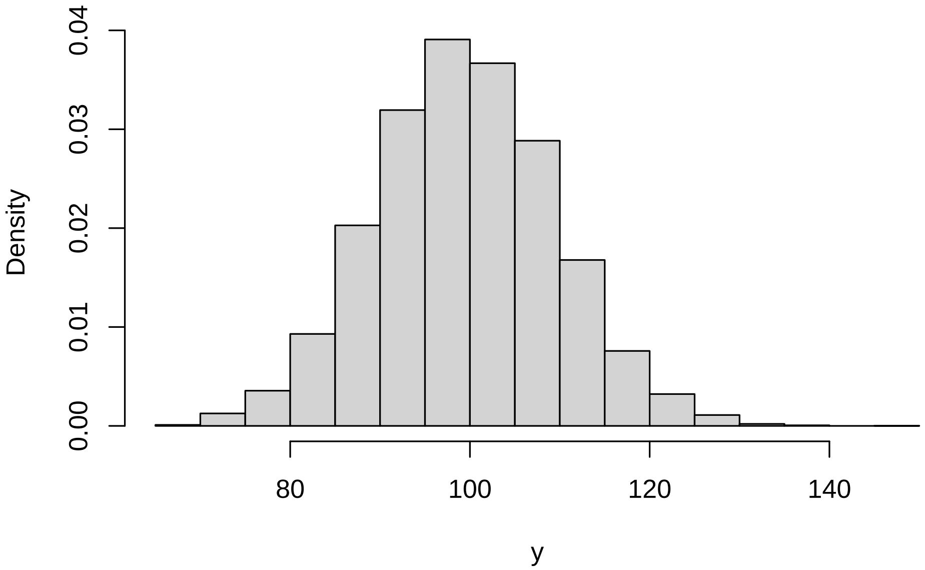
R has four built-in functions for working with distributions. They take the form rdist, ddist, pdist, and qdist. You replace the dist part with the name of the distribution you want to work with, for example unif for the uniform distribution or norm for the normal distribution. As we are working with the the Poisson distribution, we will use pois. The prefixes allow you to work with the distribution in different ways: r gives you random numbers sampled form the distribution, d evaluates the density function, p evaluates the density function, and q evaluates the inverse density function (or quantile function).

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)
```

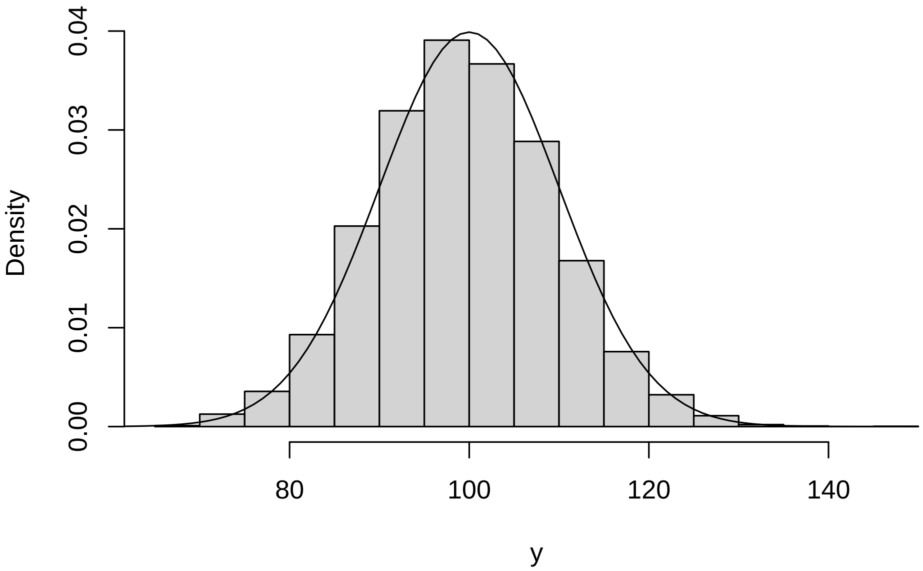
We can generate a histogram of y using the hist command. Setting freq = FALSE, makes R plot a density histogram instead of a frequency histogram. Typing ?hist will give you more information about this

