Practical 1: Using R for Simple Bayesian Inference

The aim of this Practical is to get some familiarity with simple Bayesian inference, and we do this by using R to obtain simple plots and summaries of posteriors from single parameter problems with conjugate priors.

Some of the functions you will need are: `curve`, `dbeta` and `qbeta`:

`curve(expr,from,to)` plots a curve corresponding to the given expression (in 'x') over the interval `(from, to)`.

`dbeta(x,a,b)` gives the density of a $Beta(a,b)$ distribution.

`qbeta(c(0.025,0.975),a,b)` calculates the 2.5% and 97.5% percentiles of a $Beta(a,b)$ distribution.

The functions `dgamma()` and `qgamma()` do similar things but for a $\Gamma(\alpha,\beta)$distribution.

Useful options for the `curve()` function are:

`main` = "*graph title*"	to add *graph title* as a title to the graph
`ylab` = "*label title*"	to add *label title* to the y-axis
`add = T`	to add this plot to an existing plot

Exercise 1

Here we are going to look at Binomial data with different priors – first of all a non-informative one and then an informative one.

Consider *n* independent trials with θ = probability of success in each trial.

y = 7 successes were observed in *n* = 20 independent trials.

What is the maximum likelihood estimate of $\hat{\theta}$? _____

(a) Using a non-informative prior $p(\theta)=1$.

What is the posterior density of θ? Record it below, stating clearly the parameters of the distribution.

Posterior density of θ: _____

Use the following function substituting in the relevant values of a and b to plot this theoretical density curve. What does the distribution look like?

```
> curve(dbeta(x,a,b),from=0,to=1)
```

From the plot what would you roughly expect a 95% credible interval for θ to be? Now use the `qbeta()` function to derive a 95% credible interval for θ.

```
> qbeta(c(0.025,0.975),a,b)
```

95% credible interval is: _____

(b) Informative prior

How do we choose the parameters a and b so that our informative prior $Beta(a, b)$ best represents whatever prior information we have about θ?

In this section we will use a Beta with a=1.4 and b=5.6. The following few lines (in italics) explain how we may see this prior as being constructed using the output of the analysis of a previous data set (but note that this is not essential, so skip it if it is confusing!).

In the binomial likelihood for the data, $p(y|\theta) \propto \theta^y(1-\theta)^{n-y}$, y is the observed number of "successes" and n – y the observed number of "failures".

The form of the beta prior $p(\theta) \propto \theta^{a-1}(1-\theta)^{b-1}$ suggests that we might think of $a-1$ and $b-1$ as the prior numbers of "successes" and "failures", respectively. We can then choose values that represent our prior knowledge best. Note that the values $a=1$ and $b=1$, corresponding to the uniform (non-informative) prior, represent zero prior "observations".

A possible method is to choose a prior mean value μ of θ, and let $\nu = a+b$ be the number of prior "observations" plus 2. The mean of the $Beta(a, b)$ distribution is $= E(\theta) = \frac{a}{a+b}$, from which it follows that

$$a = \nu\mu, \qquad b = \nu(1-\mu)$$

Suppose previous knowledge suggests an approximate mean value for θ is 0.2, and that this was derived from 5 prior observations. We have used this information to construct a beta prior distribution for θ (if you want, you can solve the two simultaneous equations above to check).

Plot the prior; and on the same axes plot the posterior density.

Calculate a 95% credible interval for θ. _____

Compare the 95% credible intervals for the informative and non-informative priors.

Optional: What happens to your results if the approximate mean value for θ is now thought to have been derived from 50 rather than 5 prior observations? Note that this requires the same reasoning as was used in the section in italics above.

_____

Practical 2: Simulation in *R*

The aim of this Practical is to give you practice at using simulation to make inferences about a posterior distribution and to explore how to sample from non-standard distributions using rejection sampling.

In the previous practical you have been introduced to functions that calculate the density of a Beta distribution and percentiles of a Beta distribution: `dbeta(x,α,β)` and `qbeta(c(0.025,0.975), α,β)` respectively. The function `rbeta(n,α,β)` enables you to generate n values from a Beta distribution $Beta(\alpha,\beta)$. Similarly the functions `rnorm(n,μ,σ)` and `rgamma(n,α,β)` generate n values from a Normal, $N(\mu,\tau)$ and Gamma, $\Gamma(\alpha,\beta)$ distribution respectively, although see additional note at the end of this practical for further details about the Gamma distribution.

