

The following questions will require you to make use of several functions within the `nclbayes` package. As discussed in class, there are a few options for accessing these functions.

Option 1: Installing the `nclbayes` package

Do not copy-and-paste this code into R – you will have to type it in by hand very carefully!

```
1 install.packages("nclbayes", repos="http://R-Forge.R-project.org")
2 library(nclbayes)
```

Option 2: Accessing the functions from a webfile

Do not copy-and-paste this code into R – you will have to type it in by hand very carefully!

```
1 source("https://www.mas.ncl.ac.uk/~nlf8/teaching/mas3902/nclbayes.txt")
```

Option 3: Accessing the functions from Canvas

- In Canvas, click *Modules* and then *Main course files*
- Scroll down until you see a link for *R code from “nclbayes” package*
- Either download this file, saving it somewhere convenient, and then copy-and-paste all of its content into your RStudio session...
- ...Or just copy-and-paste directly into your RStudio session

Questions to try

Question One: The Gibbs Sampler

In this question you will use R to investigate the Gibbs sampler for Bayesian inference, as discussed in Section 4.4 of the lecture notes.

The increase in protein synthesis observed in athletes taking part in a specific muscle-building training programme is assumed Normally distributed with mean μ and standard deviation σ grams per day. After extensive discussions with sports scientists, the following independent priors are assumed:

$$\mu \sim N(2, 0.5), \quad \tau \sim Ga(3, 0.2).$$

- Why is a Gamma distribution appropriate as a prior for τ ?
- Write down the prior means, and prior standard deviations, for μ and τ .

3.3	2.3	2.2	3.1	2.2	2.2	2.8	3.0	2.1	3.3
2.0	3.7	1.9	2.2	2.4	2.8	1.9	2.2	2.2	2.6

Table 1: Increase in protein synthesis, in grams per day, for 20 athletes.

The scientists have collected data from a random sample of 20 participants who underwent this training program – their increase in protein synthesis, in grams per day, is shown in Table 1 above.

(c) Enter these data into R; for example:

```
1 protein = c(3.3, 2.3, 2.2, 3.1, ..., 2.6)
```

(d) Use R to find

$$\bar{x} = \frac{1}{20} \sum_{i=1}^{20} x_i \quad \text{and} \quad s^2 = \frac{1}{20} \sum_{i=1}^{20} (x_i - \bar{x})^2.$$

In Section 4.2 of the lecture notes, we found that the posterior density in this case is proportional to:

$$\pi(\mu, \tau | \mathbf{x}) \propto \tau^{g+\frac{n}{2}-1} \exp \left\{ -\frac{c}{2}(\mu - b)^2 - h\tau - \frac{n\tau}{2} [s^2 + (\bar{x} - \mu)^2] \right\},$$

where here we have $b = 2$, $c = 1/0.5$, $g = 3$ and $h = 0.2$, and $\bar{x}$ and s^2 are as calculated in part (d).

(e) Why do we need to use MCMC to explore the posterior distribution for $(\mu, \tau)^T$?

(f) Use results from Example 4.2 in the lecture notes to write down the Full Conditional Distribution (FCD) for $\mu | \tau, \mathbf{x}$ and $\tau | \mu, \mathbf{x}$.

We will now run a Gibbs sampler to make posterior inferences on μ and τ .

(g) Using the function `gibbs2`, run a Gibbs sampler for $N = 1,000$ iterations. Recall from lectures the syntax:

```
1 test = gibbs2(N, data.values=c(n,xbar,s), prior.params=c(b,c,g,h))
```

Where:

- **N** is the number of MCMC iterations
- The vector **data.values** take the sample size, sample mean, and standard deviation n , $\bar{x}$ and s (respectively)
- The vector **prior.params** take the prior parameters b , c , g and h from the priors for μ and τ

(h) Now use the following code to examine the output, and make decisions on the burn-in period and the thinning factor (you will find this code on the Course Canvas page, to copy-and-paste into RStudio):

```

1 par(mfrow = c(2, 3))
2 #Output for mu:
3 plot(ts(test[, 1]), xlab = "Iteration", ylab = "mu")
4 acf(test[, 1], main = "ACF plot for mu")
5 hist(test[, 1], main = "Histogram for mu", xlab = "mu", prob = TRUE)
6 lines(density(test[, 1]), col = "red", lwd = 2)
7
8 #Output for tau:
9 plot(ts(test[, 2]), xlab = "Iteration", ylab = "tau")
10 acf(test[, 2], main = "ACF plot for tau")
11 hist(test[, 2], main = "Histogram for tau", xlab = "tau", prob = TRUE)
12 lines(density(test[, 2]), col = "red", lwd = 2)

```

- (i) Re-run the Gibbs sampler for $N = 10K$ iterations once you have settled on an appropriate thinning factor, and (if necessary) thin the output before re-producing the plots in part (h). *Hint: If your autocorrelation function suggests a thinning factor of 5, the following code could be used:*

```

1 test2 = gibbs2(10000, ... ..)
2 ind = seq(1, 10000, 5)
3 mu.thin = test2[, 1][ind]
4 tau.thin = test2[, 2][ind]

```

- (j) Looking at the code for the function `gibbs2`, what starting values were used for μ and τ ?
- (k) Now you have (approximate) samples from the posteriors for μ and τ , complete the following table:

	Prior			Posterior		
	Mean	St. dev.	95% CI	Mean	St. dev.	95% CI
μ	2	0.707	(0.614, 3.386)			
τ	15	8.660	(1.518, 32.006)			
$\sigma = 1/\sqrt{\tau}$						

- How did we find the 95% confidence intervals for the priors for μ and τ ?
- Can you remember what distribution τ follows?
- Use your MCMC output to obtain the mean, standard deviation and 95% confidence intervals for the posteriors.