```
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

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

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(market.index, 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 cirlce
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

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

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 rate 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 is 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 build 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]  0.30435145 -0.81130779  0.06190906  0.42479362  0.51982703  0.31529447
##  [7] -0.41550978  0.61593000 -0.70939597  0.95880796
```

The functions 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 above 10 random numbers were generated from the interval [-1, 1]. In R, you don't need to label the arguments, so the following will sample the same number of samples from the same interval:

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

Although in most cases it helps to label the arguments for readability and avoiding undefined behaviour. Note that if you decide to omit the argument names in the function call, the arguments must appear exactly in the order defined by the function prototype (check the documentation ?function for specific 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

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

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. Write a function to evaluate the log of the probability density function of a Poisson distribution with rate λ.

Exercise 2.6. 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.

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 functions 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(X,type ='l')
hist(X[50,]);hist(X[100,]);quantile(X[50,], c(0.25, 0.5, 0.75));quantile(X[100,], c(0.2
```

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 3 is much more practical. We are going to be deriving posterior distributions and proving when it does and doesn't work.

3.1 The Binomial Distribution

The first example we are 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 engineer collects a random sample 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 are 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, we obtain the following,

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

Likelihood function $\pi(y \mid \theta)$. We observe 200 trials each with the same probability of success (denoted by θ) and probability of failure (given by $1 - \theta$). The Binomial distribution seems the most suitable way of modelling this. Therefore, the likelihood function is given by,

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

assuming that any two accounts being a bot are independent of one another.

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

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

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

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

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

3.2 Reporting Conclsuions from Bayesian Inference

In the previous example, we derived the posterior distribution $\theta \mid y \sim \text{Beta}(4, 198)$. 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 out beliefs. In a frequentist framework, this would be via a confidence interval. The Bayesian analogues 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

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

Likelihood function $\pi(y \mid \lambda)$. 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 of one another, 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 $\lambda \sim \text{Exp}(\gamma)$.

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

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

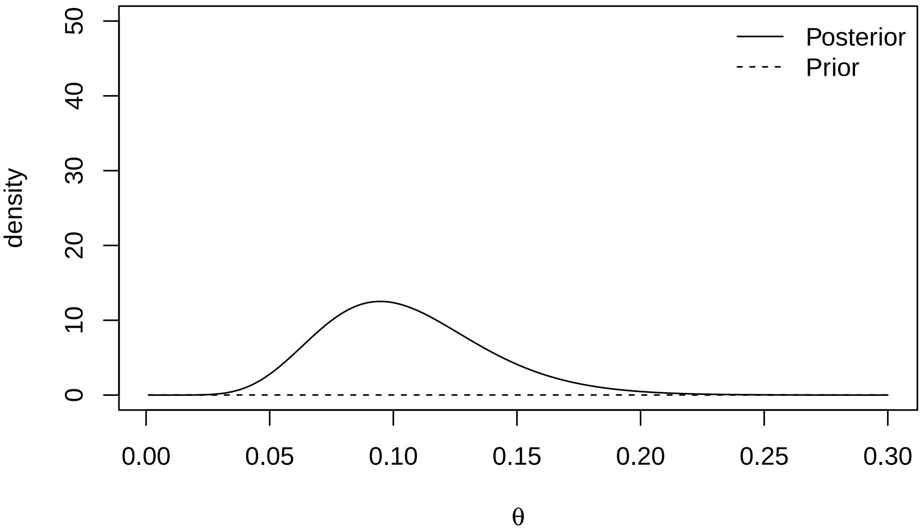
The functional form tells us that the posterior distribution is a Gamma distribution. The PDF of a gamma random variable with shape α and rate β is