Sampling from standard distributions in *R*

In previous exercises you considered the case of $n=20$ independent trials with θ = probability of success in each trial in which you observed $y=7$ successes so that $(y\mid\theta)\sim Bin(n,y)$.If the prior is $\theta\sim Beta(\alpha,\beta)$ the posterior is of the form

$$p(\theta\mid y)\propto q(\theta\mid y)=\theta^{\alpha+y-1}(1-\theta)^{\beta+n-y-1}$$
$$\Rightarrow(\theta\mid y)\quad Beta(\alpha+y,\beta+n-y)$$

In this exercise the aim is to use R to sample from the posterior distribution and compare the simulation results with theoretical results. The file **prac simulation ex standard Rcode.R** provides some R code to help you.

Open this file in an R script window (File/Open script…). Relevant lines can be submitted to R using **Ctrl R** – this will submit either the line on which the cursor sits or the highlighted text. Those lines commencing with a # are comments and will not be executed by R.

(a) For a non-informative prior $p(\theta)=1$ equivalent to $\theta\quad Beta(1,1)$, calculate $a=\alpha+y$ and $b=\beta+n-y$, the values of the parameters for the posterior distribution of $\theta\mid y$ and define them using

```
> a<-
> b<-
```

Now sample $n=100$ independent samples from the posterior density of θ and save them in `sim.post.100` using the commands

```
> n<-100
> sim.post.100<-rbeta(n,a,b)
```

Plot a histogram of these values scaled to have unit area using

```
> hist(sim.post.100,freq=F)
```
or
```
hist(sim.post.100,prob=T)
```

And add a curve

```
> curve(dbeta(x,a,b),add=T)
```

(b) Estimate a 95% credible interval for θ using

```
> quant<-c(0.025,0.975)
> cred.100<-quantile(sim.post.100,quant)
```

and compare them to the theoretical value you obtained in practical 1.

```
> cred.theory<-qbeta(quant,a,b)
```

Credible interval for θ using 100 simulations ……

Credible interval for θ from theory……

(c) Calculate log odds, $\log_e\left(\frac{\theta}{1-\theta}\right)$, for each draw from the posterior distribution

```
> log.odds.100<-log(sim.post.100/(1-sim.post.100))
```

and plot using either a histogram

```
> hist(log.odds.100)
```

or a density plot

```
> plot(density(log.odds.100))
```

And obtain a 95% credible interval for the log odds

```
> quantile(log.odds.100,quant)
```

Credible interval for the log odds from 100 simulations ……

Consider the ease of obtaining the distribution and summary statistics for a function of θ using simulation compared with using distributional theory.

(d) Repeat this exercise using a sample of 10,000 values, by changing `n<-10000` and compare with the results from your sample of 100 values. To make comparison easy you might want to change the names to distinguish, for example `sim.post.10000` etc

Credible interval for θ from 10,000 simulations ……

Credible interval for the log odds from 10,000 simulations ……

Practical 3: Using WinBUGS

The aim of this Practical (which is in two parts) is to get you started with WinBUGS solving some simple problems.

The *WinBUGS User Manual*, although rather terse in places, is important and is available on-line as an HTML document.

Section 1: Learning to use WinBUGS

In this section you learn the basics of how to use WinBUGS without having to check for convergence.

Estimating the sex ratio

A particular maternal condition during pregnancy was thought to be associated with the sex of the child. The proportion of female births in the population at large was 0.485. A sample of 98 births to women with the condition had 43 female births. Is this evidence that the proportion of female births to mothers with the condition is less than the proportion in the population?

If θ denotes the proportion of female births to women with the condition, we can assume that the observed number, *y* of female births is distributed as *Bin(θ, n)* where *n* = 98.

We wish to estimate the posterior mean of θ and the posterior mean of the sex ratio $=(1-\theta)/\theta$.

(a) *Non-informative prior for* θ

Firstly we will use a non-informative prior, a uniform distribution of θ on (0,1).

Start WinBUGS.

Select the menu item **File, New** to get an empty sheet for specifying the model. Then type:

```
model
{
```

Next, specify the distribution of y by typing:

```
y ~ dbin(theta, n)
```

the non-informative prior for θ can be specified either with a uniform distribution:

```
theta ~ dunif(0, 1)
```

Alternatively, specify the equivalent beta distribution:

```
theta ~ dbeta(1, 1)
```

The sex ratio can be calculated using the code:

```
ratio <- (1-theta)/theta
```

Finally type

```
}
```

to complete the model specification.

