

# SDS 383C - Statistical Modeling 1: Homework 5

Rahul Nandakumar

Graduate Student, ORIE Program (E-ID: rn9355)

December 3, 2022

1.

**Solution:**

We are given the true posterior of a parameter  $\theta$  as  $p(\theta \mid \text{data}) \equiv t_3(\theta)$ .

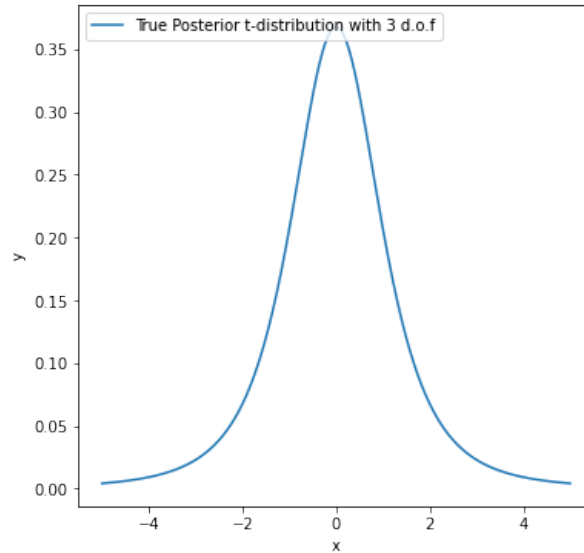


Figure 1: True Posterior t-distribution with 3 d.o.f

*a.* We now implement importance resampling in Python with the normal distribution as the reference density. Here, our density of interest is the t-distribution with 3 degrees of freedom which is hard to sample from. We have the standard normal distribution which is easy to sample from. We calculate the weights of the sample as the ratio between the pdf of the t distribution with 3 d.o.f and the standard normal distribution with each sample. After normalizing these weights, we resample from the original sample with probabilities given by the normalized weights. This process is importance resampling. The Python code is attached with this assignment and the results are displayed below. We carry out this resampling process for a 100 and 10000 samples in 2 separate simulations. Our observation is that as the number of samples increases, we are able to better sample from the posterior distribution as evident from the fitting of the true posterior with the histogram.

