

**Homework 2 - STAT 540**  
 Amal Agarwal

## Answer 1

(a) Psudocode:

- Let the unnormalized target kernel be  $h(x)$ . Evaluate the endpoints for the rectangle as  $b = \sup_x(\sqrt{h(x)})$ ,  $c = -\sup_x(x\sqrt{h(x)})$ , and  $d = \sup_x(x\sqrt{h(x)})$  using numerical maximization.
- Uniformly sample in 2 dimensions from the rectangle  $[c \leq u \leq d] \times [0 \leq v \leq b]$ .
- For each sample pair  $(u,v)$ , check the condition  $0 \leq v \leq h\left(\sqrt{\left(\frac{u}{v}\right)}\right)$  and keep only those pairs which satisfy the condition. Note that since the support is  $(1, +\infty)$ , we must have  $u \geq v$ .
- Samples from the target density can be obtained by evaluating  $\frac{u^*}{v^*}$  where  $(u^*, v^*)$  are the accepted pairs from previous step.

(b) R code for the algorithm is given in "aua257HW2.R" file.

(c) Histogram of the samples along with true target kernel is given below:

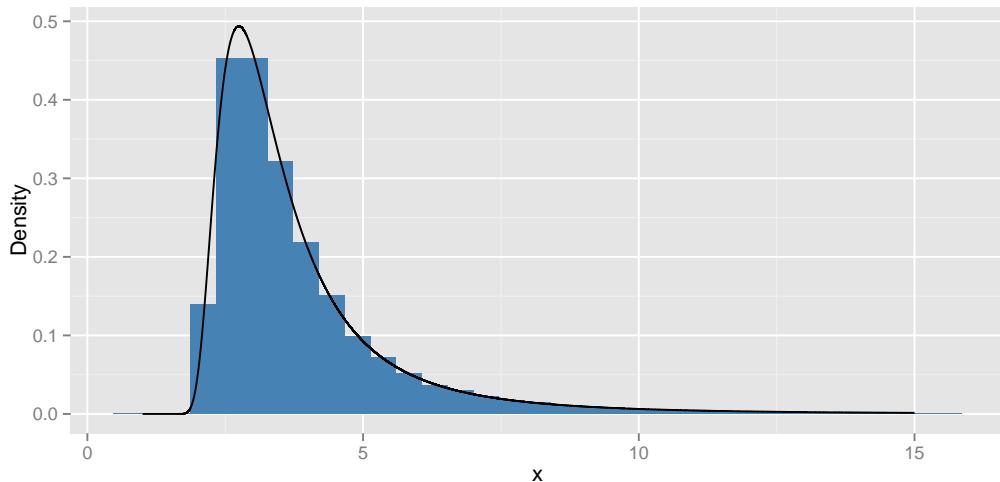


Figure 1: Plot of histogram of samples and comparing with true target kernel

Plot of the ratio of uniforms of regions is given below:

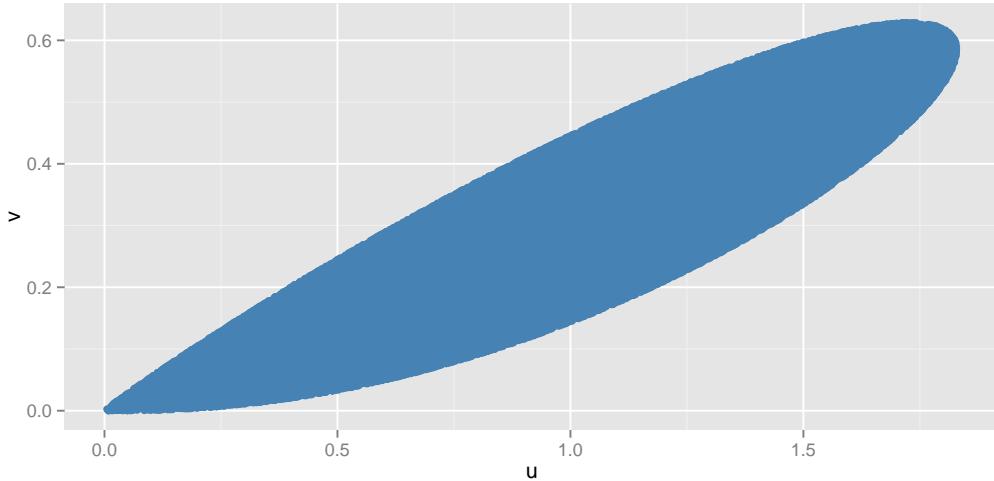


Figure 2: Plot of ratio of uniforms region

- (d) The optimal sample size such that the Monte Carlo standard error is less than 5% of the estimated expectation was found as 248. This was obtained by running the ratio of uniforms function multiple items generating different sets of samples and calculating the optimal size in each case and taking the mean of the optimal sizes.

The estimate of the expected value for  $n = 100,000$  was found as **3.97**.

