# MATH48122 - Coursework 1

# Student ID: 10876957

## Question 1

Since $X \sim LN(\mu, \sigma^2)$, then $Y_i = log(X_i) \sim N(\mu, \sigma^2)$, i.e. $Y$ is normally distributed and the probability distribution function of each $y$ is equal to:

$$f_y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(y-\mu)^2)$$

.

Therefore, the likelihood function is the following:

$$L(\mu,\sigma^2|\mathbf{y}) = L(\mu,\sigma^2|y_1,...,y_n) = \prod_{i=1}^{n} f_y(y) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(y_i-\mu)^2}{2\sigma^2}) = (2\pi\sigma^2)^{-n/2} \cdot \exp(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2})$$

$$= (2\pi\sigma^2)^{-n/2} \cdot \exp(-\frac{\sum_{i=1}^{n} y_i^2 - 2\mu\sum_{i=1}^{n} y_i + n\mu^2}{2\sigma^2}) \tag{1}$$

Now, let us first find the distribution, which we will condition from:

$$\pi(\mu,\sigma^{-2},\lambda_0|\mathbf{y}) \propto \pi(\mathbf{y}|\mu,\sigma^{-2},\lambda_0)\cdot\pi(\mu,\sigma^{-2},\lambda_0) \propto \pi(\mathbf{y}|\mu,\sigma^2)\cdot\pi(\mu)\cdot\pi(\sigma^{-2}|\lambda_0)\cdot\pi(\lambda_0) \tag{2}$$

In order to find expression in 2, we need the prior distributions, which can be found as follows:

$$\pi(\mu) = \frac{1}{\sqrt{2\pi w_0}} \cdot \exp(\frac{-1}{2/w_0}(\mu-\alpha_0)^2) \propto \exp(\frac{-w_0(\mu-\alpha_0)^2}{2}) \tag{3}$$

$$\pi(\sigma^{-2}|\lambda_0) = \lambda_0 \cdot \exp(-\lambda_0\sigma^{-2}) \tag{4}$$

$$\pi(\lambda_0) \propto \lambda_0^{\beta_0-1} \cdot \exp(-\gamma_0\lambda_0) \tag{5}$$

Substituting 3, 4 and 5 into 2 gives:

$$\pi(\mu,\sigma^{-2},\lambda_0|\mathbf{y}) \propto (\sigma^2)^{-n/2} \cdot \exp(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}) \cdot \exp(\frac{-w_0(\mu-\alpha_0)^2}{2}) \cdot \lambda_0 \cdot \exp(-\lambda_0\sigma^{-2}) \cdot \lambda_0^{\beta_0-1} \cdot \exp(-\gamma_0\lambda_0) \tag{6}$$

Hence the conditional distributions are:

- $$\pi(\mu|\sigma^{-2},\lambda_0,\mathbf{y}) \propto \exp(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}) \exp(\frac{-w_0(\mu-\alpha_0)^2}{2}) \propto \exp(-\frac{1}{2}((\frac{n}{\sigma^2}+w_0)\cdot\mu^2 - 2(\frac{\sum_{i=1}^{n} y_i}{\sigma^2}+w_0\alpha_0)\cdot\mu)) \tag{7}$$

  If $A = \frac{n}{\sigma^2} + w_0$ and $B = \frac{\sum_{i=1}^{n} y_i}{\sigma^2} + w_0\alpha_0$, then $\mu \sim N(B/A, 1/A)$.

- $$\pi(\sigma^{-2}|\mu,\lambda_0,\mathbf{y}) \propto (\sigma^2)^{-n/2}\cdot\exp\Big(-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\Big)\cdot\exp(-\lambda_0\sigma^{-2}) \propto (\sigma^{-2})^{n/2}\cdot\exp\Big(-\Big(\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2}+\lambda_0\Big)\cdot\sigma^{-2}\Big) \quad (8)$$

  If $A=\frac{n}{2}$ and $B=\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2}+\lambda_0$, then $\sigma^{-2}\sim Gamma(A+1,B)$ and $\sigma^2 \sim Inverse-Gamma(A+1,B)$.