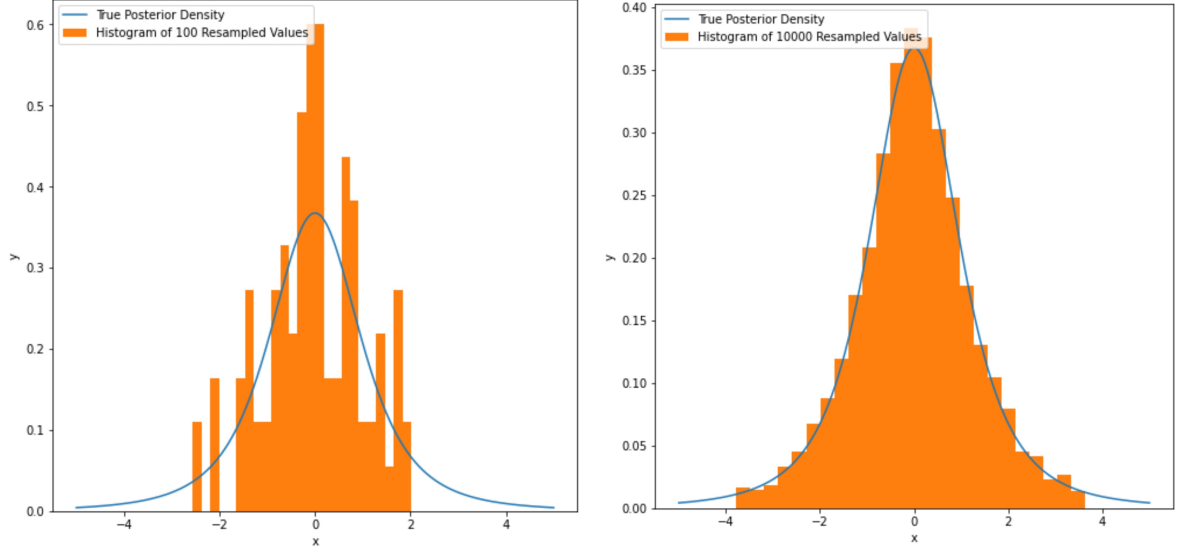


Figure 2: Importance resampling for 100 and 10000 samples

*b.* Importance sampling is similar in implementation to importance resampling, except here we don't resample from the original sample using the normalized weights as the probabilities of picking each sample. Instead, importance sampling can be used to calculate other useful parameters such as the mean and the variance of a given sample, which can be used to estimate the true mean and variance of the distribution. Our implementation is the same as in importance resampling, where we find the weights as the ratio between the pdf of the  $t$  distribution with 3 d.o.f and the standard normal distribution with each sample. We then normalize these weights. Now, the mean of the samples is calculated as the sum-product of the weights and samples, and the variance is calculated as the sum-product of the weights and the square of the difference between the sample value and the sample calculated previously. This is implemented in Python and the results are as follows. The estimated mean for 100 samples is evaluated as 0.3838500475232623, and the estimated variance for 100 samples is estimated as 1.9797163514580358. The estimated mean for 10000 samples evaluates to 0.012160083789871339 and the estimated variance for 10000 evaluates to 2.842751834188354. The true mean of the  $t$ -distribution is 0 and the true variance of the  $t$ -distribution with  $v$  d.o.f is given by  $\frac{v}{v-2} (= \frac{3}{1} = 3$  in our case). We observe as the number of samples increase, we are able to better predict the true mean and variance.

2.

### ***Solution***

We are given a bivariate normal distribution with the following parameters.

$$\begin{aligned}(\mu_1, \mu_2) &= (0, 2) \\ (\sigma_1^2, \sigma_2^2) &= (1, 1) \\ \rho &= 0.75\end{aligned}$$

We are asked to construct a Gibbs Sampler that draws samples from this bivariate normal

distribution. Using the value of  $\rho$ , we can construct the covariance matrix using the relation,

$$\rho = \frac{Cov(X, Y)}{\sqrt{var(X) \times var(Y)}}$$

$$Cov(X, Y) = 0.75$$

Therefore,

$$\Sigma = \begin{bmatrix} var(X) & Cov(X, Y) \\ Cov(X, Y) & var(Y) \end{bmatrix} = \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

Using this covariance matrix and the means, we implement the Gibbs Sampler. The implementation has been carried out in R for the reason to obtain the autocorrelation plots and the contour plots with ease. We carry out the sampling for 1000 burned-in and thinned samples, The results are as follows, and the R file is attached along with this assignment.