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

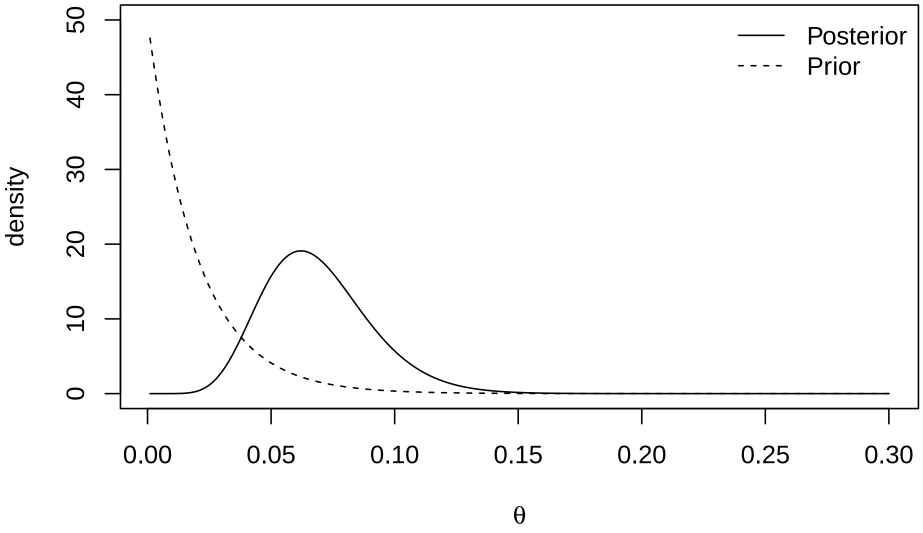
The distribution of the rate of the claims given the observed data is $\lambda \mid y \sim \text{Gamma}(10, 95 + \gamma)$.

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 $\gamma = 50$.

```r
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 Distribtuion

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. Suppose we observe N data points $y = \{y_1, \dots, y_N\}$ and we assume $y_i \sim N(\mu, \sigma^2)$ and each observation is independent. Suppose that, somehow, we know the population standard deviation and we wish to estimate the population mean μ. By Bayes' theorem, the posterior distribution is

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

Likelihood function. As the observations are independent, the likelihood function is given by the product of the N normal density functions as follows,

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

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.

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

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

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

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

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

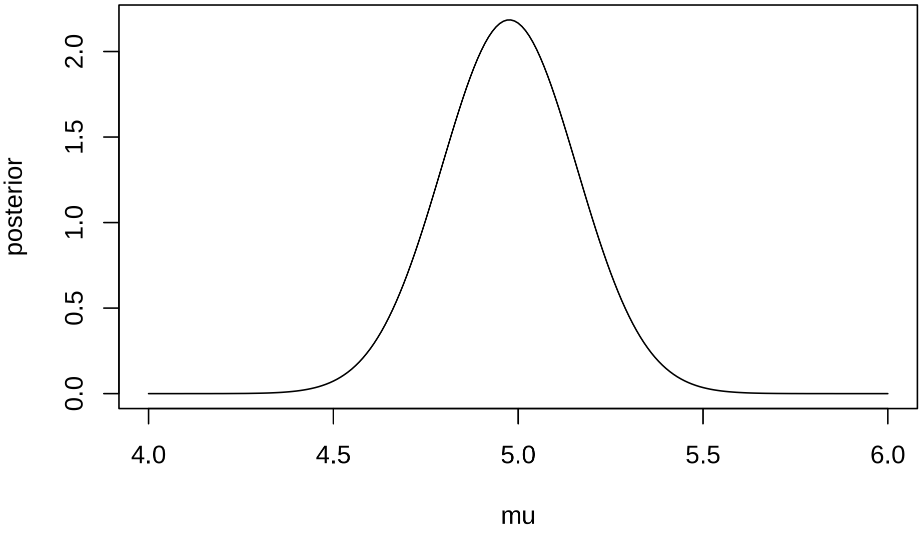
The posterior mean is partially defined through the weighted average of the population and prior means, where the weighting depends on the number of data points collected and how precise the distributions are.

Now we have derived the posterior distribution, we can explore it using R. We simulate some data with $N = 30$, $mu = 5$ and $\sigma^2 = 1$.