Plot of the estimates of expected value with the sample size is given as:

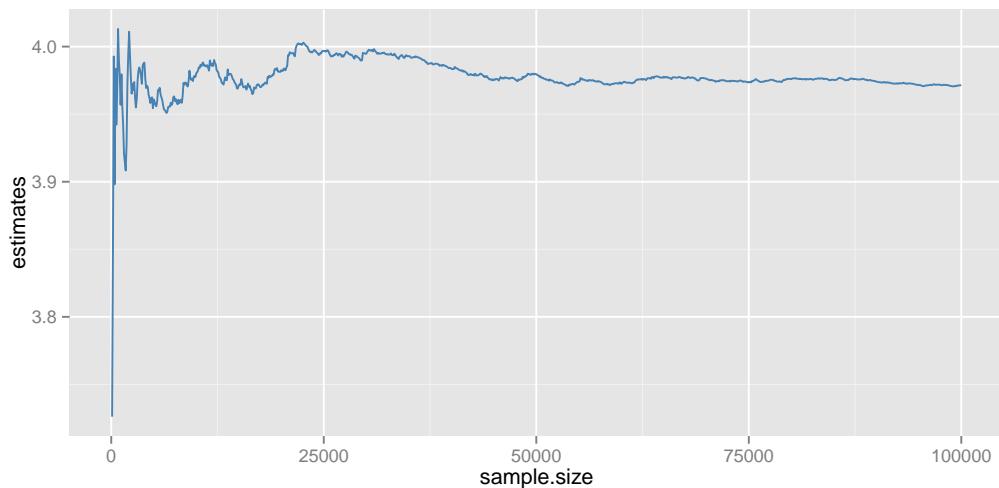


Figure 3: Plot of Monte Carlo estimates vs. sample size

Plot of the Monte Carlo standard errors with the sample size is given as:

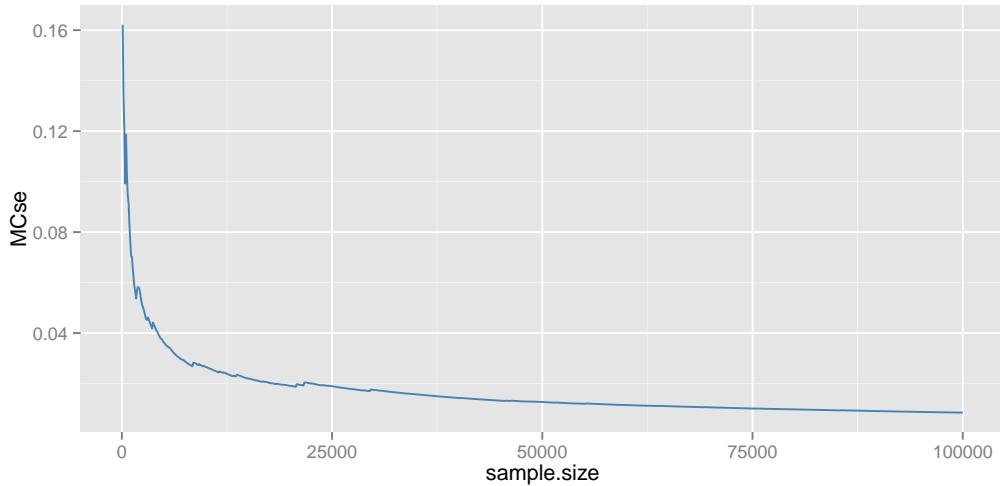


Figure 4: Plot of Monte Carlo standard errors vs. sample size

- (e) The number of samples generated per second were found to be **40193**.

## Answer 2

- (a) Refer to the file "aua257HW2.R" for R code. Pseudocode is given as follows:

- Start off with  $x^{(0)}$  in the support  $(1, +\infty)$  chosen suitably after a series of pilot runs. Choose the length of the Markov chain as  $m$ .
- Run loop from  $t = 0 : (m - 1)$ 
  - Generate a candidate  $x^*$  from the proposal  $q$  as truncated normal  $N(x^{(t)}, \tau^2, 1, +\infty)$ . For sampling  $x^*$ , inverse cdf transformation is being used as follows:
    - \* Generate  $u \sim Uniform(0, 1)$ .
    - \* To obtain  $x^*$ , do the inverse cdf transform on  $u$  as  $x_i^* = qnorm(u \times (pnorm(b_i) - pnorm(a_i)) + pnorm(a_i))$  where  $qnorm$  and  $pnorm$  are quantile and cdf functions for a normal distribution with parameters same as the mean and variance of  $x^*$  defined above.
  - Accept  $x^*$  as next state  $x^{(t+1)}$  with the following acceptance probability  $\alpha$ , else assign the current state as next state.

$$\alpha(x^*, x^{(t)}) = \min \left( 1, \frac{h(x^*)q(x^*, x^{(t)})}{h(x^{(t)})q(x^{(t)}, x^*)} \right)$$

where  $h$  is the target kernel without the factor  $(1/\sigma)$  in the target density.