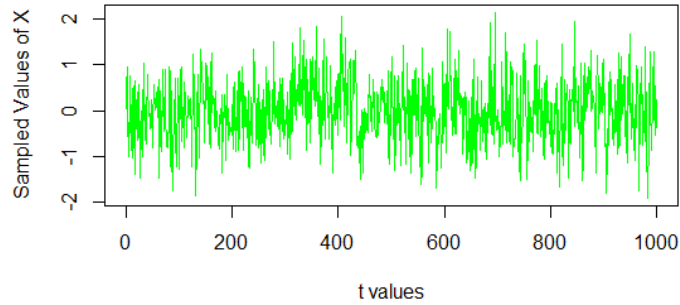


Figure 3: Traceplot Gibbs Sampled values of X

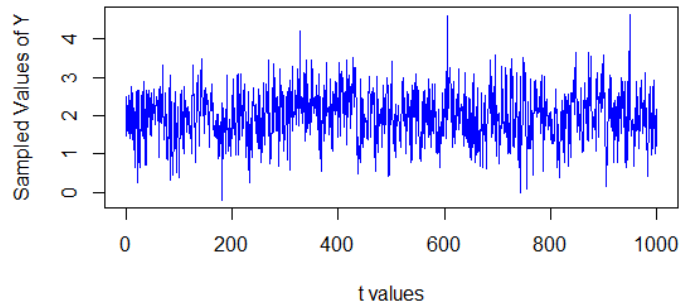


Figure 4: Traceplot Gibbs Sampled values of Y

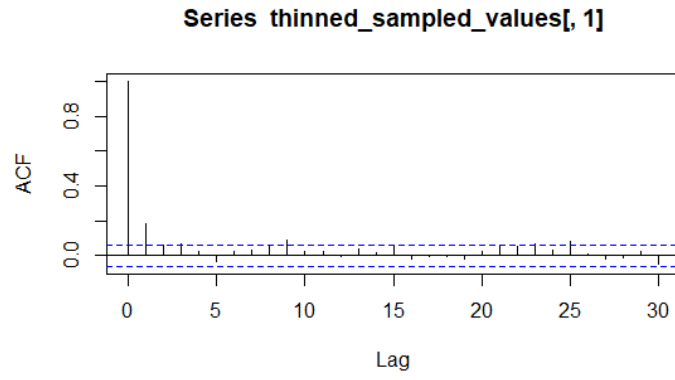


Figure 5: Autocorrelation plot for X

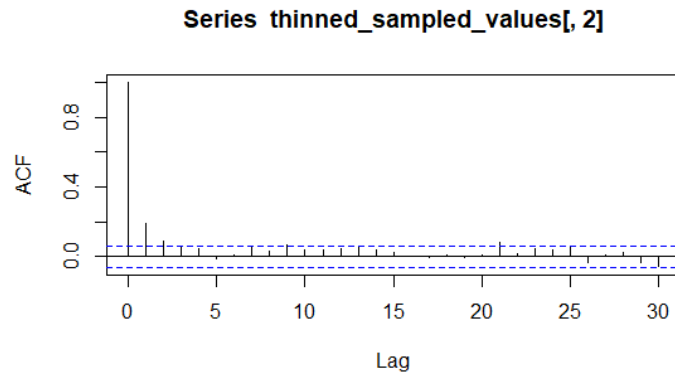


Figure 6: Autocorrelation Plot for Y

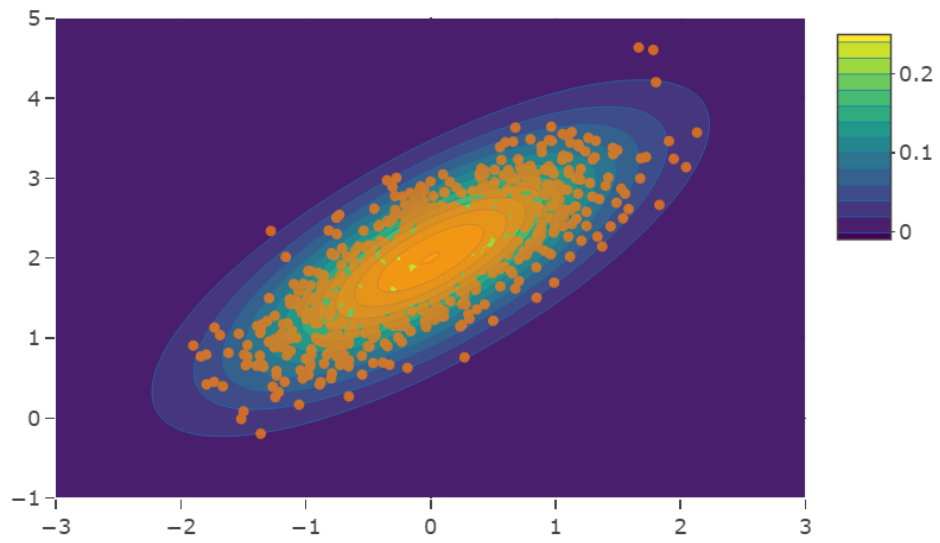


Figure 7: Contour Plot of actual density superimposed over sampled values

3.

**Solution:**

We are given an AR(1) process with mean  $\mu$ , correlation  $|\rho| < 1$  and error variance  $\sigma^2$  which evolves as follows.

$$y_t = \mu + \rho(y_{t-1} - \mu) + \epsilon_t$$

Where  $\epsilon_t$  iid with  $\mathbb{E}(\epsilon_t) = 0$  and  $\text{var}(\epsilon_t) = \sigma^2$ . Here, we understand that  $\mathbb{E}(y_t) = \mu$

a. We know that

$$\begin{aligned} \text{var}(y_t) &= \mathbb{E}(y_t^2) - \mathbb{E}(y_t)^2 \\ &= \mathbb{E}(y_t(\mu + \rho(y_{t-1} - \mu) + \epsilon_t)) - \mu^2 \\ &= \mathbb{E}(y_t(\mu(1 - \rho))) + \rho\mathbb{E}(y_t y_{t-1}) + \mathbb{E}(y_t \epsilon_t) - \mu^2 \\ &= (\mu(1 - \rho))\mathbb{E}(y_t) + \rho\mathbb{E}(y_t y_{t-1}) + \mathbb{E}(y_t \epsilon_t) - \mu^2 \\ &= (\mu(1 - \rho))\mathbb{E}(y_t) + \rho(\text{Cov}(y_t, y_{t-1}) + \mathbb{E}(y_t)\mathbb{E}(y_{t-1})) + \mathbb{E}(y_t \epsilon_t) - \mu^2 \\ &= \rho^2 \text{var}(y_t) + \sigma^2 \\ \text{var}(y_t)(1 - \rho^2) &= \sigma^2 \\ \text{var}(y_t) &= \frac{\sigma^2}{(1 - \rho^2)} \end{aligned}$$

b. Let us consider

$$\begin{aligned} y_t &= \mu + \rho(y_{t-1} - \mu) + \epsilon_t \\ y_t &= \mu(1 - \rho) + \rho(y_{t-1}) + \epsilon_t \\ y_t &= \rho(y_{t-1}) + A_t; \text{ Let } \mu(1 - \rho) + \epsilon_t = A_t \\ y_{t-k}y_t &= \rho(y_{t-1})y_{t-k} + A_t y_{t-k}; \text{ Multiplying both sides by } y_{t-k} \\ \mathbb{E}(y_{t-k}y_t) &= \rho\mathbb{E}((y_{t-1})y_{t-k}) + \mathbb{E}(A_t y_{t-k}); \text{ Taking expectation of this expression} \end{aligned}$$