```
#data
N <- 30
sigma <- 1
y <- rnorm(N, 5, sigma)

#prior
sigma0 <- 1000
mu0     <- 0

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

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

```
## [1] 4.97575608 0.03333333
```

```
#Create plot
mu <- seq(4, 6, 0.01)
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] 4.617917 5.333595
```

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 Hierarchical 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 3.3 again. We have some data y that are assumed to have been generated from an Exponential distribution with rate parameter 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 hierachy formed is

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

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

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

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.

3.6 Prediction

In many cases, although we are interested in drawing inference for the model parameters, what we may also be interested in is predicting new values, whose distribution is determined by the model parameters and observed data.

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

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

Example 3.6. 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 their coursework in late, denoted by Y, using a Binomial distribution, i.e. $Y \sim \text{Bin}(n, \theta)$ where n is the number of students and θ is the probability of any particular student handing in their coursework late. As in Example 3.1, we assign a uniform prior distribution to $\theta \sim U[0, 1]$. Given then observed data, we can derive $\theta \mid y \sim Beta(4, 28)$ (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

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

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 Beta$(5 + z, 58 - z)$ random variable. This integrates to 1.

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

This code implements the distribution

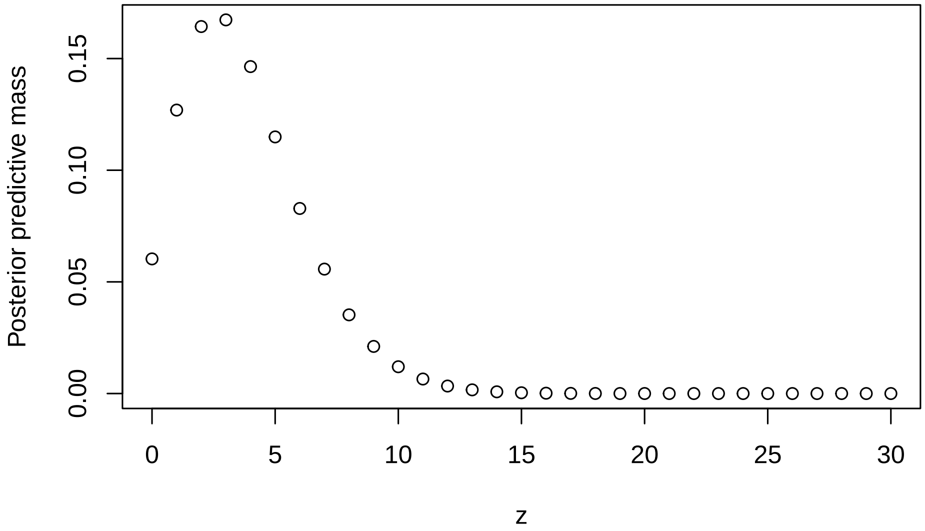
```r
beta.binom.posterior.predictive.distribution <- function(z){


  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 that our posterior predictive distribution is a valid probability mass function by checking that the probabilities sum to one.

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

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