Your model specification should look similar to the following:

```
model
{
   y ~ dbin(theta, n)
   theta ~ dbeta(1, 1)
   ratio <- (1-theta)/theta
}
```

Now select the menu item **Model, Specification** to get the **Specification Tool**. Click the **check model** button. If all is OK, the **load data** and **compile** buttons will be enabled and you will get the message "model syntactically correct" at the bottom left of the screen.

Below the model specification, type a line giving the data values for n and y:

```
list(n = 98, y = 43)
```

Select this line and click the **load data** button. Then click the **compile** button. If the compilation is successful, the **load inits** button will be enabled.

Type another line giving an initial value to theta:

```
list(theta = 0.5)
```

Select this line and click the **load inits** button.

The model is now fully specified to WinBUGS, and now we have to specify how it is to be run. Select the menu item **Inference, Samples** to display the **Sample Monitor Tool**. Type **theta** into the **node** box and click the **set** button. Repeat this for **ratio**. In the node box, type ***** and click the **trace** button. This specifies that we want to trace the sampled values of each node while running the MCMC.

Select the menu item **Model, Update**. Change the number of updates to **10000** and click **Update** to run the model. When it is completed, go back to the Sample Monitor Tool, type ***** into the node box to select all variables, and click the **density** and **stats** buttons. This will produce the posterior densities of **theta** and **ratio**, and their statistics. Note down their means, standard deviations and credible intervals:

Parameter	*posterior mean*	*posterior SD*	*95% credible interval*
theta			
ratio			

(b) *Beta prior with mean 0.485*:

We could choose to use the background population information to form a prior. To do this, we may choose as prior for θ a beta distribution with a mean of 0.485. Its parameters should be chosen to represent the amount of prior precision that we think is reasonable. In practice this will depend on the source of the population information. Modify your code for (a) so that θ has a beta prior distribution with

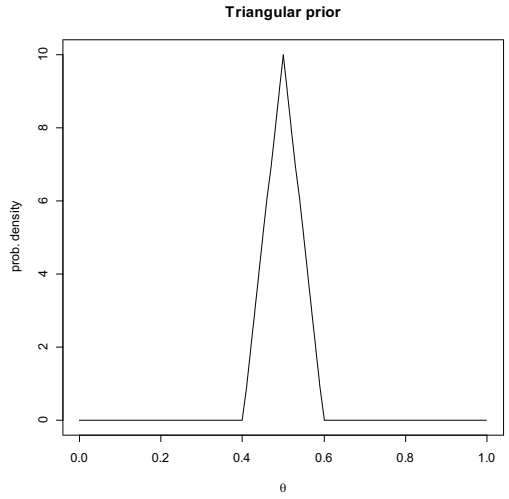
$\alpha + \beta = 100$ and a mean of 0.485 (a fairly precise informative prior for θ). This implies that $\alpha = 48.5$ and $\beta = 51.5$. Use

```
theta ~ dbeta(48.5,51.5).
```

Again record the posterior means and SDs and 95% credible intervals for the parameters, and compare the results with (a) …

Parameter	*posterior mean*	*posterior SD*	*95% credible interval*
theta			
ratio			

(c) *Non-conjugate prior*: it could be argued that priors (a) and (b) are unrealistic in that they allow values of θ close to 0 or 1, which is obviously implausible. So the triangular distribution on (0.4, 0.6) has been proposed. This has zero probability outside the range (0.4, 0.6).



There is no triangular distribution in BUGS, but you can use the fact that the sum of two uniformly distributed random variables has a triangular distribution: the code

```
theta1 ~ dunif(0.2,0.3)
theta2 ~ dunif(0.2,0.3)
theta <- theta1+theta2
```

generates the required triangular distribution for `theta`. Note that you will need to initialise `theta1` and `theta2` rather than `theta`, and will need to choose initial values in the range 0.2 – 0.3 for each.

Record the results and compare with (a) and (b) …

Parameter	*posterior mean*	*posterior SD*	*95% credible interval*
theta			
ratio			

(d) To further investigate the claim that the condition is associated with the sex of the child, we need to look at the posterior probability that $\theta < 0.485$, the population proportion of female births.