Let us denote the covariance of  $y_t$  and  $y_{t-k}$  by  $\gamma_k$ . Now,

$$\begin{aligned} \gamma_k &= \rho\gamma_{k-1} \\ \gamma_k &= \rho^k \gamma_0; \text{ Going recursively to } \gamma_0 \text{ from } \gamma_{k-1} \end{aligned}$$

From this relation, we can say that

$$\begin{aligned} \text{corr}(y_t, y_{t-k}) &= \frac{\gamma_k}{\text{var}(y_t)\text{var}(y_{t-k})} \\ &= \frac{\rho^k \gamma_0}{\text{var}(y_t)\text{var}(y_{t-k})} \\ &= \rho^k \end{aligned}$$

c.

We know that,

$$y_t = \mu + \rho(y_{t-1} - \mu) + \epsilon_t$$

Now,

$$\begin{aligned}
\mathbb{E}(\bar{y}_n) &= \mathbb{E}\left(\frac{y_1 + y_2 + \dots + y_n}{n}\right) \\
&= \frac{1}{n}(\mathbb{E}(y_1) + \dots + \mathbb{E}(y_n)) \\
&= \frac{1}{n}(n\mu) \\
&= \mu
\end{aligned}$$

d.

e. We know from the previous result that

$$\begin{aligned}
\text{var}(\bar{y}_n) &= \frac{\sigma^2}{n^2(1-\rho^2)} \left( n + 2 \sum_{j=1}^{n-1} (n-j)\rho^j \right) \\
n\text{var}(\bar{y}_n) &= \frac{\sigma^2}{n(1-\rho^2)} \left( n + 2 \sum_{j=1}^{n-1} (n-j)\rho^j \right); \text{ Multiplying both sides with } n \\
&= \frac{\sigma^2}{(1-\rho^2)} \left( 1 + 2 \sum_{j=1}^{n-1} \left(1 - \frac{j}{n}\right) \rho^j \right) \\
&= \frac{\sigma^2}{(1-\rho^2)} \left( 1 + 2(\rho + \rho^2 + \dots + \infty) \right); \text{ Since } n \rightarrow \infty, \frac{1}{n} \rightarrow 0 \\
&= \frac{\sigma^2}{(1-\rho^2)} \left( 1 + 2 \left( \frac{\rho}{1-\rho} \right) \right); \text{ Sum of infinite G.P.} \\
&= \frac{\sigma^2}{(1-\rho^2)} \left( \frac{1+\rho}{1-\rho} \right) \\
n\text{var}(\bar{y}_n) &= \frac{\sigma^2}{(1-\rho)^2}
\end{aligned}$$

f.

Now,

$$\begin{aligned}
\mathbb{E}(s_n^2) &= \mathbb{E}\left(\frac{n-1}{n}s^2\right) \\
&= \mathbb{E}\left(1 - \frac{1}{n}s^2\right) \\
&= \left(1 - \frac{1}{n}\right)\mathbb{E}(s^2)
\end{aligned}$$

As  $n \rightarrow \infty$ ,  $\frac{1}{n} \rightarrow 0$  and  $\mathbb{E}(s^2) \rightarrow \text{var}(y_t)$ . Therefore, we can conclude that as  $n \rightarrow \infty$

$$\mathbb{E}(s_n^2) \rightarrow \text{var}(y_t)$$

4.

We are given a AR(1) process and we are asked to simulate 500 points from this process for analysis. The process has mean  $\mu = 0$ , correlation  $\rho = 0.75$  and Normally distributed errors with  $\sigma^2 = 1$ . Now, we fit a Bayesian Model to this data points.

a. Our given process is,

$$y_t = \rho y_{t-1} + e_t$$

This implies that our model can also be written as,

$$f(y_t \mid y_{t-1}, \rho, \sigma^2) \sim \text{Normal}(\rho y_{t-1}, \sigma^2)$$

The likelihood function for this model can be written as

$$\begin{aligned} p(y_{1:n} \mid \mu, \sigma^2) &= \prod_{i=2}^t f(y_i \mid y_{i-1}, \rho, \sigma^2) \\ &= \prod_{i=2}^t \text{Normal}(\rho y_{i-1}, \sigma^2) \\ &= (2\pi\sigma^2)^{-\frac{(n-1)}{2}} \exp\left(\frac{-1}{2\sigma^2} \sum_{i=2}^t (y_i - \rho y_{i-1})^2\right) \end{aligned}$$