```
plot(z, ppd, xlab = "z", ylab = "Posterior predictive mass")
```



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 Distibrutions

We have seen in a few examples how the choice of the prior distribution (and prior parameters) can impact 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 3.7. In Example 3.1, we assigned a uniform prior distribution to the parameter θ.

Such a prior distribution can have interesting and perhaps unintended side effects. Suppose we do indeed have some parameter θ and we place a uniform prior distribution on θ such that $\theta \sim U[0,1]$. This means, for example, our prior beliefs about θ are that it is equally likely to be in $[0, 0.1]$ as it is to lie in $[0.8, 0.9]$ or any other interval of size 0.1. However, our prior beliefs about θ^2 are not uniform. Letting $\phi = \theta^2$, changing variables gives $\pi(\phi) = \frac{1}{2\sqrt{\phi}}$, something that is not uniform. That raises the question, if we have little to say about θ *a priori*, shouldn't we have little to say about any reasonable transformation of 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

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

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. □

Example 3.8. In Example 3.1, we modelled the number of bot accounts on a social media website by $Y \sim \mathrm{Bin}(n, \theta)$. To construct Jeffrey's prior distribution for θ, we must first derive the Fisher information.

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

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*

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

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 Berstein-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. The proof of this theorem is out of the scope of this module, but can be found in Asymptotic Statistics (2000) by A. W. van der Vaart.

3.9 Lab

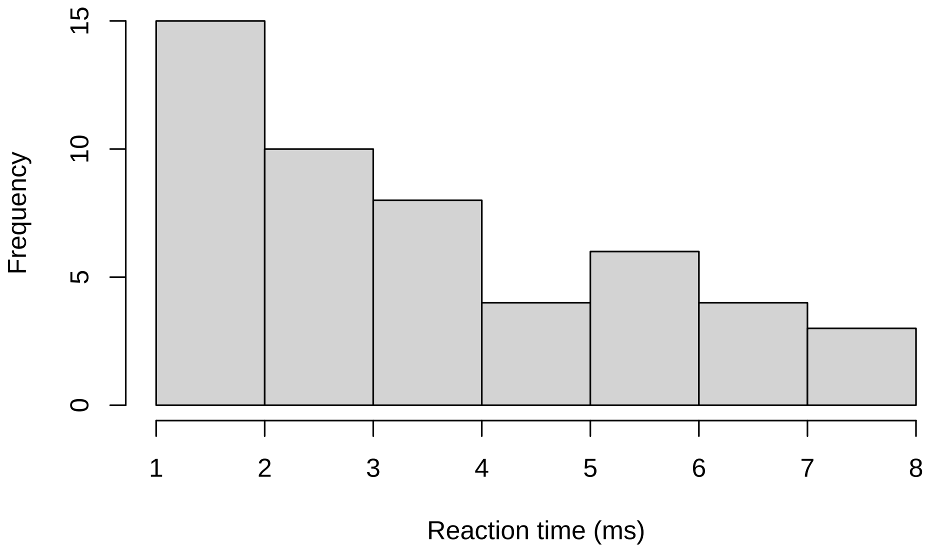
The aim of this lab is to work with some posterior distributions in cases when the prior distribution is or is not conjugate. Recall the definition of a conjugate prior distribution:

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**.

Working with conjugate prior distributions often makes the analytical work much easier, as we can work with the posterior distribution. But sometimes, conjugate prior distributions may not be appropriate. This is where R can help, as we do not need a closed form to carry out computations.

Example 3.9. The total number of goals scored in 50 games of a low level football league is shown below.

```r
y <- c(2, 6, 2, 3, 4, 3, 4, 3, 1, 2, 3, 2, 6, 6, 2, 3, 5, 1, 2, 2, 4, 2, 5, 3,
       6, 4, 1, 2, 7, 8, 4, 3, 7, 3, 3, 5, 2, 6, 1, 3, 7, 4, 2, 6, 8, 8, 4, 5,
       7, 4)
hist(y, main = "", xlab = "Reaction time (ms)")
```