- (b) The optimal sample size such that the Monte Carlo standard error is less than 5% of the estimated expectation was found as 1275. Again this was obtained by running the MH function multiple items generating different sets of samples and calculating the optimal size in each case and taking the mean of the optimal sizes. Note that in this case, since the batchmeans does not work below 1000, the lower bound for optimum in each case was taken as 1000.

The estimate of the expected value for  $n = 100,000$  was found as **3.87**

- (c) Diagnostic plots: Verifying the acf plot for the chain with starting value as 3.5 and tuning parameter as 9.4

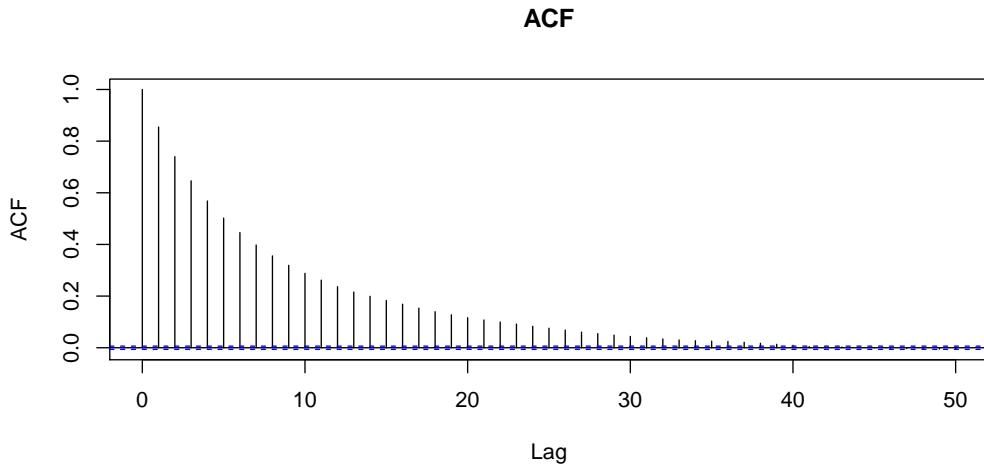


Figure 5: Plot of ACF os samples

The above plot is reasonably good. For  $n = 100,000$ , the effective sample size was found as **5947**. Thus we can conclude the tuning parameter value of 9.4 may not be the best but still it works reasonably good.

For the 3 different starting values as 3.5, 3.7, 3.9, the plot of the estimates as function sample size is given as:

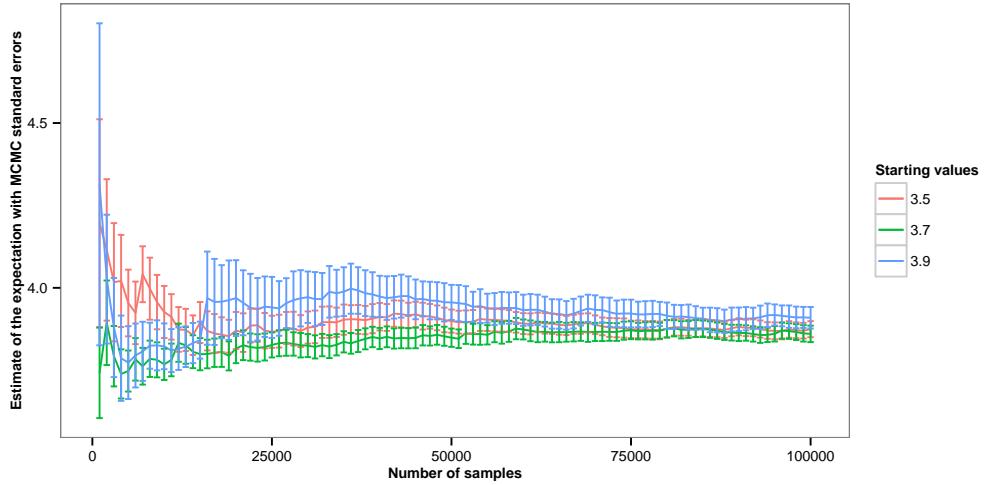


Figure 6: Plot of Monte Carlo estimates vs. sample size

Now plotting the MCMC standard errors as a function of sample size for the same starting values:

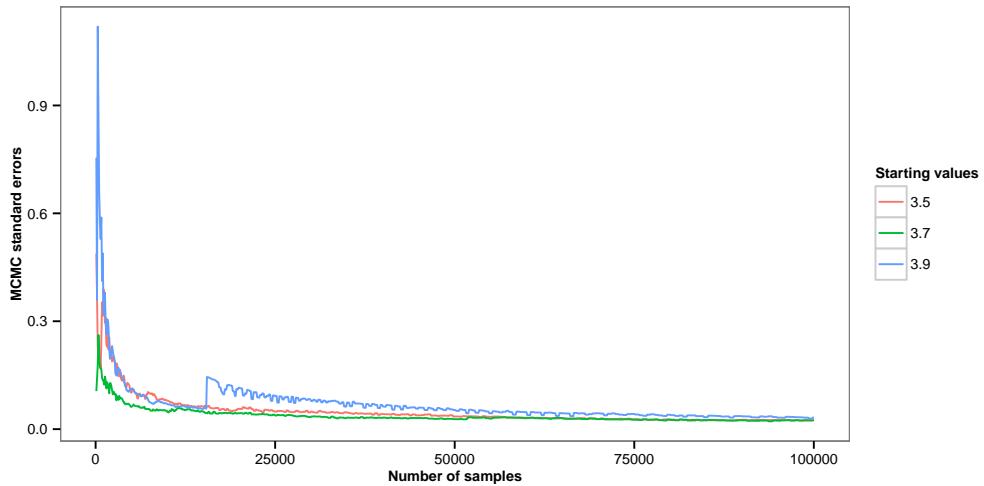


Figure 7: Plot of MCMC standard errors vs. sample size

The above plots show that for different starting values, the estimates converge to same value and MCMC standard errors converge to 0. This verifies the algorithm to some extent.

Plotting the estimated density after  $n/2$  and after  $n$ :

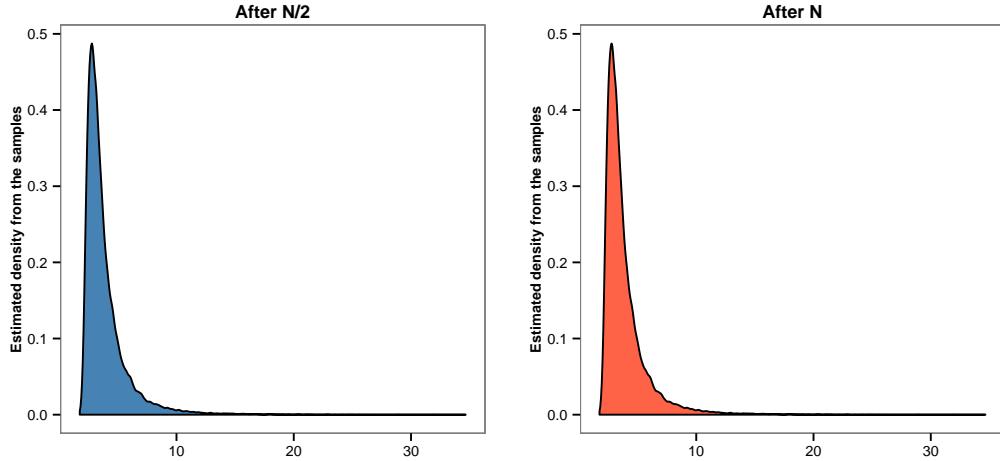


Figure 8: Plot of estimated density

Plotting the estimated density for different starting values:

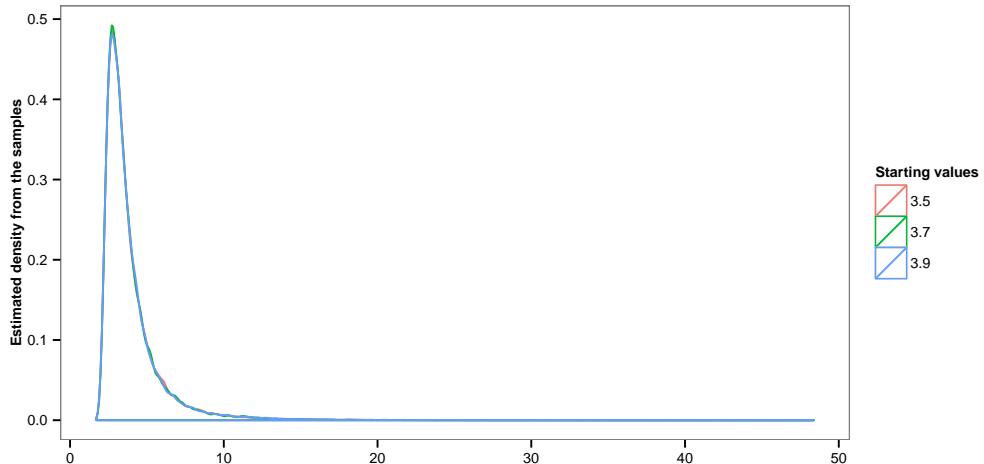


Figure 9: Plot of estimated density

From the above plots, the estimated densities after  $n/2$  and after  $n$  look reasonably identical. Also the estimated density for different starting values also overlap to a good extent. This further verifies the robustness and convergence of the algorithm.

- (d) The effective samples produced per second was found to be **94**. This is much lesser than what was obtained in Q1 part (d) i.e. **40193**, which shows that ratio of uniforms is much more efficient for sampling from the given GEV distribution.