From this likelihood, we can estimate the priors of our model as having a conjugate NIG prior. Mathematically, it is expressed as,

$$\begin{aligned} p(\rho \mid \sigma^2) &= \text{Normal}\left(\mu_0, \frac{\sigma^2}{k_0}\right) \\ p(\sigma^2) &= \text{Inverse-Gamma}\left(\frac{v_0}{2}, \frac{v_0\sigma_0^2}{2}\right) \end{aligned}$$

b. Under these conjugate priors, we can obtain the full conditionals as follows. First, we evaluate the posterior.

$$\begin{aligned} p(\rho, \sigma^2 \mid y_{1:n}) &= p(\rho, \sigma^2) \times p(y_{1:n} \mid \rho, \sigma^2) \\ &= (2\pi\sigma^2)^{-\frac{(n-1)}{2}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=2}^n (y_t - \rho y_{t-1})^2\right) \times p(\rho \mid \sigma^2) \times p(\sigma^2) \\ &= (2\pi\sigma^2)^{-\left(\frac{v_0+1}{2}\right)-1} \exp\left(\frac{-1}{2\sigma^2} (v_0\sigma_0^2 + \kappa_0\rho^2)\right) \times (2\pi\sigma^2)^{-\frac{(n-1)}{2}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=2}^n (y_t - \rho y_{t-1})^2\right) \\ &\propto (\sigma^2)^{-\frac{(v_0+n)}{2}-1} \exp\frac{-1}{2\sigma^2} \left(v_0\sigma_0^2 + (\rho - \rho_n)^2\kappa_n - \kappa_n\rho_n^2 + \sum_{t=2}^n y_t^2\right) \end{aligned}$$

This is the kernel of an NIG Distribution with parameters  $(\rho_n, \kappa_n, v_n, \sigma_n^2)$ . Here, we have the hyperparameters equal to

$$\begin{aligned} \kappa_n &= \kappa_0 + \sum_{t=2}^n y_{t-1}^2 \\ \rho_n &= \frac{1}{\kappa_n} \left( \sum_{t=2}^n y_t y_{t-1} \right) \\ \sigma_n^2 &= \frac{1}{2} \left( v_0\sigma_0^2 - \kappa_n\rho_n^2 + \sum_{t=2}^n y_t^2 \right) \\ v_n &= v_0 + n - 1 \end{aligned}$$

From this kernel, we can obtain the full conditionals for  $\rho$  and  $\sigma^2$  by assuming either one of the parameters are given. This gives us,

$$p(\rho \mid \sigma^2, y_{1:n}) = \text{Normal}\left(\rho, \frac{\sigma^2}{\kappa_n}\right)$$

$$p(\sigma^2 \mid \rho, y_{1:n}) = \text{Inverse-Gamma}\left(\frac{v_n}{2}, \frac{2\sigma_n^2 + \kappa_n(\rho - \rho_n)^2}{2}\right)$$

c. Now, we implement the Gibbs sampler in R by using these full conditionals. First, we simulate 500 points from this process with parameters mean  $\mu = 0$ , correlation  $\rho = 0.75$ . We also choose suitable values for the hyperparameters as shown in the R Code. We generate 10000 samples initially, from which we burn in the samples by removing the first 2000 samples and we thin the samples by a factor of 4. This gives us 2000 burned-in and thinned gibbs samples from which we can estimate the values of  $\rho$  and  $\sigma^2$ . The R code is attached with this assignment and the traceplot is attached below.