```r
mean(y)
```

```
## [1] 3.92
```

We can model the number of goals scored using a Poisson distribution

$$y \sim \text{Po}(\lambda).$$

By Bayes' theorem, the posterior distribution is given by

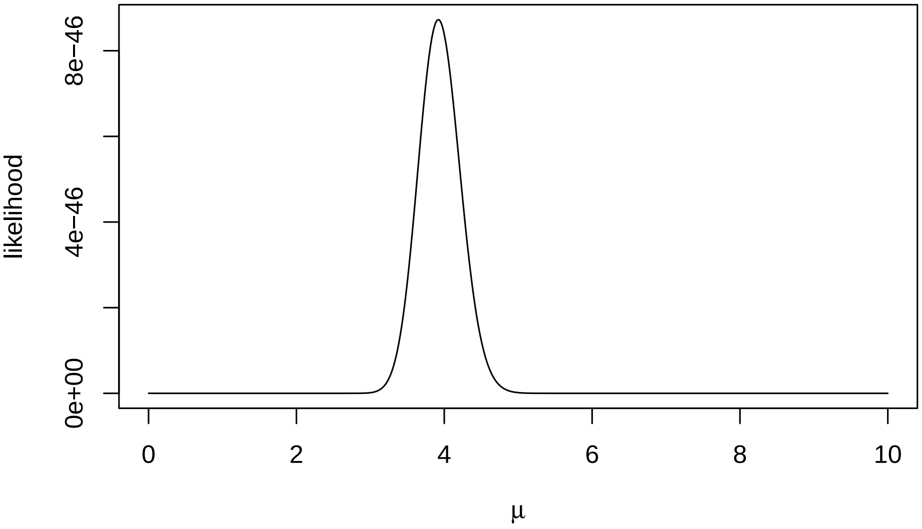
$$\pi(\lambda \mid y) \propto \pi(y \mid \lambda)\pi(\lambda).$$

The likelihood function is given by

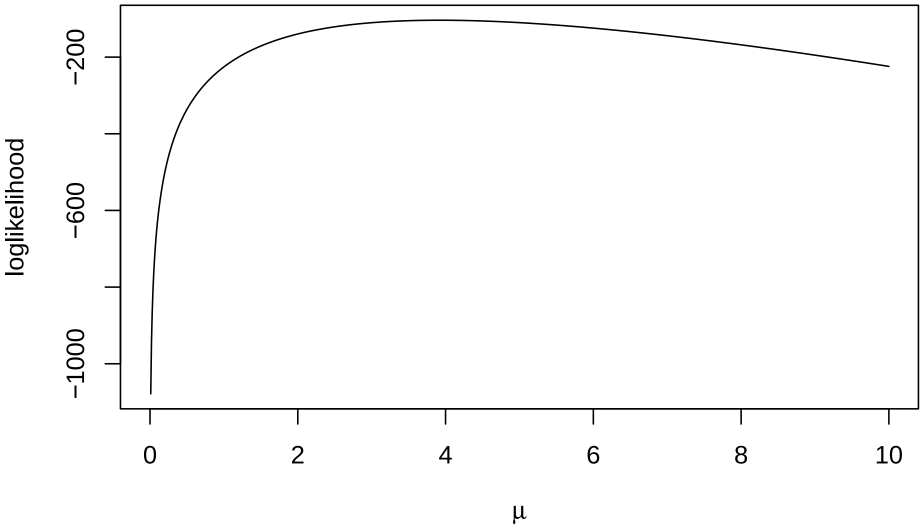
$$\begin{aligned}\pi(y \mid \lambda) &= \prod_{i=1}^{5} 0\frac{e^{-\lambda}\lambda^{y_i}}{y_i!} \\ &= \frac{e^{-50\lambda}\lambda^{\sum y_i}}{\prod_{i=1}^{5} 0y_i!}\end{aligned}$$

R has a set of inbuilt functions for working with the Poisson distribution so we can rely on those to write functions for the likelihood and loglikelihood.