- $$\pi(\lambda_0|\mu,\sigma^{-2},\mathbf{y}) \propto \lambda_0\cdot\exp(-\lambda_0\sigma^{-2})\cdot\lambda_0^{\beta_0-1}\cdot\exp(-\gamma_0\lambda_0) \propto (\lambda_0)^{\beta_0}\cdot\exp(-(\sigma^{-2}+\gamma_0)\cdot\lambda_0) \quad (9)$$

  Then $\lambda_0 \sim Gamma(\beta_0+1,\sigma^{-2}+\gamma_0)$.

# Question 2

Now, let us use the code shown in Figure 1 to obtain samples from the joint distribution of $(\mu,\sigma^2)$ using the Gibbs sampler algorithm. Note that the vector in which the priors are stored is: priors=$(\beta_0,\omega_0,\alpha_0,\gamma_0)$.

```
1 #priors=(beta0, w0, alpha0,gamma0)
2 sgibbs=function(y,mu,sigmasq,lambda0,priors,run) #creating the function containing of our data, parameters, hyperparameters, and number of times we wish to run the algorithm
3 {
4 output=matrix(0,ncol=2,nrow=run) #creating a matrix with 2 columns, where the first stores all the outputs for mu and the second - these for sigma squared
5 n=length(y) #n is the number of data observations
6 for(i in 1:run) #run the for loop for the number of iterations we choose
7 {
8 mu=rnorm(1, ((sum(y))/(sigmasq)+(priors[3]*priors[2]))/((n/(sigmasq))+priors[2]), sqrt(1/((n/sigmasq)+priors[2])) ) #generating a random sample for mu drawn from the normal distribution with the posterior mean and variance
9 a=n/2+1 #posterior mean for sigma squared
10 b=(1/2)*(sum((y-mu)**2))+lambda0 #posterior variance for sigma sqared
11 sigmasq=1/rgamma(1,a,b) #creating a random sample for sigma squared drawn from the Inverse-Gamma distribution
12 lambda0=rgamma(1,priors[1]+1, 1/sigmasq+priors[4]) #generating a random sample for lamda0 drawn from the Gamma distribution with its posterior mean and variance
13 output[i,]=c(mu,sigmasq) #output the results for the chosen number of iterations
14 }
15 output
16 }
```

Figure 1: R code for using Gibbs sampler to obtain samples from the joint distribution of $(\mu,\sigma^2)$

**Code implementation**: On line 2 we define our function for performing the Gibbs sampler algorithm. Then on lines $8, 11$ and $12$ we implement the distributions we got in equations 7, 8 and 9 respectively.

# Question 3

Apply now the Gibbs sampler algorithm to the particular data we have. This is done in Figure 2.

```
x=scan("Survival.txt") #read the data
y<-log(x) #create our y vector as the log of the data stored in x
sm=mean(y) #sample mean for y

priors=c(1,1,0,1) #set initial values for the hyperparameters
q=sgibbs(y,sm,1,1,priors,1100) #as initial values for mu, sigma squared and lambda0 we chose
    the sample mean, 1 and 1 respectively and repeat the algorithm 1100 times
q=q[101:1100,] #discard the first 100 iterations as burn-in

psi <- exp(q[,1] + q[,2]/2) #find the values of psi for all 1100 iterations
mean(psi) #estimate of psi
sd(psi) #standard deviation of psi

mu<-(q[,1])
mean(mu) #estimate of mu
sd(mu) #standard deviation of mu

sigmasq<-(q[,2])
mean(sigmasq) #estimate of sigma squared
sd(sigmasq) #standard deviation of sigma squared
```