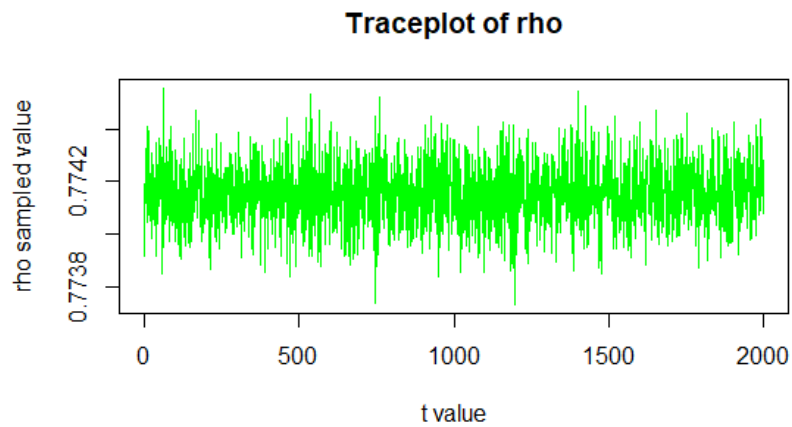


Figure 8: Traceplot for Gibbs sampled values of  $\rho$

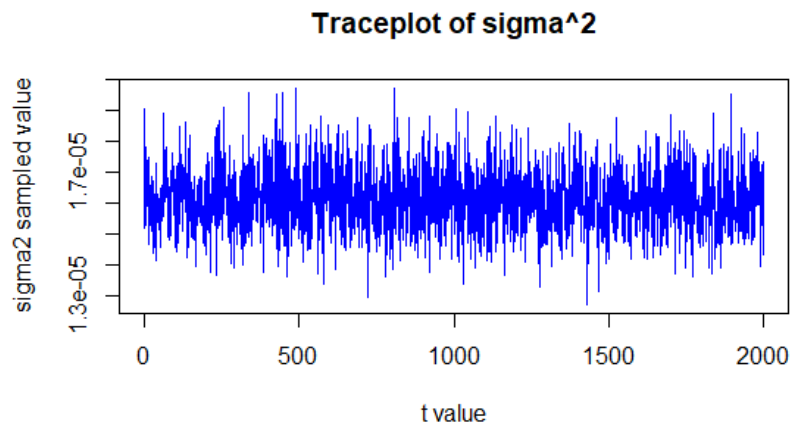


Figure 9: Traceplot for Gibbs sampled values of  $\sigma^2$

After the sampling process, the final values of  $\rho$  and  $\sigma^2$  equals 0.7742298 and 1.579148e-05 respectively.



d. Now, we construct the 90% credible intervals for  $\rho$  and  $\sigma^2$ . We do this in R, using the Quantile based credible intervals, as it was mentioned in the lecture notes that Quantile based credible intervals are used much more in practice. It is defined as,

$$\begin{aligned} C_{\theta,\alpha} &= \{\theta : Pr(C_{\theta,\alpha} | y_{1:n}) = 1 - \alpha\} \\ &= \{\theta_{n,1-\alpha_1} \leq \theta \leq \theta_{n,\alpha_2}, \alpha_1 + \alpha_2 = \alpha\} \end{aligned}$$

Implementing this in R, we get, mean lower bound = 0.694460081689918 and mean upper bound = 0.694460082861651 and variance lower bound =  $1.42763850294374 \times 10^{-05}$  and variance upper bound =  $1.427638503367 \times 10^{-05}$

## 5.

**Solution:** We are given a normal mixture normal model to fit to the galaxies dataset in R. This dataset consists of velocities in km/sec of 82 galaxies from 6 well-separated conic sections of an unfilled survey of the Corona Borealis region. We desire to fit a normal location - scale mixture model of the form

$$y_i \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \text{Normal}(y_i | \mu_k, \sigma_k^2)$$

We are given priors for this model as follows.

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dir}(1/K, \dots, 1/K) \\ \mu_k &\sim \text{Normal}(\mu_0, \sigma_0^2) \\ \sigma_k^2 &\sim \text{Inverse-Gamma}(a_0, b_0) \end{aligned}$$

We need to sample  $z, \mu_k, \sigma_k^2$ . We can get the full conditionals as follows,

$$\begin{aligned} p(z_i = k | \dots) &= \text{Mult}(1, \boldsymbol{\pi}), \pi_{ik} = \frac{\pi_k \times \text{Normal}(y_i | \mu_k, \sigma_k^2)}{\sum_{j=1}^K \pi_j \times \text{Normal}(y_i | \mu_j, \sigma_j^2)} \\ p(\mu_k | \dots) &= \text{Normal}(\mu_{k,n}, \sigma_{k,n}^2), \mu_{k,n} = \sigma_{k,n}^2 \left( \frac{\mu_0}{\sigma_0^2} + \frac{n_k \bar{y}_k}{\sigma_k^2} \right), \sigma_{k,n}^2 = \left( \frac{1}{\sigma^2} + \frac{n_k}{\sigma_k^2} \right)^{-1} \\ \text{where, } n_k &= \sum_i 1(z_i = k), \bar{y}_k = \frac{\sum_i 1(z_i = k) y_i}{n_k} \\ p(\sigma_k^2 | \dots) &= \text{Inv-Gamma}(a_{k,n}, b_{k,n}) \\ a_{k,n} &= (a_0 + \frac{n_k}{2}), b_{k,n} = \left( b_0 + \frac{(n_k - 1)s_k^2}{2} + \frac{n_k(\bar{y}_k - \mu_k)^2}{2} \right) \end{aligned}$$

We implement the gibbs sampler in R by sampling from these conditional distributions. For the pointwise credible region, we calculate them using the idea illustrated in question 4. We calculate the upper and lower bound for each point using the values of these credible regions at each point. The R file containing this results has been attached along with this assignment.

