Assignment 1: ECO 545 (Fall 2020)

I pledge on my honor that I have not given or received any unauthorized assistance.
Name: Deepesh Kumar Lall Roll No.: 170236

---

1. **Solution:** According to the question we have the probability distribution of `n`(# of captured frogs) as a binomial distribution thereby the likelihood function for $p(n|N,p)$ is given as

$$p(n|N,p) \propto \binom{N}{n} p^n (1-p)^{N-n}$$

and prior over the p and N, Given they are independent Parameter are $p \sim Beta(\alpha, \beta)$ and $p(N) \propto 1/N^2$.

As we know According to Bayesian Inference and ignoring the Normalization Constant(i.e. Marginal Probability in denominator of Parameter in Bayes rule), we have posterior for joint distribution (N,p) as:

$$f(p,N|n) \propto p(n|N,p) \cdot p(N) \cdot p$$

$$f(p,N|n) \propto p^{\alpha-1} \cdot (1-p)^{\beta-1} \cdot \frac{1}{N^2} \cdot \frac{N!}{(N-n)!n!} \cdot p^n \cdot (1-p)^{N-n}$$

$$\boxed{f(p,N|n) \propto p^{n+\alpha-1} \cdot (1-p)^{N-n+\beta-1} \cdot \frac{(N-1)!}{N(N-n)!}}$$

Now in order to find marginal distribution for the posterior of N we need to integrate out p (in limits from 0 to 1 as its prior is Beta dist.) from the Joint distribution of the above posterior,

$$p(N|n) \propto \int_0^1 f(p,N|n) \cdot dp$$

$$p(N|n) \propto \frac{(N-1)!}{N(N-n)!} \int_0^1 p^{n+\alpha-1} \cdot (1-p)^{N-n+\beta-1} \cdot dp$$

As, $\int_0^1 p^{\alpha-1} \cdot (1-p)^{\beta-1} \cdot dp = \frac{\Gamma(\alpha) \cdot \Gamma(\beta)}{\Gamma(\alpha+\beta)}$

$$\boxed{p(N|n) \propto \frac{(N-1)!}{N(N-n)!} \cdot \frac{\Gamma(N-n+\beta)}{\Gamma(N+\alpha+\beta}}$$

The comparison of the above probability density with change in hyper-parameter and parameter n is plotted in the code with results as shown below.

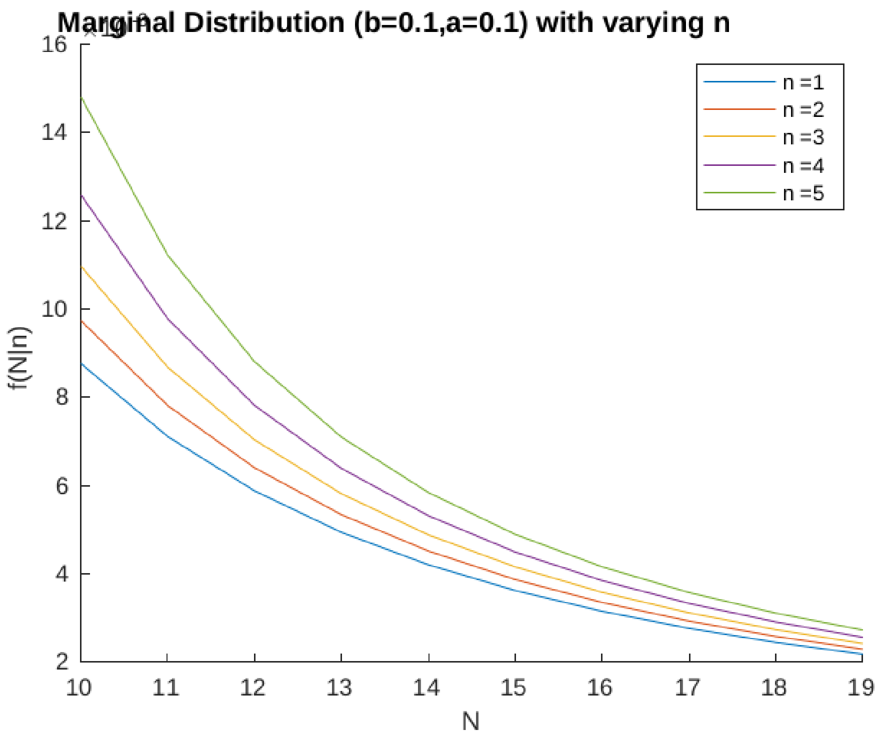

Figure 1: Plot with changing parameter n.