Figure 2: R code for applying Gibbs sampler to the data

```
> mean(psi) #estimate of psi
[1] 18.01649
> sd(psi) #standard deviation of psi
[1] 1.610966
> mean(mu) #estimate of mu
[1] 2.564398
> sd(mu) #standard deviation of mu
[1] 0.07688195
> sigmasq<-(q[,2])
> mean(sigmasq) #estimate of sigma squared
[1] 0.6458683
> sd(sigmasq) #standard deviation of sigma squared
[1] 0.09135508
```

Figure 3: R code output for applying Gibbs sampler to the data

From the output presented in Figure 3, it can be seen that an estimate for $\psi$ in this case is 18.01649, which suggests that, on average, a female breast cancer patient can expect to survive for approximately 18 years after undergoing mastectomy. The standard deviation is 1.610966 and it suggests that the estimated expected lifetime values deviate from the mean by approximately 2 years. Therefore, we can conclude that the estimate is quite good.

Now let us analyse the trace plot to check if the MCMC algorithm has effectively sampled from the posterior distribution. The above will be done by the following code:

```r
plot.ts(q)
```

Figure 4: R code for creating a trace plot

The graph we get is then shown in Figure 5 below. The top series is $\mu$ and the bottom series is $\sigma^2$. It can be observed that there is convergence and the trace plots oscillate about the estimates of $\mu$ and $\sigma^2$ shown on lines 6 and 11 in Figure 3. Therefore, we conclude that the MCMC algorithm is effectively sampling from the posterior distribution without getting stuck in any particular region.