This can be found by using the `step` function. Define a new variable `diff` by

```
diff <- step(0.485-theta)
```

This makes `diff` a binary variable = 1 if $\theta < 0.485$, and = 0 otherwise.

The posterior mean of `diff` is equal to the posterior probability that $\theta < 0.485$

Find the required posterior probability for each of the three priors in (a), (b), (c).

Prior	*posterior prob. that* $\theta < 0.485$
Uniform	
Beta(48,5, 51.5)	
Triangular	

Section 2: MCMC Diagnostics

In this section you look at how to check for convergence.

The example is simple and there are no major convergence issues.

Estimating the sex ratio

We wish to rerun the sex ratio example above to check whether convergence has been reached.

The file **prac WB ex ratio.odc** contains the model, data and two sets of initial values. Note that these three sets of initial values cover the range of possible values that *θ* can take.

(a) Running the analysis.

Use the **Specification Tool** to check the model and load the data.

Before you click on **compile** change the number of chain to **2**.

Compile the model and then load the three sets of initial values – one for each chain.

Use the **Sample Monitor Tool** to monitor information on `theta`, `ratio` and `dif` and finally set * and look at the trace of these three random variables.

Run the model for 10,000 updates.

(b) Diagnostics. Look at the following in the **Sample Monitor Tool** for each of the three random variables to carry out diagnostic checking, and decide on a burn-in period if you think it has reached convergence. Note that you can control which chains are plotted, the range of iterates and whether any thinning should be applied. In most of the plots the three colours correspond to results from the three chains, chain 1 – red, chain 2 – blue, chain 3 - green.

history: This produces a plot of the iteration history (the trace) of the selected nodes

quantiles: This produces a plot of the “running mean” of the node and running 2.5% and 97.5% quantiles.

bgr: The Gelman and Rubin plot for each node. Remember that in this case the different colours do not represent different chains but different summaries.

IMPORTANT: The bgr plots are usually very small. You can expand the graph by clicking on it and then dragging one of the corners out. Usually the y-axis of the bgr plot is set between 0 and 1. This can make it very difficult to see what is going on. All graphs can be edited in WinBUGS by right clicking on the graph in question and selecting **Properties**. From here you can change the axes, titles and other parts of the graph. It is possible to change the setting for all bgr graphs at the same time by using the **All plots** tab.

autocorr: This produces autocorrelation plots.

If you wish to produce plots of the correlation between random variables you need to go **Inference, Correlations** and enter the random variables between which you want the correlation plotted.

(c) Posterior inference.

Once you have decided on a burn-in look at your posterior summaries for the relevant period – set the **beg** box in the **Sample Monitor Tool** to exclude the burn-in and look at **density** and **stats**. Note that the results from all three chains can be used together – check your sample size in the **stats** – or separately if you want to see whether inferences are the same from all three chains.

(d) In this example do you think it was necessary to run the diagnostic checks on all three random variables?

Section 3: Putting it all together

Infant weight gain

Salk (1973) collected data on weight gain among infants from birth to day 4 after birth according to whether or not they were continuously exposed to the recorded sound of the mother's heartbeat.

There were $n_1 = 20$ and $n_2 = 36$ observations in the exposed and non-exposed groups respectively. The aim of the modelling is to calculate a 95% credible interval for the difference in weight gain between the two groups. We will model the weight gain y_{ij} of individual i in group j as being drawn from a Normal distribution with a different mean and precision for each group so that:

$$y_{ij} \sim N(\mu_j, \tau_j)$$
$$\mu_j \sim N(0, 10^{-6})$$
$$\tau_j \sim Gamma(0.001, 0.001) \text{ and } \sigma_j = \frac{1}{\sqrt{\tau_j}}$$

The model, data and initial values are contained in the file **prac WB ex infant.odc**. If you wish to do any exploratory data analysis in R then the file **prac WB ex infant Rcode.R** contains the code to start you off.

Run two chains for 10,000 iterations. Carry out diagnostic checks for convergence. Once you are happy it has converged then look at the results

What are the 95% credible intervals for:

The difference in weight gain? …….

The standard deviation of group 1, σ_1…….. and group 2, σ_2 ……

Survival times with informative prior

Imagine that we have some previous information which leads us to expect an average survival time of 30 weeks, or mean of log(surv time) = 3.4, and the precision (= 1/variance) of log(surv time) is expected to be somewhere around 2.