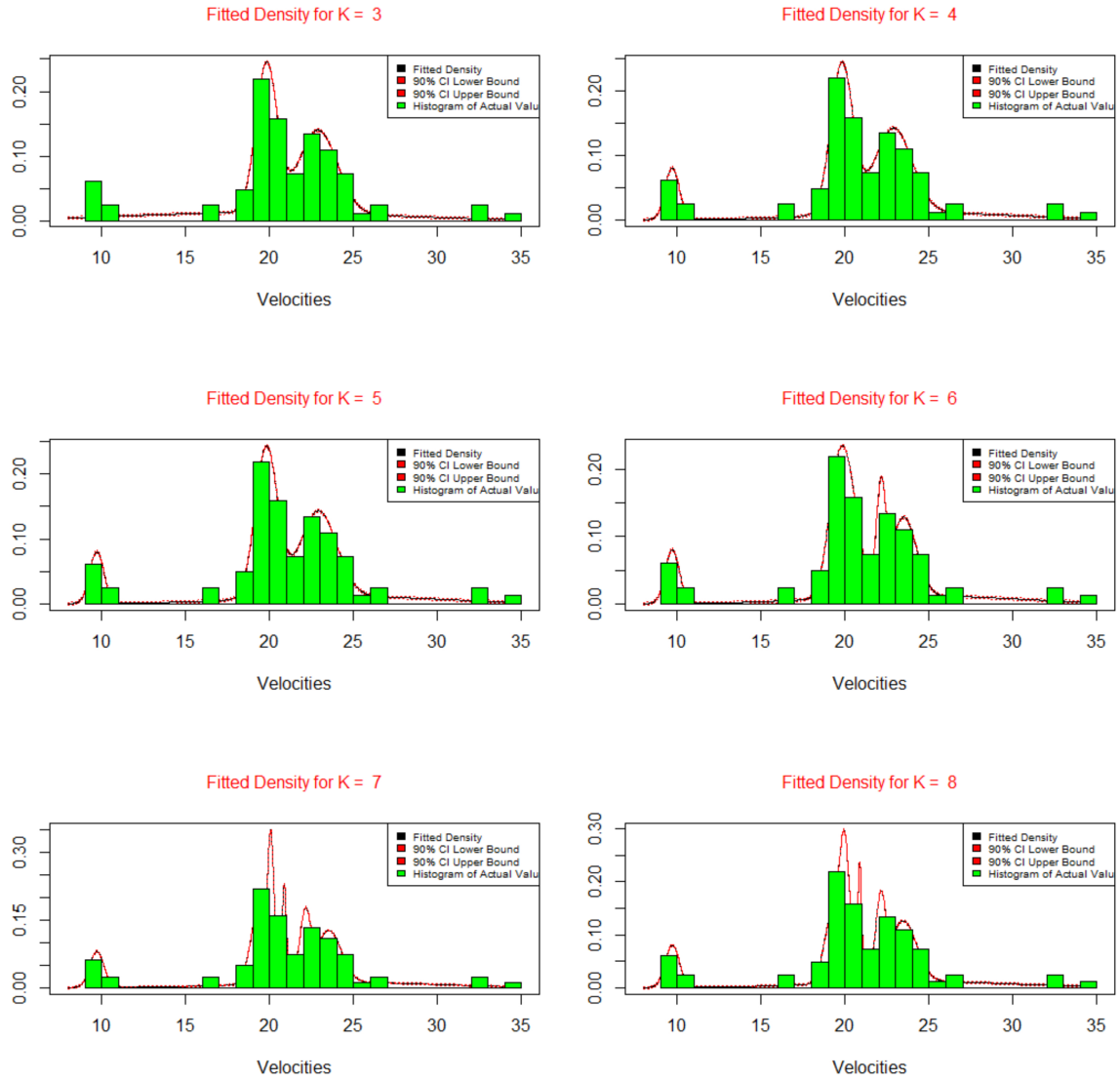


Figure 10: Plot of Fitted Densities, Credible Intrevals and Histogram of sample values

6.