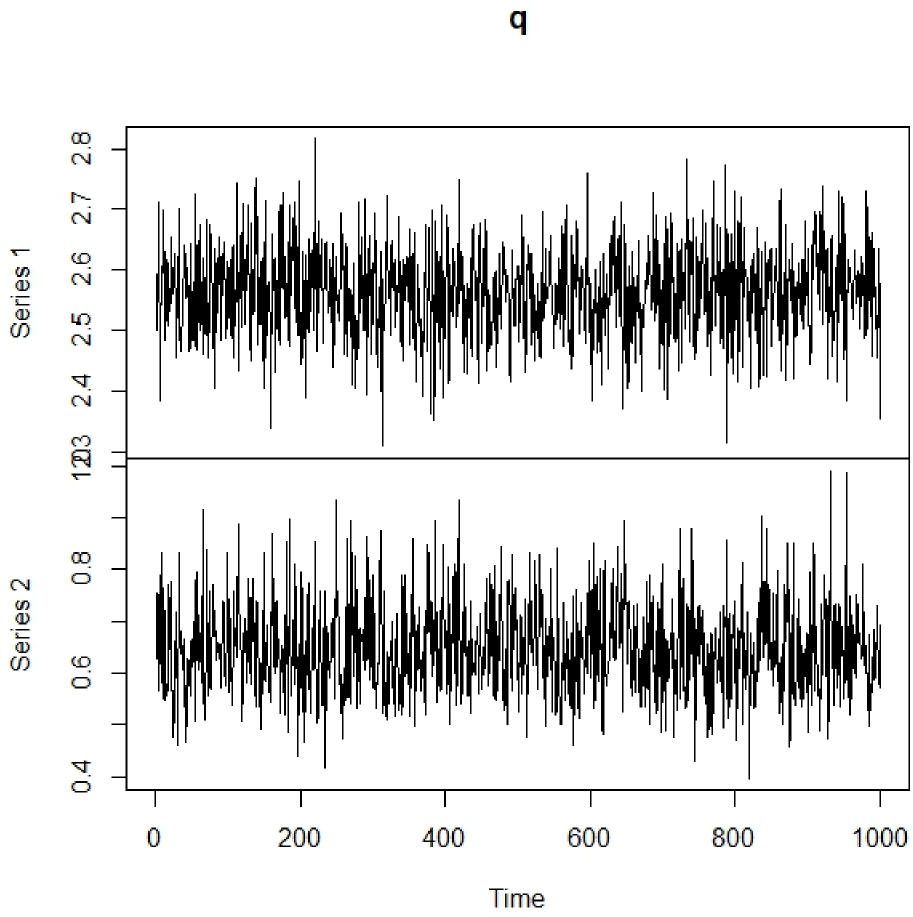


Figure 5: Trace plot

In order to check if the algorithm is reliable, it could be also helpful to plot the posterior densities for the two parameters using the code in Figure 6.

```r
plot(density(q[,1]),xlab=expression(mu),ylab="Posterior Density",main="")
plot(density(q[,2]),xlab=expression(sigmasq),ylab="Posterior Density",main="")
```

Figure 6: R code for plotting the posterior densities

In Figure 7 we observe that the posterior distribution for $\mu$ is almost symmetric. Hence, the posterior mode and the mean are quite similar. This indicates an adequate model and subsequently, a good estimator, since there is no obvious skewness towards lower or higher values. In 8 the distribution for $\sigma^2$ is slightly right-skewed, indicating that the posterior mode is slightly larger than the mean. However, we can still conclude that our estimators are plausible.

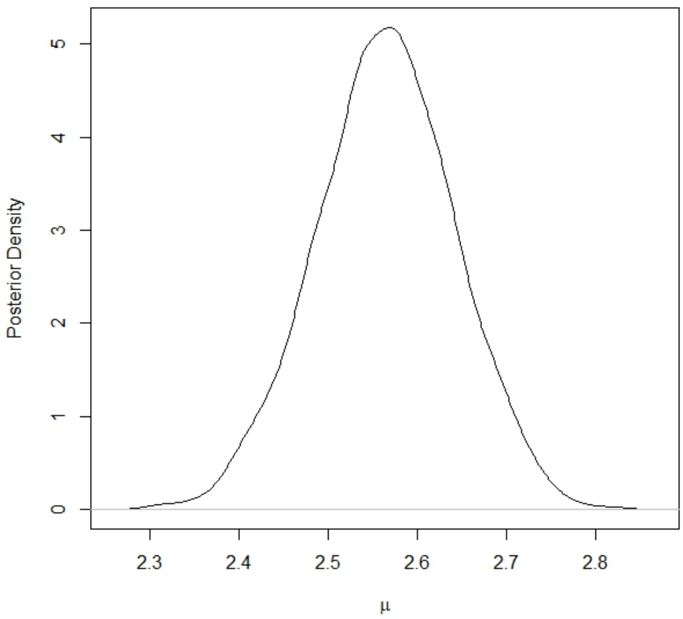


Figure 7: Posterior density for $\mu$

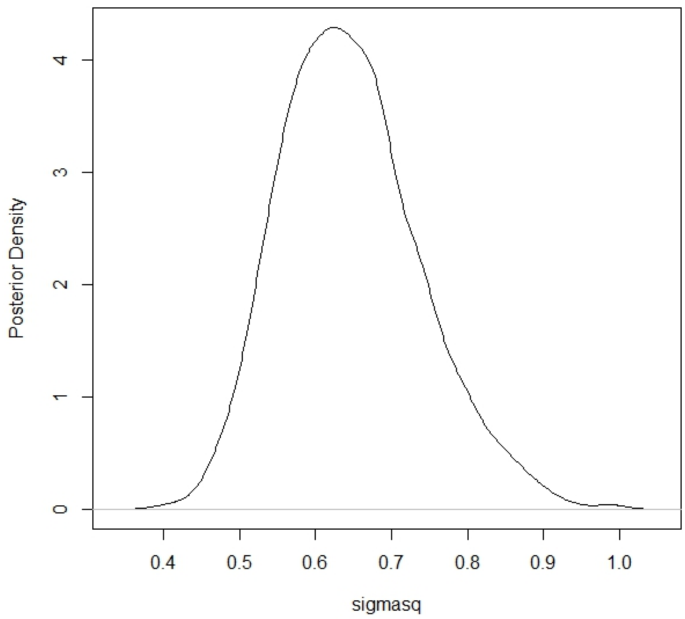


Figure 8: Posterior density for $\sigma^2$

# Question 4

It is given in the problem that the augmented likelihood function is the one shown in 10:

$$L(\mu, \sigma^2|\mathbf{w}^*) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(w_i - \mu)^2}{2\sigma^2}) = (2\pi\sigma^2)^{-n/2} \cdot \exp(-\frac{\sum_{i=1}^{n}(w_i - \mu)^2}{2\sigma^2}) \tag{10}$$

Hence, 10 has the form of a normal distribution and the problem becomes similar to Question 1 and the conditional distribution for $\mu$, $\sigma^2$ and $\lambda_0$ are presented in 11, 12 and 13 respectively:

$$\pi(\mu|\sigma^2, \lambda_0, \mathbf{w}^*) \propto \exp(-\frac{\sum_{i=1}^{n}(w_i - \mu)^2}{2\sigma^2}) \exp(\frac{-w_0(\mu - \alpha_0)^2}{2}) \propto \exp(-\frac{1}{2}((\frac{n}{\sigma^2}+w_0)\cdot\mu^2 - 2(\frac{\sum_{i=1}^{n} w_i}{\sigma^2}+w_0\alpha_0)\cdot\mu)) \tag{11}$$

If $A = \frac{n}{\sigma^2} + w_0$ and $B = \frac{\sum_{i=1}^{n} w_i}{\sigma^2} + w_0\alpha_0$, then $\mu \sim N(B/A, 1/A)$.

$$\pi(\sigma^{-2}|\mu, \lambda_0, \mathbf{w}^*) \propto (\sigma^2)^{-n/2} \cdot \exp(-\frac{\sum_{i=1}^{n}(w_i - \mu)^2}{2\sigma^2}) \cdot \exp(-\lambda_0\sigma^{-2}) \propto (\sigma^{-2})^{n/2} \cdot \exp(-(\frac{\sum_{i=1}^{n}(w_i - \mu)^2}{2} + \lambda_0) \cdot \sigma^{-2}) \tag{12}$$

If $A = \frac{n}{2}$ and $B = \frac{\sum_{i=1}^{n}(w_i-\mu)^2}{2} + \lambda_0$, then $\sigma^{-2} \sim Gamma(A+1, B)$ and $\sigma^2 \sim Inverse-Gamma(A+1, B)$.

$$\pi(\lambda_0|\mu, \sigma^2, \mathbf{y}) \propto \lambda_0 \cdot \exp(-\lambda_0\sigma^{-2}) \cdot \lambda_0^{\beta_0 - 1} \cdot \exp(-\gamma_0\lambda_0) \propto (\lambda_0)^{\beta_0} \cdot \exp(-(\sigma^{-2} + \gamma_0) \cdot \lambda_0) \tag{13}$$

Then $\lambda_0 \sim Gamma(\beta_0 + 1, \sigma^{-2} + \gamma_0)$.

# Question 5

First note that $z_i^{\prime}s$ are independent. For sampling $z_i$'s we need to determine what distribution they have. Looking at our likelihood function in 10, we observe that it is proportional to a Normal density. But note that we are looking for a conditional distribution, so now for $z_i$'s we have a Normal distribution, conditioned for values greater than the censored value. Since $y_i = log(x_i)$, we used $log(c) = log(25)$ as our censored value. The expression we get as a result is derived in 14:

$$\pi(z_i|\mu,\sigma^2,\mathbf{y},\mathbf{z}_{i-}) = \pi(z_i|\mu,\sigma^2,\mathbf{y}_i) \propto f(z_i|\mu,\sigma^2)I(z_i > log(c)) \propto \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(-\frac{(z_i-\mu)^2}{2\sigma^2})I(z_i > log(25)) \tag{14}$$