We specify a *joint prior* $p(\mu,\tau)$ for μ and τ (the mean and precision of log(surv time).

To do this, we use the factorisation $p(\mu,\tau) = p(\mu|\tau)p(\tau)$. So we first specify the prior for τ, and then the prior for μ given τ.

Use a gamma prior for τ: $\tau \quad \Gamma(\alpha,\beta)$ and a normal prior for μ given τ: $(\mu|\tau) \quad N(\mu_0,\tau)$.
We must first choose suitable values for the parameters α, β and μ_0.

α and β must be chosen so that the prior for τ has a mean of 2 and a variance that reflects the appropriate degree of uncertainty. (This is an artificial exercise, so we just pretend that we have some idea of how the distribution should look to represent what we know.) By an appropriate degree of uncertainty we would expect about 95% of the distribution to be between 0.5 and 5.

The mean of the gamma prior is $E(\tau) = \alpha/\beta$ and its variance is $\text{var}(\tau) = \alpha/\beta^2$. So we choose α and β so that $\alpha = 2\beta$.

For this it may be helpful to plot the gamma density $\Gamma(2\beta,\beta)$ for various values of β to get a feel for the spread and to look at the middle 95% of the distribution. You could do this in *R* by executing

```
b <- 2.5; curve(dgamma(x,2*b,b), 0,8); qgamma(c(0.025,0.975),2*b,b)
```

with different values for `b` (just change the `2.5`).

Try various values of α and β with $\alpha = 2\beta$ until you get a distribution which adequately describes the prior knowledge that you are pretending to have.

Having chosen the prior for τ, choose the prior for μ: $\mu \quad N(\mu_0,\tau)$, where $\mu_0 = \log(30)$.

The file **prac WB survival.odc** contains incomplete BUGS code for the model specification. Complete this and run it.

Produce posterior densities and summary statistics for the mean and SD of log survival time, and an interval estimate for a new predicted value.

Also estimate the posterior probability that survival time > 150 weeks.

Practical 4: Linear Models, GLMs, DIC and model checking with WinBUGS

The aim of this Practical is to learn the basic ideas of using WinBUGS for (a) simple linear models and GLMs and (b) model selection with DIC

1. Regression models with DIC

The fictitious data in file **prac reg fict data.txt** contains variables *y*, x_1, x_2, x_3, and the file **prac reg fict mod.odc** contains the BUGS code for fitting a linear regression model of *y* on all three *x*-variables with non-informative priors for all parameters. i.e.

$$y_i \sim N(\mu_i, \tau)$$

$$\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}$$

$$\beta_j \sim N(0, 10^{-6}), \tau \sim \Gamma(0.001, 0.001), \tau = 1/\sigma^2$$

First, before doing a Bayesian analysis, use *R* to plot the data and fit a multiple linear regression models with all three *x*-variables. Then fit models for each pair of *x*-variables. Record the number of estimated parameters (including σ) and the *AIC* statistic for each model. R-code to help you read the data and fit the models is in the file **prac reg fict Rcode.R**. Open this in an *R* script window.

Next fit the Bayesian regression models with WinBUGS and record *DIC* and *pD* for each model. Run two chains for each model and provide suitable initial values. Remember to get convergence (using *bgr* plots to assess convergence) before setting the *DIC* (in the **inference** menu) and do not close the *DIC* dialog box once it is set.

First fit the model with all three x-variables and record 95% credible intervals for the regression coefficients:

parameter	95% credible interval
β_1	
β_2	
β_3	

Next fit the three models each with two x-variables.

Hint: to remove a term, β_3 say, from the model, in the BUGS code, replace the prior

```
beta3 ~ dnorm(0,1.0E-6)
```

with an assignment statement: `beta3 <- 0`

Complete the following table and compare *DIC* with *AIC*, and *pD* with the number of parameters:

		"Classical" models		Bayesian models	
Model	*x*-variables	No. of parameters	*AIC*	*pD*	*DIC*
1	x_1, x_2, x_3				
2	x_1, x_2				
3	x_2, x_3				
4	x_3, x_1				

Check that the DIC results are consistent with the credible intervals above.

Finally, investigate the effect on *DIC* of an informative prior which strongly conflicts with the data:

> In the model with all three variables, replace the non-informative prior for β_1 with `beta1 ~ dnorm(0,16)` i.e. $\sigma = 0.25$, low variance (high precision).