## Answer 3

(a) Pseudocode for Rejection sampling:

- Generate samples  $x^*$ , such that  $x_i^*$  is drawn from univariate truncated normal with bounds  $(a_i, b_i)$  such that  $(a_1, b_1) = (-1, 1)$ ,  $(a_2, b_2) = (2, 4)$  and  $(a_3, b_3) = (3, 5)$ . The means and variances for the proposal of  $x_1^*$ ,  $x_2^*$  and  $x_3^*$  are 0, 1 & 0 and 1, 2 & 3 respectively. For sampling each  $x_i^*$ , inverse cdf transformation is being used as follows:
  - Generate  $u \sim Uniform(0, 1)$ .
  - To obtain  $x_i^*$ , do the inverse cdf transform on  $u$  as  $x_i^* = qnorm(u \times (pnorm(b_i) - pnorm(a_i)) + pnorm(a_i))$  where qnorm and pnorm are quantile and cdf functions for a normal distribution with parameters same as the mean and variance of  $x_i^*$  defined above.
- Evaluate  $K = \sup_x \left( \frac{f(x)}{q(x)} \right)$  using numerical maximization where  $f(\cdot)$  and  $q(\cdot)$  are target and proposal densities,  $q$  being the product of 3 univariate truncated normal densities with parameters as defined above.
- Do accept-reject sequence for each  $x^*$  generated as follows:
  - Generate  $u \sim Uniform(0, 1)$ .
  - Accept  $x^*$  as sample from the target if  $u \leq \frac{f(x_i^*)}{Kq(x_i^*)}$ . Else reject.
- The accepted samples are independent and identically distributed from the target distribution.

(b) The estimate of the expected value for a sample size of  $n = 100,000$  using above algorithm was found to be **(0.35, 2.65, 3.59)**. The corresponding Monte Carlo standard errors for the 3 components were obtained as **(0.03, 0.03, 0.03)**.

Plots of estimates of expected values of the 3 components and Monte Carlo standard errors with sample size are given below:

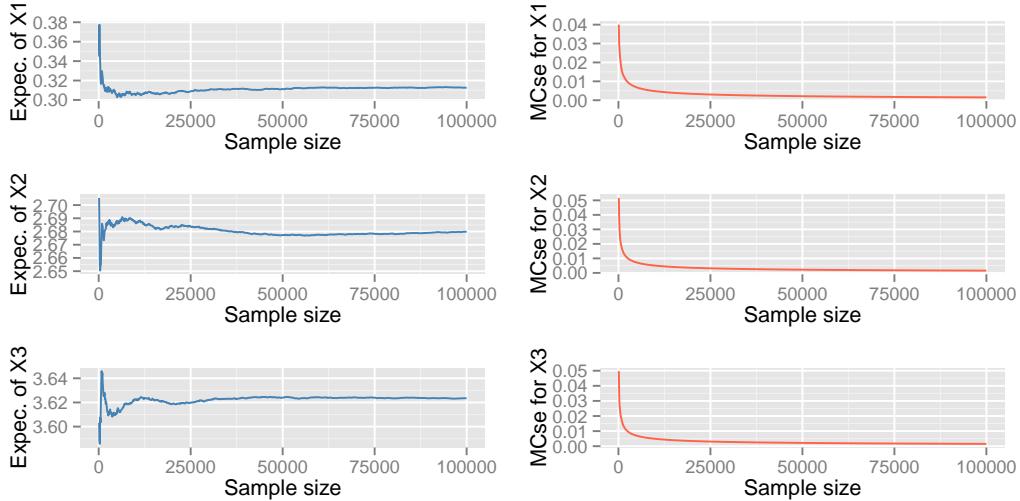


Figure 10: Plot of estimates and MCse vs. sample size for all 3 components

The above plots show that for all the 3 components, the estimates of the expected values converge to some value and MCMC standard errors converge to 0. This verifies the rejection sampling algorithm to a great extent.

- (c) The number of samples generated per second by the above algorithm is **292**. Note here the number of effective samples generated per 10,000 samples is again 10,000 since this is an iid sampler.

## Answer 4

- (a) Pseudocode for the basic importance sampler is given as:

- Generate samples  $x^*$ , such that  $x_i^*$  is drawn from univariate truncated normal with bounds  $(a_i, b_i)$  such that  $(a_1, b_1) = (-1, 1)$ ,  $(a_2, b_2) = (2, 4)$  and  $(a_3, b_3) = (3, 5)$ . The means and variances for the proposal of  $x_1^*$ ,  $x_2^*$  and  $x_3^*$  are 0, 1 & 0 and 1, 2 & 3 respectively. For sampling each  $x_i^*$ , inverse cdf transformation is being used as follows:
  - Generate  $u \sim Uniform(0, 1)$ .
  - To obtain  $x_i^*$ , do the inverse cdf transform on  $u$  as  $x_i^* = qnorm(u \times (pnorm(b_i) - pnorm(a_i)) + pnorm(a_i))$  where  $qnorm$  and  $pnorm$  are quantile and cdf functions for a normal distribution with parameters same as the mean and variance of  $x_i^*$  defined above.