, where $\mu$ and $\sigma^2$ are the posterior mean and variance with the distributions found in 11 and 12. Therefore, we have a Truncated Normal distribution with the following parameters:

$$z_i|\mu,\sigma^2,\mathbf{y},\mathbf{z}_{i-} \sim TruncNormal(\mu,\sigma^2,log(c)) \tag{15}$$

# Question 6

Note that the vector in which the priors are stored is: priors=$(\omega_0,\alpha_0,\beta_0,\gamma_0)$.

```
#priors=(w0,alpha0,beta0,gamma0)
library(truncnorm)
surv=function(y,n,mu,sigmasq,lambda0,priors,size,c)
{
output=matrix(0,ncol=3,nrow=size) #creating a matrix with 3 columns in which we will store the
    output results for mu, sigma^2 and lambda_0
n=length(y) #set n to be number of data observations
z=y[y==log(c)] #initialize vector z in which we will store the missing values
p<- y[y != log(c)] #define vector p with all the values different from the censored one
m=n-length(p) #number of missing values
w=c(p,z) #define vector w which contains y_1 to y_m and z_(m+1) to z_n
for(i in 1:size) #for loop for the number of iterations
{
#for loop for sampling the missing values with the Truncated Normal distribution we found in
    question 5 and updating vector w
for(k in 1:m)
{
z[k]=rtruncnorm(1, a = log(c), b = Inf, mean = mu, sd = sqrt(sigmasq))
w=c(p,z)
}
#Creating rand samples for mu, sigma^2 and lambda0 according to their conditional distributions
A=n/sigmasq+priors[1]
B=sum(w)/sigmasq+priors[2]*priors[1]
mu=rnorm(1,B/A,1/A)

sigmasq=1/rgamma(1,n/2+1,(1/2)*(sum((w-mu)**2))+lambda0)

lambda0=rgamma(1,priors[3]+1, 1/sigmasq+priors[4])
output[i,]=c(mu,sigmasq,lambda0)
}
output
}
```

Figure 9: R code for using augmented Gibbs sampler to obtain samples from the joint distribution of $(\mu,\sigma^2)$

**Code implementation**: First, we need to implement the Gibbs sampler algorithm by defining a function containing the data vector with its length, the parameters, the hyperparameters, the number of iterations and in this case we need the censored value $/c = 25/$ too. This is done on line 3 in Figure 9. The idea is to create a vector $w$, consisting of the observed values in the data vector $y$ and also of the missing values, i.e. the ones equal to $log(25)$ in $y$, which we will store in another vector $z$. This is done on lines 7 to 10. From equations 14 and 15, we found that we can generate random samples for the missing values in $z$ drawn from the Truncated Normal distribution and we do that in a for loop on lines 14 to 18. Then we proceed in the same way as we did in Question 2.

# Question 7

```
priors=c(1,0,1,1) #setting the priors
x=scan("Survivalcensored.txt") #read the data
y<-log(x) #creating the data vector we will use
n=length(y) #set n to be number of data observations
sm=mean(y) #mean of vector y

q=surv(y,n,sm,1,1,priors,1100,25) #apply the augmented gibbs sampler function from above
q=q[101:1100,] # Discard the first 100 iterations as burn-in

mu=q[,1]
sigmasq=q[,2]
mean(mu) #estimate of mu
sd(mu) #standard deviation of mu
mean(sigmasq) #estimate of sigma^2
sd(sigmasq) #standard deviation of sigma^2

psi <- exp(q[,1] + q[,2]/2) #defining psi
mean(psi) #estimate of psi
sd(psi) #standard deviation of psi
```