**Solution:** In this problem, we are given a bivariate normal mixture model to fit to the faithful dataset in R. The dataset consists of the waiting times between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA. We desire to fit a Multivariate Normal Mixture Distribution with  $K$  componenets of the form

$$\mathbf{y}_1, \dots, \mathbf{y}_n \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Where the priors are given as follows,

$$\begin{aligned}\boldsymbol{\pi} &\sim \text{Dir}(1/k, \dots, 1/k) \\ \boldsymbol{\mu}_k &\sim \text{Multivariate - Normal}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \\ \boldsymbol{\Sigma}_k &\sim \text{Inverse - Wishart}(v_0, \boldsymbol{\Psi}_0)\end{aligned}$$

Now, from the posterior distributions, we obtain the following full conditionals.

$$\begin{aligned}p(\boldsymbol{\pi} \mid \dots) &= \text{Dir}(\alpha/k, \dots, \alpha/k) \\ p(\boldsymbol{\mu}_k \mid \dots) &= \text{Normal}(\boldsymbol{\mu}_{k,n}, \boldsymbol{\Sigma}_{k,n}^2) \\ \text{where } \boldsymbol{\mu}_{k,n} &= \boldsymbol{\Sigma}_{k,n}(\boldsymbol{\mu}_0 \boldsymbol{\Sigma}_0^{-1} + n_k \bar{y}_k \boldsymbol{\Sigma}_k^{-1}), \\ \boldsymbol{\Sigma}_{k,n} &= (\boldsymbol{\Sigma}_0^{-1} + n_k \boldsymbol{\Sigma}_k^{-1})^{-1} \\ p(\boldsymbol{\Sigma}_k \mid \dots) &= \text{Inverse-Wishart}(v_{k,n}, \boldsymbol{\psi}_{k,n}) \\ \text{where } v_{k,n} &= v_0 + n_k, \\ \boldsymbol{\psi}_{k,n} &= \boldsymbol{\psi}_0 + \sum_{z_i=k} (y_i - \boldsymbol{\mu}_k)(y_i - \boldsymbol{\mu}_k)^T\end{aligned}$$

We implement the gibbs sampler in R by sampling from these conditional distributions and reuse the code from the previous assignment to plot the contour plots. We generate a contour plot by sampling from 100 values of x between 0.1 and 0.6 and y between 0.3 and 1. Here, we note that we worked with scaled down values of eruptions and waiting time, that is, we divide all eruptions by 10 and all waiting times by 100. The results are as shown below. The R file containing this results has been attached along with this assignment.

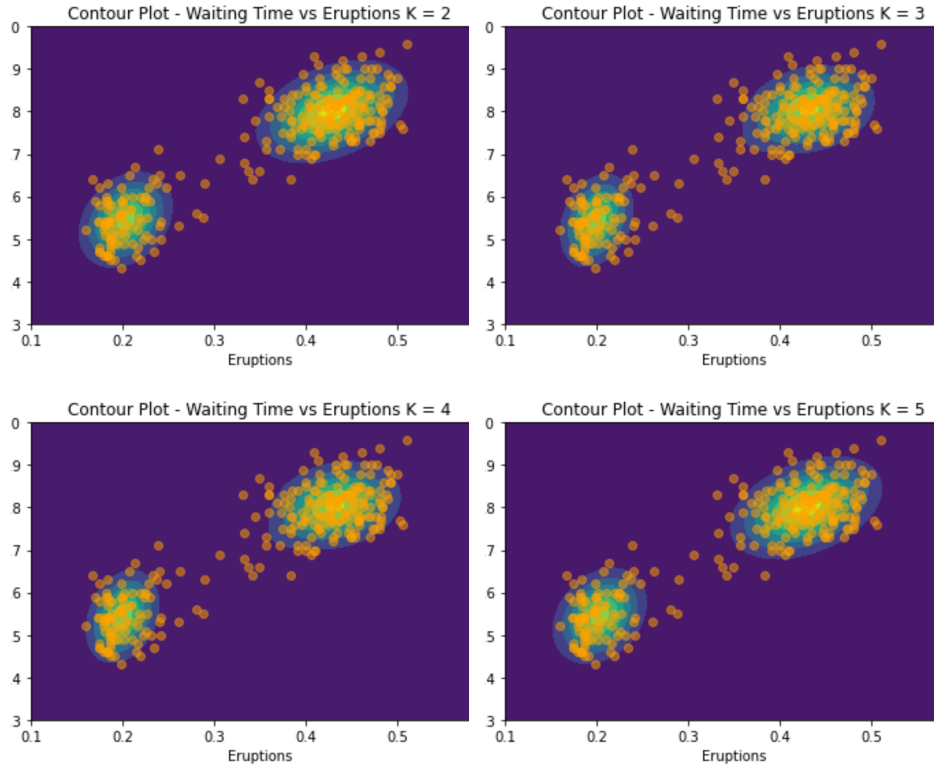


Figure 11: Contour plots superimposed over actual data distribution