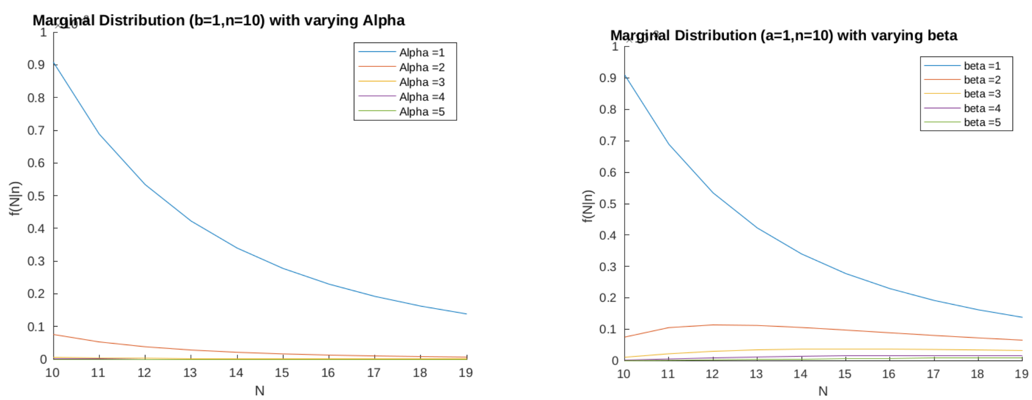

Figure 2: Plot with changing Hyperparameter.

The code to run the MATLAB file is simple run in the GUI mode after uploading and it will generate and Save the above plots in the same directory where the .m file is present. The code is sectioned into 3 equal parts with the 1st part for plotting change in plots with changing Beta, 2nd with changing alpha and 3rd with changing n , each saved with the same corresponding name. □

2. **Solution:** Given the outcomes for all faces of dice for n = 100 or 1000. Let

$$y_i = \begin{cases} 1, & \text{if outcome of i-th trial} = 1 \text{ (i.e. 1st face)} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Then we can represent the Likelihood of all the n outcomes with a given Parameter that we are interested in θ_1 i.e. the probablity of Dice to give 1st face as:

$$f(y_1, y_2, ..., y_n|\theta_1) = \theta_1^{y_1}(1-\theta_1)^{1-y_1} \cdot \theta_1^{y_2}(1-\theta_1)^{1-y_2} \cdot ... \cdot \theta_1^{y_n}(1-\theta_1)^{1-y_n}$$
$$f(y_1, y_2, ..., y_n|\theta_1) = \theta_1^{\sum_{i=1}^n y_i}(1-\theta_1)^{n-\sum_{i=1}^n y_i}$$

We can notice that in the above equation the value of $\sum_{i=1}^n y_i$ is the total no of favourable outcomes in all the n trial i.e. no of Outcome with 1st face. As given in the Table so we know the likelihood function for both n = 100 and 1000. As Given the parameter θ_1 is sampled prior from Dirichlet distribution; where each $\alpha_i = 2$ thereby, we can say considering only the random variable of our concern θ_1 we can consider the probability of happening any other outcome is $\propto (1-\theta_1)^{\sum_{i\neq 1}\alpha_i - 1}$ and is more or less this is $\propto \Pi_{i\neq 1}\theta_i^{\alpha_i - 1}$, Thereby we can write Prior for θ_1 as a Beta function i.e. Beta(α_1,$\sum_{i\neq 1}\alpha_i$):

$$\pi(\theta_1) \propto \theta_1^{\alpha_1 - 1} \cdot (1-\theta)^{\sum_{i\neq 1}\alpha_i - 1}$$
$$\pi(\theta_1) \propto \theta_1^{2-1} \cdot (1-\theta)^{10-1}$$

Hence now using the bayesian inference we can compute the posterior distribution as:

$$\pi(\theta_1|y_1, y_2, ..., y_n) \propto f(y_1, y_2, ..., y_n|\theta_1) \cdot \pi(\theta_1)$$
$$\pi(\theta_1|y_1, y_2, ..., y_n) \propto \theta_1^{\sum_{i=1}^n y_i}(1-\theta)^{n-\sum_{i=1}^n y_i} \cdot \theta_1^{2-1} \cdot (1-\theta)^{10-1}$$
$$\pi(\theta_1|y_1, y_2, ..., y_n) \propto \theta_1^{\sum_{i=1}^n y_i + 1}(1-\theta)^{n-\sum_{i=1}^n y_i + 9}$$
$$\pi(\theta_1|y_1, y_2, ..., y_n) \propto Beta(\sum_{i=1}^n y_i + 2, n - \sum_{i=1}^n y_i + 10)$$