- Calculate the weights  $w(x^*) = \frac{f(x^*)}{q(x^*)}$  where  $f(\cdot)$  is the target truncated multivariate normal density and  $q(\cdot)$  is the product of 3 independent univariate truncated normal proposals.
- Estimate the expectation by taking the sample mean of  $x^* \times w(x^*)$ .

- (b) The estimate of the expected value for a sample size of  $n = 100,000$  using above algorithm was found to be **(0.29, 2.63, 3.52)**. The corresponding Monte Carlo standard errors for the 3 components were obtained as **(0.04, 0.11, 0.14)**.

Plots of estimates of expected values of the 3 components and Monte Carlo standard errors with sample size are given below:

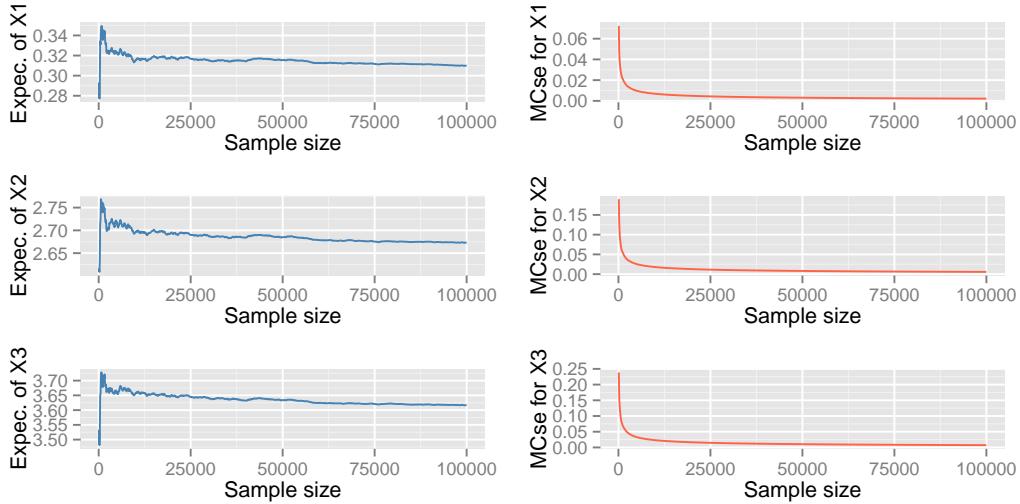


Figure 11: Plot of estimates and MCse vs. sample size for all 3 components

The above plots show that for all the 3 components, the estimates of the expected values converge to some value and MCMC standard errors converge to 0. This verifies the importance sampling algorithm to a great extent.

- (c) The number of effective samples for every 10,000 samples generated was obtained as **7358**. The number of effective samples generated per second were **789**.

## Answer 5

- (a) Pseudocode for the Metropolis Hastings algorithm for the given target density:

- Start off with  $x^{(0)}$  in the support  $(-1, 2, 3) \times (1, 4, 5)$  chosen suitably after a series of pilot runs. Choose the length of the Markov chain as  $m$ .
- Run loop from  $t = 0 : (m - 1)$

- Generate a candidate  $x^*$ , such that  $x_i^*$  is drawn from univariate truncated normal with bounds  $(a_i, b_i)$  such that  $(a_1, b_1) = (-1, 1)$ ,  $(a_2, b_2) = (2, 4)$  and  $(a_3, b_3) = (3, 5)$ . The mean for the proposal of  $x_i^*$  is  $x_i^t$  and variance  $\tau_i^2$  being the tuning parameter. Again for sampling each  $x_i^*$ , inverse cdf transformation is being used as follows:
  - \* Generate  $u \sim Uniform(0, 1)$ .
  - \* To obtain  $x_i^*$ , do the inverse cdf transform on  $u$  as  $x_i^* = qnorm(u \times (pnorm(b_i) - pnorm(a_i)) + pnorm(a_i))$  where `qnorm` and `pnorm` are quantile and cdf functions for a normal distribution with parameters same as the mean and variance of  $x_i^*$  defined above.
- Accept  $x^*$  as next state  $x^{(t+1)}$  with the following acceptance probability  $\alpha$ , else assign the current state as next state.

$$\alpha(x^*, x^{(t)}) = \min \left( 1, \frac{h(x^*)q(x^*, x^t)}{h(x^{(t)})q(x^t, x^*)} \right)$$

where  $h$  is the target truncated multivariate normal kernel and  $q$  is the product of the 3 independent univariate truncated normal proposals.

- (b) The estimate of the expected value for a sample size of  $n = 100,000$  using above algorithm was found to be **(0.35, 2.76, 3.63)**. The corresponding Monte Carlo standard errors for the 3 components were obtained as **(0.05, 0.07, 0.05)**.