- (l) How have your beliefs about μ and τ changed in light of the data?

Question Two: The Gibbs Sampler (using nclbayes)

In Question One you used code that had been written “from first principles” to run the Gibbs sampler. The following block of code uses functions from the `nclbayes` package to do the same. Try out this code, making sure you replace the question marks ? with suitable values after inspecting your MCMC diagnostics.

```

1 #prior parameters:
2 b=?; c=?; g=?; h=?
3
4 #Run the Gibbs Sampler for the first time, and produce diagnostic plots:
5 posterior = gibbsNormal(N=10000, initial=c(b, g/h), priorparam=c(b, c, g,
6   h), n=?, xbar=?, s=?)
7 op = par(mfrow=c(2,2))
8 mcmcAnalysis(posterior, rows=2, show=FALSE)
9
10 #Re-run, thinning and removing burn-in as appropriate:
11 posterior2 = mcmcProcess(input = posterior, burnin=?, thin=?)
12 mcmcAnalysis(posterior2, rows=2, show=TRUE)
13 mcmcCi(posterior2, level=0.95)

```

Question Three: Metropolis-Hastings schemes

In this question you will use R to investigate the Metropolis-Hastings sampler for Bayesian inference, as discussed in Sections 4.5–4.6 of the lecture notes.

The size of meteorites landing to Earth is assumed Gamma-distributed with index parameter α and scale parameter λ . After discussions with some astrophysicists at Newcastle University, the following independent priors are assumed:

$$\alpha \sim Ga(2, 1), \quad \lambda \sim Ga(3, 1).$$

We collect data from a recent meteor shower, which gives the following summaries for a random sample of 50 meteors:

$$\bar{x} = 0.62, \quad \bar{x}_g = 0.46, \quad s = 0.4,$$

where $\bar{x}$, $\bar{x}_g$ and s are the sample mean, geometric mean and standard deviation (respectively).

- Use results from Example 4.3, and the information given in this question, to find the Full Conditional Distribution (FCD) for $\lambda|\alpha, \mathbf{x}$.
- Why will we need to use a Metropolis step in any MCMC scheme to construct a posterior sample for α ?

We will now run a Metropolis-within-Gibbs scheme to make posterior inferences on α and λ .

- Write down an appropriate starting value for α , $\alpha^{(0)}$.

(d) The following code will run a Metropolis-within-Gibbs scheme for this Gamma-Gamma-Gamma example, with a Normal random walk in the updating for α .

- Initially we will run the scheme for **N** = 20K iterations
- The argument **initval** should be set to the starting value you proposed in part (c)
- The argument **innov** represents the variance of the Normal random walk for α - this has been initially set to be 4 (but may need to change when you tune the algorithm)
- The vector **priorparam** is set at the values of the prior parameters $a = 2, b = 1, c = 3, d = 1$
- **xbar** and **xgbar** are set to the arithmetic and geometric means for the data (respectively), and **n** is the sample size

```
1 posterior = mwgGamma(N=20000, initial=?, innov=4, priorparam=c
  (2,1,3,1), n=50, xbar=0.62, xgbar=0.46, show=TRUE)
```

(e) Look at the output in the Console window. What is the average acceptance probability in the Metropolis steps used for α ? Should the value of **innov** be increased or decreased?

(f) Repeat the code in the above listing, but use trial-and-error to tune the algorithm (by experimenting with different values of **innov**), to obtain an average acceptance probability close to 23% (the optimal value, as discussed in lectures).

(g) Once you are happy with your value of **innov**, and the average acceptance probability it gives, run the following code:

```
1 posterior = mwgGamma(N=20000, initial=?, innov=?, priorparam=c
  (2,1,3,1), n=50, xbar=0.62, xgbar=0.46, show=TRUE)
2 mcmcAnalysis(posterior, rows=2, show=F)
```

(h) Using the diagnostic plots produced in part (g), what might be an appropriate burn-in value and thinning factor?

(i) The following code can be used to process the output from part (g), using the burn-in period and thinning factor you identified in part (h) (which should be entered in place of the question marks):

```
1 posterior2 = mcmcProcess(input = posterior, burnin = ?, thin = ?)
2 mcmcAnalysis(posterior2, rows=2)
```

(j) Complete the following table:

	Prior		Posterior	
	Mean	St. dev.	Mean	St. dev.
α	2	1.414		
λ	3	1.732		