Using the Table and value of $\sum_{i=1}^n$ for both the n we get,

For n = 100 : $Beta(19 + 2, 100 - 19 + 10) = \boxed{Beta(21, 91)}$

For n = 1000 : $Beta(190 + 2, 1000 - 190 + 10) = \boxed{Beta(192, 820)}$

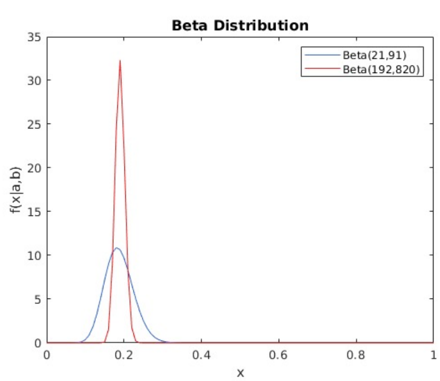

Figure 3: Plot of Beta function for different parameter.

Plotting the above 2 plots we can see the variance decreases as the no of trials are increased. □

3. **Solution:** As given to us the multiple regression model analysis with $y_i = x\prime_i\beta + \epsilon_i$, where $\epsilon_i|x_i \sim N(0, \sigma^2)$ for all $i = 1, ..., n$. We can say that:

$$f(y_1, y_2, ..., y_n|\beta, \sigma^2) = (\frac{1}{\sigma^2})^{n/2} exp(-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta))$$

Also considering the given prior for the 2 independent parameter β and σ^2 as $N_k(\beta_0, B_0)$ and $IG(\alpha_0/2, \delta_0/2)$ respectively.

$$\pi(\beta, \sigma^2) = \pi(\beta)\pi(\sigma^2) = N_k(\beta_0, B_0) \cdot IG(\alpha_0/2, \delta_0/2)$$

Thereby We can compute the joint Posterior distribution of both the parameter for above multi-regression model as:

$$\pi(\beta, \sigma^2|y_1, y_2, ..., y_n) \propto f(y_1, y_2, ..., y_n|\beta, \sigma^2) \cdot N_k(\beta_0, B_0) \cdot IG(\alpha_0/2, \delta_0/2)$$

$$\pi(\beta, \sigma^2|y_1, y_2, ..., y_n) \propto (\frac{1}{\sigma^2})^{n/2} exp(-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta)) \cdot N_k(\beta_0, B_0) \cdot IG(\alpha_0/2, \delta_0/2)$$

$$\pi(\beta, \sigma^2|y_1, y_2, ..., y_n) \propto (\tfrac{1}{\sigma^2})^{n/2} exp(-\tfrac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta)) \cdot exp(-\tfrac{1}{2}(\beta - \beta_0)^T B_0^{-1}(\beta - \beta_0)) \cdot (\tfrac{1}{\sigma^2})^{\alpha_0/2+1} exp(-\tfrac{\delta_0}{2\sigma^2})$$

$$\pi(\beta, \sigma^2|y_1, y_2, ..., y_n) \propto (\frac{1}{\sigma^2})^{(\alpha_0+n)/2+1} exp(-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta) - \frac{\delta_0}{2\sigma^2} - \frac{1}{2}(\beta - \beta_0)^T B_0^{-1}(\beta - \beta_0))$$

(a) Now accumulating all the term having β on one side and dfferent from σ^2 from the above joint posterior distribution we can get marginalized posterior for β as the 2 parameter are mentioned independent. Hence we can see in the above distribution all the β term are in an $exp(.)$ and maximum order of β goes 2 in $\beta^T\beta$, thereby we can say $\pi(\beta|\sigma^2, y_1, y_2, ..., y_n)$ is a Normal Distribution of β with let's say mean of $\bar{\beta}$ and Variance of B_1. Now as

$$\pi(\beta|\sigma^2, y_1, y_2, ..., y_n) = N_k(\bar{\beta}, B_1)$$

$$\pi(\beta|\sigma^2, y_1, y_2, ..., y_n) \propto exp(-\frac{1}{2}(\beta - \beta_0)^T B_1^{-1}(\beta - \beta_0))$$

Now comparing the coeff. of β in the above eqn. with those in joint marginal distribution:

Comparing Coeff of $\beta^T\beta$: $-\frac{1}{2}B_1^{-1} = -\frac{1}{2\sigma^2}X^TX - \frac{1}{2}B_0^{-1}$ .... (1)

Comparing Coeff of β^T : $\frac{1}{2}B_1^{-1}\bar{\beta} = \frac{1}{2\sigma^2}X^Ty - \frac{1}{2}B_0^{-1}\beta_0$ .... (2)

As we have 2 eqn and 2 variable we can compute B_1 from eqn 1, and put its value in eqn2 and compute $\bar{\beta}$ as (can be verified by comparing for coeff. of β it comes LHS = RHS):

$$\boxed{\pi(\beta|\sigma^2, y_1, y_2, ..., y_n) \sim N_k(\bar{\beta}, B_1) \text{ , for } B_1 = [\sigma^{-2}X^TX + B_0^{-1}]^{-1} \text{ ,} \bar{\beta} = B_1[\sigma^{-2}X^Ty + B_0^{-1}\beta_0]}$$

(b) Similarly accumulating term for σ^2 keeping β constant in the joint posterior distribution we can straightforwardly see that:

$$\pi(\sigma^2|\beta, y_1, y_2, ..., y_n) \propto (\frac{1}{\sigma^2})^{(\alpha_0+n)/2+1} exp(-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta) - \frac{\delta_0}{2\sigma^2})$$

$$\pi(\sigma^2|\beta, y_1, y_2, ..., y_n) \propto (\frac{1}{\sigma^2})^{(\alpha_0+n)/2+1} exp(-\frac{1}{2\sigma^2}((y - X\beta)^T(y - X\beta) + \delta_0))$$

$$\pi(\sigma^2|\beta, y_1, y_2, ..., y_n) \propto IG((\alpha_0 + n)/2, ((y - X\beta)^T(y - X\beta) + \delta_0)/2)$$

$$\boxed{\pi(\sigma^2|\beta, y_1, y_2, ..., y_n) \propto IG(\alpha_1/2, \delta_1/2) \text{ , for } \alpha_1 = \alpha_0 + n \text{ ,} \delta_1 = (y - X\beta)^T(y - X\beta) + \delta_0}$$

$\square$

4. **Solution:** Given Target density: $Beta(3.7, 4.8)$(let be $f(x)$), So theoretically the Mean of this distribution is $\mathbf{E}[X] = \frac{\alpha}{\alpha+\beta} = \boxed{4.35294}$

(1) In this Sub-part we need to assume the proposal density to be $Beta(4, 5)$(let be $g(x)$) and thereby use AR algorithm to generate the Data. For implementing this algorithm we need a constant c s.t. $f(x) < c \cdot g(x) \forall x \in [0, 1]$ as Beta distribution are defined in range [0,1].

For the above condition to be true we have,

$$f(x) < c \cdot g(x) \forall x \in [0, 1]$$

$$\frac{\Gamma(3.7 + 4.8)}{\Gamma(3.7) \cdot \Gamma(4.8)} x^{3.7-1}(1-x)^{4.8-1} < c \cdot \frac{\Gamma(4 + 5)}{\Gamma(4) \cdot \Gamma(5)} x^{4-1}(1-x)^{5-1}$$

$$c > \frac{\Gamma(8.5) \cdot \Gamma(4) \cdot \Gamma(5)}{\Gamma(9) \cdot \Gamma(3.7) \cdot \Gamma(4.8)} \cdot \frac{x^{2.7}(1-x)^{3.8}}{x^3(1-x)^4}$$

$$c > \frac{\Gamma(8.5) \cdot \Gamma(4) \cdot \Gamma(5)}{\Gamma(9) \cdot \Gamma(3.7) \cdot \Gamma(4.8)} \cdot x^{-0.3}(1-x)^{-0.2}$$

$$\forall x \in [0, 1]$$

As we can notice the above function i.e. $x^{-0.3}(1-x)^{-0.2}$ doesn't have a maxima and is bounded to infinity and a unique minima at x=0.6 we can argue that there exist no real c s.t. $f(x) < c \cdot g(x) \forall x \in [0, 1]$, Although taking a Larger c will decrease the chances when $f(x) < c \cdot g(x) \forall x \in [0, 1]$ but there will always exist sampling from Beta distribution close to x=0 or x=1 where $f(x) > c \cdot g(x)$ but taking a large c will lead to less probability of proposed sample to be accepted and thereby we need to repeat the AR algorithm for larger no of time to get the plausible approximation of Target Density as rejection frequency will increase.