```
lambda <- seq(0, 10, 0.01) #grid of lambda values
likelihood.function <- function(lambda, y) prod(dpois(y, lambda)) #compute likelihood
log.likelihood.function  <- function(lambda, y) sum(dpois(y, lambda, log = TRUE)) #com
likelihood <- sapply(lambda,  likelihood.function, y) #evaluate at grid of points
log.likelihood <- sapply(lambda,  log.likelihood.function, y) #evaluate at grid of poi

#Plot likelihood
plot(lambda, likelihood,
     xlab = expression(mu), ylab = "likelihood", type = 'l')
```



```
plot(lambda, log.likelihood,
     xlab = expression(mu), ylab = "loglikelihood", type = 'l')
```



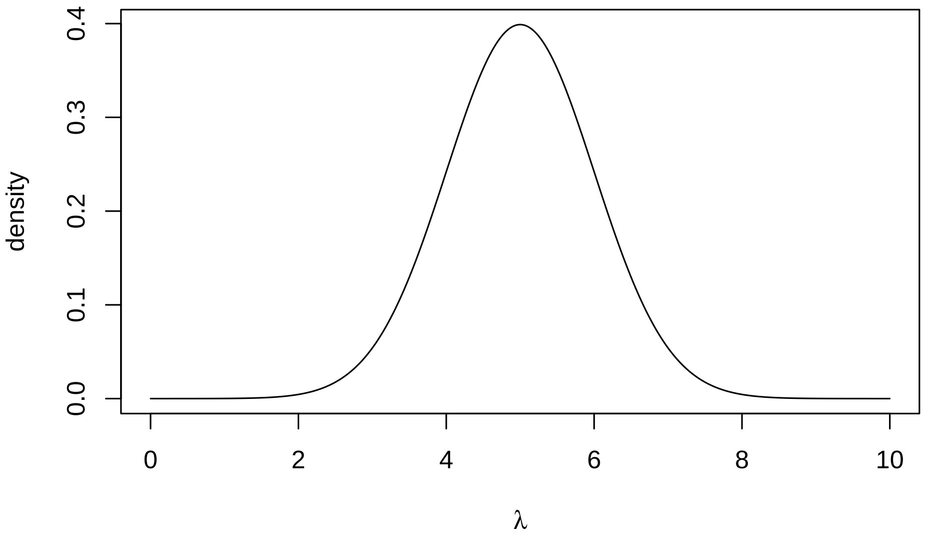
When coding posterior distributions, we often work on the log scale because the numbers can be smaller that R can deal with. The denominator with the factorial can get very large very quickly.

After speaking to football experts, we decide to place a normal prior distribution on λ with mean 5 goals and standard deviation one goal, i.e.

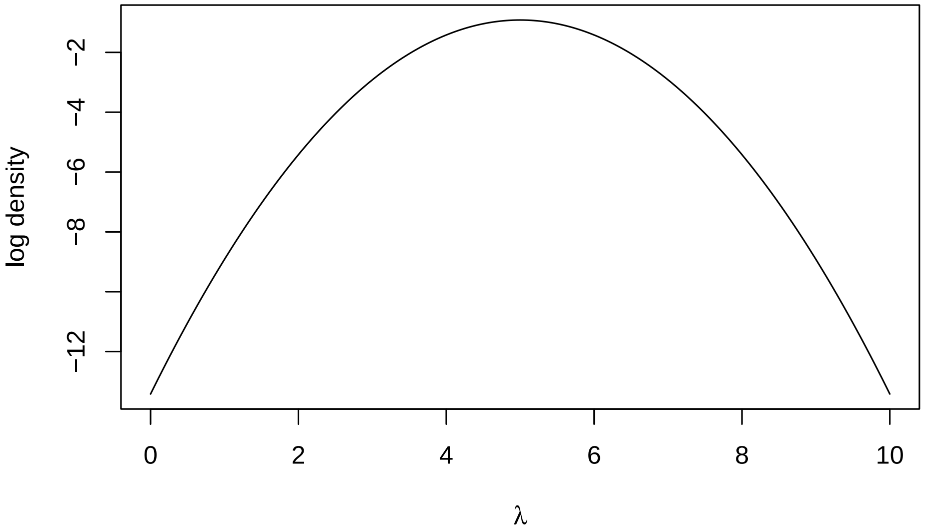
$$\lambda \sim N(5,1).$$

The prior distribution can be plotted by