Plot of the ACF of samples is given as:

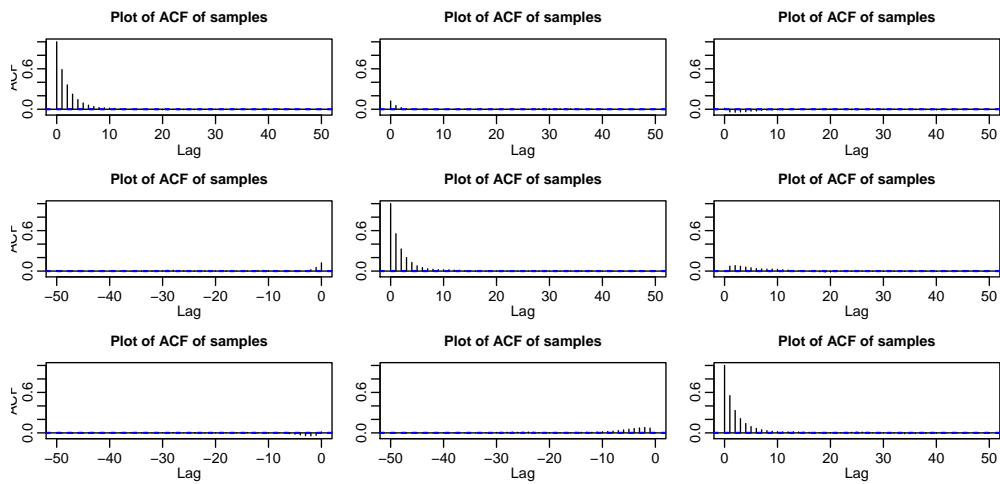


Figure 12: Plot of ACF of samples

For all the 3 components, the above plot is reasonably good which indicates that the tuning parameter value of  $(1, 2, 3)$  may not be the best but still it works reasonably good.

For 3 different starting values as  $(0.30, 2.66, 3.66)$ ,  $(0.35, 2.71, 3.71)$  and  $(0.25, 2.61, 3.61)$ , labelled as 1, 2 and 3, plots of estimates of expected values of the 3 components and Monte Carlo standard errors with sample size are given below:

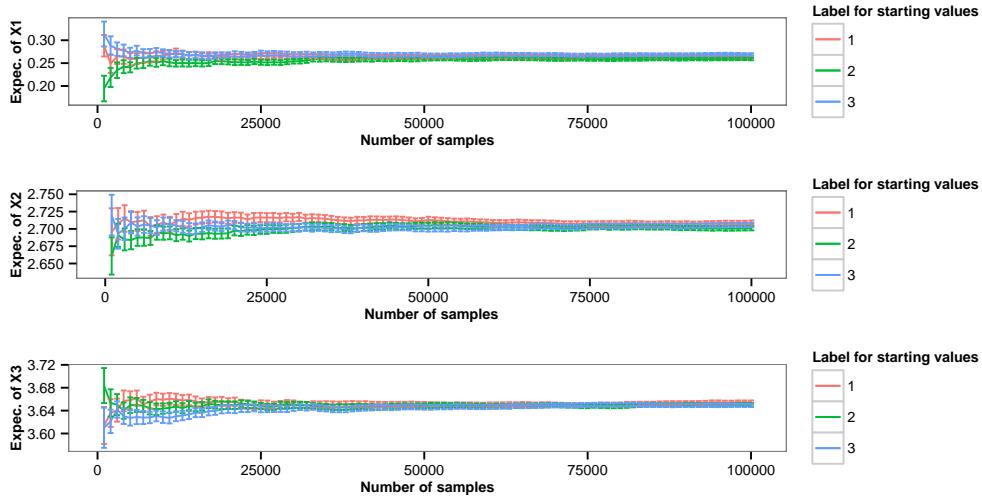


Figure 13: Plot of estimates vs. sample size with error bars for all 3 components

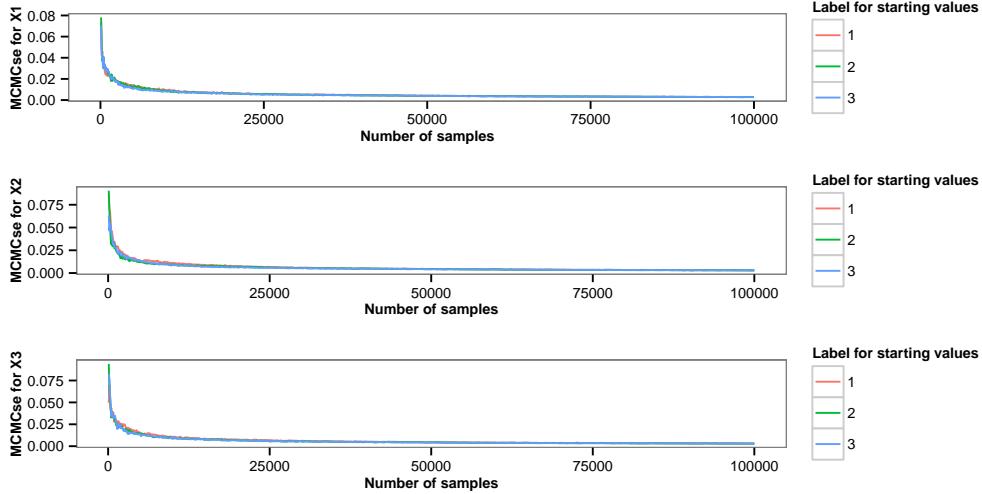


Figure 14: Plot of MCMC se vs. sample size for all 3 components

The above plots show that for all the 3 components and for different starting values, the estimates converge to same value and MCMC standard errors converge to 0. This verifies the algorithm to some extent. Plotting the estimated density after  $n/2$  and after  $n$ :