Figure 10: R code for applying augmented Gibbs sampler to our data

```
> mean(mu) #estimate of mu
[1] 2.576436
> sd(mu) #standard deviation of mu
[1] 0.02234109
> mean(sigmasq) #estimate of sigma^2
[1] 0.6796824
> sd(sigmasq) #standard deviation of sigma^2
[1] 0.1178167
>
> psi <- exp(q[,1] + q[,2]/2) #defining psi
> mean(psi) #estimate of psi
[1] 18.52226
> sd(psi) #standard deviation of psi
[1] 1.379679
```

Figure 11: R code output after applying the augmented Gibbs sampler function to the data

We applied the censored data we have to our algorithm by the code shown in Figure 10. The output we get is presented in Figure 11. As we can observe on lines 12 and 14 in Figure 11, the estimate for $\psi$ in this case is $\psi = 18.52226$. This suggests that a female breast cancer patient can expect to survive for approximately 19 years after undergoing mastectomy. The standard deviation here is 1.379679, so the patients can assume that the expected lifetime may vary for approximately 1 year. Comparing these results with the ones in Figure 3, we can see there is a small difference of approximately 1 year in the results. The estimates obtained from the augmented data seem to be higher, whereas their standard deviations look lower. In our case, imputing missing values introduces bias, leading to higher estimates of parameters. It also reduces the variability of estimates, which leads to lower standard deviations.

Now, let us analyse the efficiency of the algorithm. As before, we will do the trace plots and the posterior densities seen in Figures 12, 13 and 14.