```
lambda    <- seq(0, 10, 0.01) #grid of lambda values
prior     <- dnorm(lambda, 5, 1)
log.prior <- dnorm(lambda, 5, 1, log = TRUE)
plot(lambda, prior, type = 'l', xlab = expression(lambda), ylab = "density")
```



```r
plot(lambda, log.prior, type = 'l',
     xlab = expression(lambda), ylab = "log density")
```



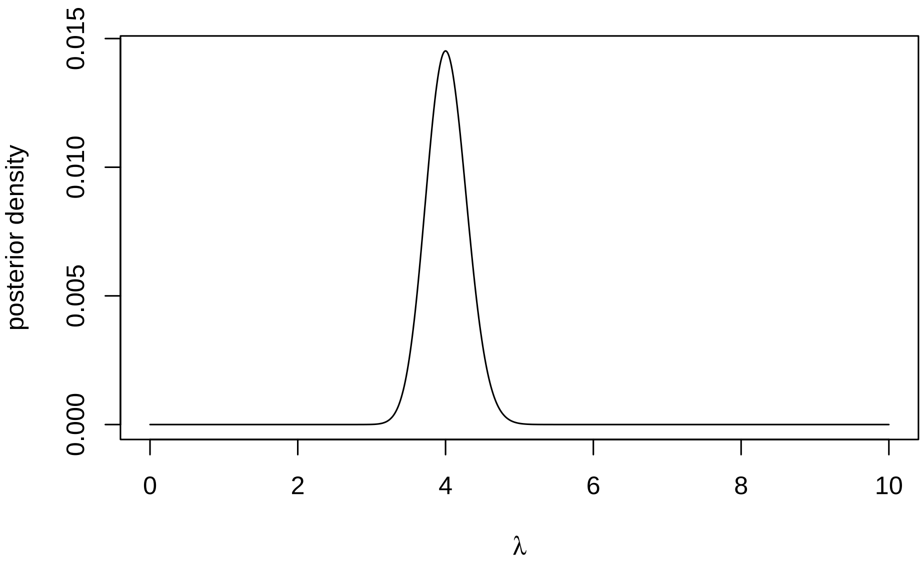
Writing the posterior distribution up to proportionality, we get

$$\pi(\lambda \mid y) \propto \exp\left(-50\lambda - \frac{1}{2}(\lambda - 5)^2\right)\lambda^{\sum y_i}.$$

There is no closed form for this distribution and it is not that nice to work with. But with R, we can easily evaluate the posterior distribution at a grid of points.

```r
posterior <- prior*likelihood
posterior <- posterior/sum(posterior) #normalise
```

```
plot(lambda, posterior, type = 'l', xlab = expression(lambda),
     ylab = "posterior density")
```



We can now visually inspect the posterior distribution and see that it has a strong peak around 4. One important statistic is the **maximum a posteriori estimation** or MAP estimate, this is the mode of the posterior distribution and it is a similar principle to the maximum likelihood estimate.

We can compute this using the command

```
lambda[which.max(posterior)]
```

```
## [1] 4
```

which shows the MAP estimate is exactly 4.

Exercise 3.1. Adapt the code in the Example above to use an exponential prior distribution with rate 0.1. Then derive the posterior distribution analytically and compare to the numerical version.

Exercise 3.2. You are given the data are exponentially distributed with rate λ, i.e. $Y_1, \ldots, Y_N \sim \text{Exp}(\lambda)$. 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 $\lambda \sim \text{Beta}(\alpha, \beta)$.

The data is given by

```
y <- c(1.95, 1.46, 4.81, 1.52, 4.24, 3.00, 0.46, 2.27, 1.76, 0.41)
```

By writing an R function to evaluate the likelihood function, evaute the posterior distribution for λ over a grid of points.