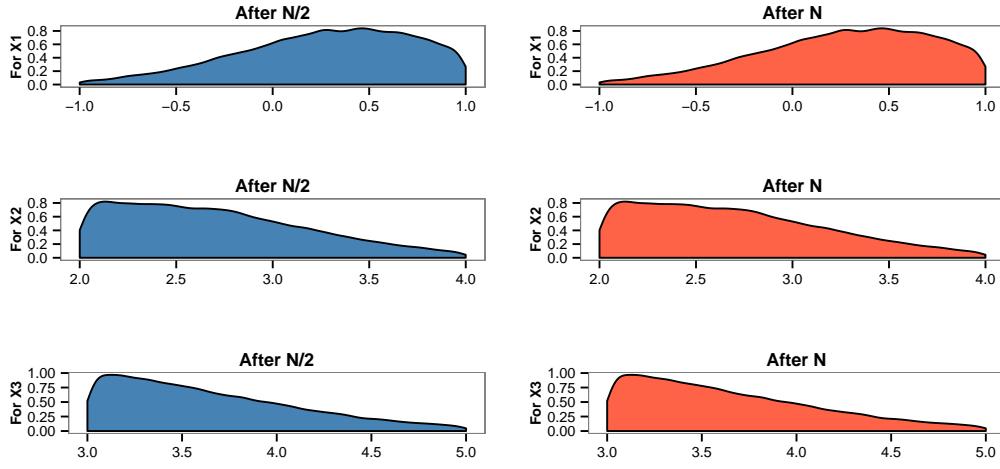


Figure 15: Plot of estimated density

Plotting the estimated density for different starting values for all the components:

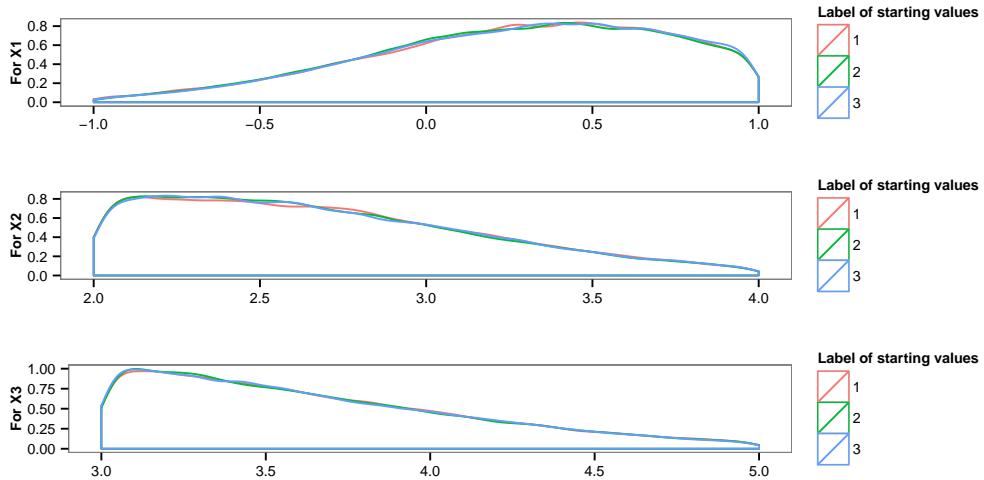


Figure 16: Plot of estimated density for all 3 components

From the above plots, the estimated densities after  $n/2$  and after  $n$  look reasonably identical for all the 3 components. Also the estimated density for different starting values also overlap to a good extent. This further verifies the robustness and convergence of the algorithm.

- (c) The effective samples obtained for every 10,000 samples generated were found to be **2299**. This was calculated by taking the minimum of the effective samples obtained for the 3 individual components. The number of effective samples generated per second was **132**.

(d) The following table gives the required information:

Table 1: Summary of above 3 algorithms for target as Multivariate truncated normal

	Mean estimates	Standard errors	ESS for 10,000 samples	ESS per second
Rejection Sampling	(0.35, 2.65, 3.59)	(0.03, 0.03, 0.03)	10000	292
Importance Sampling	(0.29, 2.63, 3.52)	(0.04, 0.11, 0.14)	7358	789
Metropolis-Hastings	(0.35, 2.76, 3.63)	(0.05, 0.07, 0.05)	2318	132