Compare the *DIC* for this model with Model 1 above.

2. Logistic regression

The following data show the number of young rats y_i from a litter of size n_i produced by a female rat i when the female is either treated, $t_i = 1$ or untreated, $t_i = 0$

Mother no. (*i*)	Litter size, n_i	No. surviving, y_i	Treatment, t_i
1	13	13	1
2	12	12	1
3	9	9	1
4	9	9	1
5	8	8	1
6	8	8	1
7	13	12	1
8	12	11	1
9	10	9	1
10	10	9	1
11	9	8	1
12	13	11	1
13	5	4	1
14	7	5	1
15	10	7	1
16	10	7	1
17	12	12	0
18	11	11	0
19	10	10	0
20	9	9	0
21	10	9	0
22	10	9	0
23	10	9	0
24	9	8	0
25	9	8	0
26	5	4	0
27	9	7	0
28	7	4	0
29	6	3	0
30	6	3	0
31	10	3	0
32	7	0	0

(a) First try a non-Bayesian logistic regression using *R*.

The data are in file **prac reg logit data.txt**. Read the data into *R* and use the following *R* code (you can find it in **prac reg logit Rcode.R**) to do a binomial logistic regression of *y* on *t* …

```
rats <- read.table("prac reg logit data.txt", header=T)
attach(rats)
resp <- cbind(y,n-y)
logr1 <- glm(resp ~ t, family=binomial)
summary(logr1)
```

From the output, note

(1) significance level of the treatment effect = ……..

(2) residual deviance and degrees of freedom = ……..

(3) *AIC* = ……..

The fact that the residual deviance is much greater than its d.f. is a symptom of overdispersion.

(b) Now for a Bayesian analysis:

Simple logistic regression:
The file **prac reg logit mod.odc** contains the data and the BUGS code for the simple binomial logistic regression model

$$y_i \sim Bin(p_i, n_i)$$

$$logit(p_i) = \beta_0 + \beta_1 t_i$$

with non-informative priors for the coefficients:

$$\beta_0 \sim N(0, 10^{-6}), \beta_1 \sim N(0, 10^{-6})$$

Choose suitable initial values and run this model for two chains. Check for convergence, set the *DIC* tool and run again. Allowing for a suitable burn-in, note the following:

(1) the posterior mean and SD of the treatment effect (β_1) = …….

(2) 95% credible interval for β_1 = ……..

(3) The *DIC* = ……..

What do you conclude about the treatment effect?

Practical 5: Metropolis-Hastings and Gibbs sampling

The aim of this Practical is to help you to understand how MCMC algorithms work by implementing them yourself. The tasks here are slightly more self-directed than on the other sheets. If you can manage to code up a Metropolis-Hastings algorithm yourself, this should provide you with a fundamental understanding of the method that will serve you well.

Sampling from a posterior using Metropolis-Hastings

Consider the example in the lecture, of sampling from the posterior distribution on the mean of a Gaussian when the variance is known. Although we know that we can perform this task using direct simulation, here we look at performing the simulation using Metropolis-Hastings. The task is to simulate 10 points from a normal distribution with some mean and variance, then to implement three approaches for finding the posterior expectation of the mean given this simulated data:

1. By calculating the true posterior (the formula is given in the lecture), and taking the expectation of this known distribution (this is the true answer).
2. By simulating directly from the true posterior, and estimating the posterior expectation from these points.
3. By simulating from the posterior using the Metropolis-Hastings algorithm, and estimating the posterior expectation from these points.

Implementing Metropolis-Hastings is not completely straightforward. I suggest the following steps.

A. Ensure that you know the likelihood and the prior on the mean.
B. Set up a vector to store the points that you simulate.
C. Set an initial value for the MCMC – chose something that is not too far away from the true answer.
D. Try to implement the Metropolis-Hastings algorithm shown in the lecture. Note that you start by setting up a “for” loop.
E. You need to choose a proposal distribution (q) to sample new points from. I suggest a normal distribution with mean the current point and variance not too large and not too small (this is a parameter that you can change).
F. Implement the accept/reject step through using the prior and likelihood from A.
G. Check for mistakes and try out your code. Fix any errors as you find them.
H. Check the results you get against the true answers (found through methods 1 and 2). If they do not look the same, try to work out why and fix your code.

Once you have working code, you might experiment with the effect of changing:

i. the variance of the proposal q;
ii. the initial value of the chain.