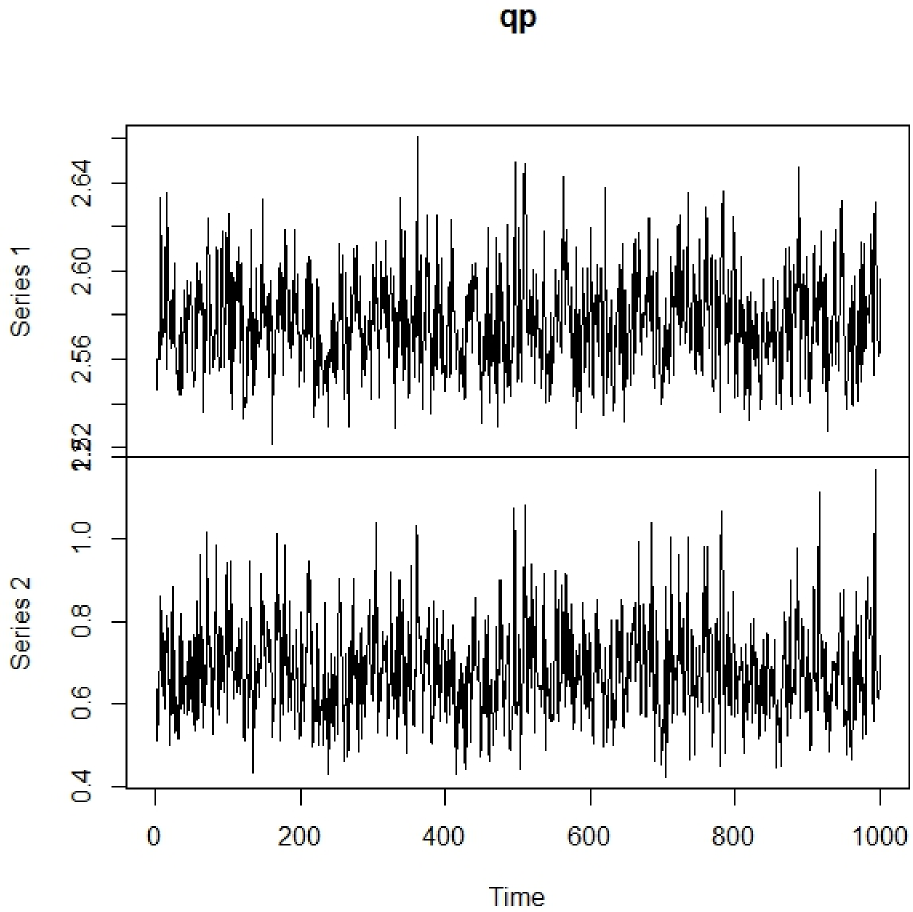


Figure 12: Trace plot

Again, in Figure 12 we have that the top series is $\mu$ and the bottom series is $\sigma^2$. The trace plots show convergence and they oscillate about the estimates of $\mu$ and $\sigma^2$ obtained on lines 2 /= 2.576436/ and 6 /= 0.6796824/ in Figure 11 respectively. Therefore, even with missing data, the MCMC algorithm handles the missing values effectively and converges to the target distribution.

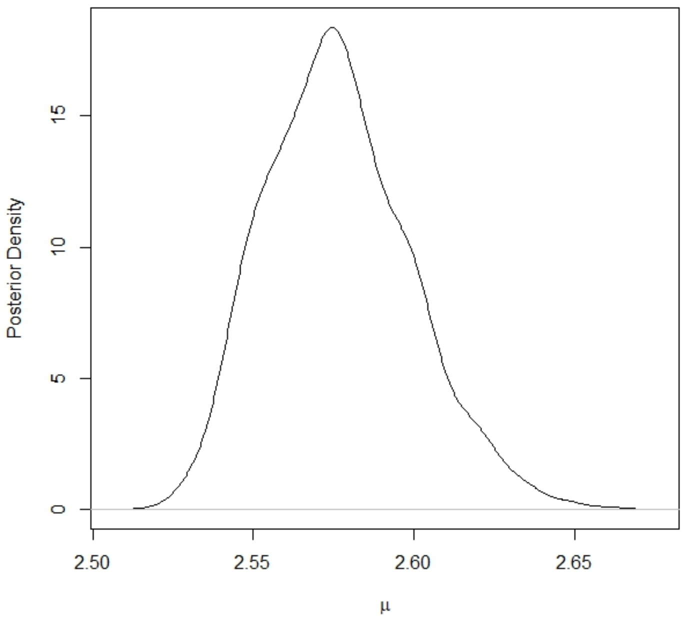


Figure 13: Posterior density for $\mu$

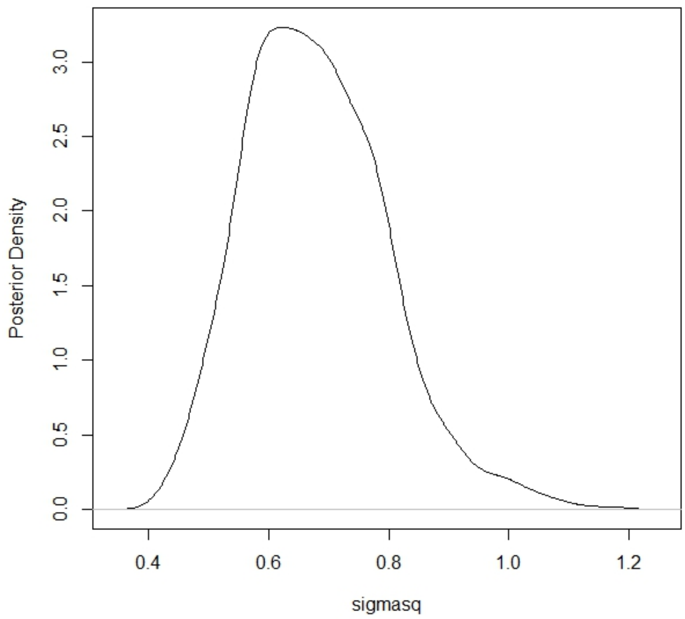


Figure 14: Posterior density for $\sigma^2$

From Figures 13 and 14 we observe slightly right-skewed distributions so the two means are a bit larger than the posterior mode, where this difference is bigger for $\sigma^2$.