Algorithm 1: AR Algorithm Pseudo Code

Result: Sample from Beta(3.7,4.8)(i.e. $f(x)$) Target Density
Let $g(x)$ = Beta(4,5) be proposal Density and c=100 (considered large).
for i := 1 to (No. of Samples) do begin
  $y \sim Beta(4, 5)$
  $u_1 \sim U(0, 1)$
  if $u_1 < Beta(y : 3.7, 4.8)/100Beta(y : 4, 5)$
    $x_i = y$
    $i = i + 1$
end

The above algorithm is implemented in the code and used for sampling 1000 samples and thereby in total Expected no of Computation $\sim$ 100000. The Evaluated mean for samples from AR algorithm = $\boxed{0.438792}$.

(2) Incase of MH algorithm with Beta(4,5) as Proposal Density (lets say it $g(x)$) the algorithm is more general with the proposal g(x,y) = g(y) = Beta(y:4,5) i.e. RANDOM WALK. Hence the algorithm goes like:

Algorithm 2: MH Algorithm Pseudo Code

Result: Sample from Beta(3.7,4.8)(i.e. $f(x)$) Target Density
Let $g(x)$ = Beta(4,5) be proposal Density and $x_0 = 0.5$
for i := 1 to (No. of Samples + Burnin) do begin
  $y \sim Beta(4,5)$
  $u_1 \sim U(0,1)$
  if $u_1 \leq min(1, \frac{Beta(y:3.7,4.8)}{Beta(y:4,5)} \cdot \frac{Beta(x_{i-1}:4,5)}{Beta(x_{i-1}:3.7,4.8)})$
    $x_i = y$
    $i = i + 1$
end
Return x_i from i = (Burnin + 1) to (No. of Samples + Burnin)

The above algorithm is implemented in the code and used for sampling 1000 samples with a Burnin of 100 and thereby in total Expected no of Computation $\sim$ 1100. The Evaluated mean for samples from AR algorithm = $\boxed{0.4295362}$.

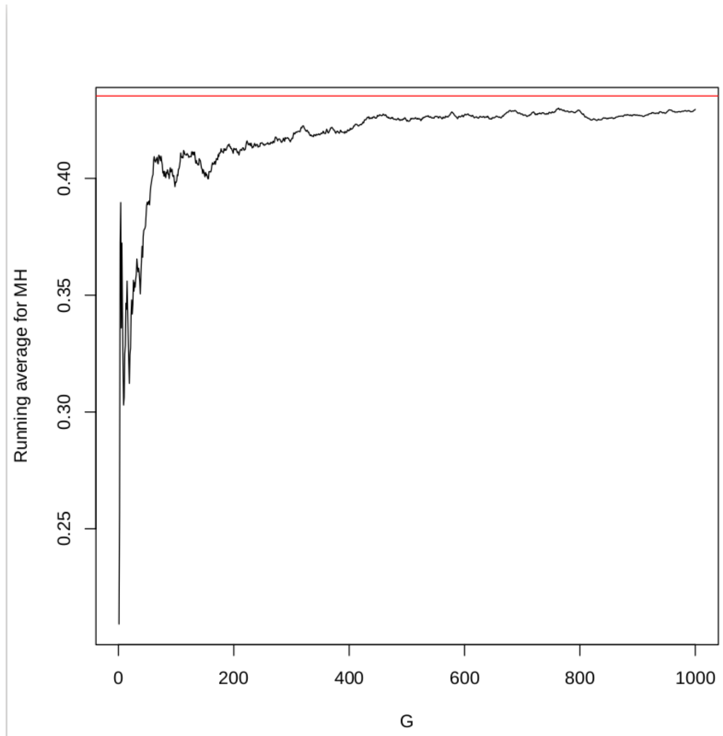

Figure 4: Results of AR and MH algoriothm.

The above Graph demonstrate the Convergence rate of average of our Sampled Data towards the True mean of Target Density w.r.t. the increasing Sample Size i.e. G.

The Code for this question is present in appendix and can be executed from Terminal by running ‘`Rscript q4.r`‘ on the command line. The code is the implementation of above psuedo algorithm and the result of the code are compared in the generated pdf i.e. q4.pdf and Final Sample are plotted in Rplots.pdf.

The Graphical results are as shown below and compared accordingly while the output on command line give the True mean and that generated from the Sampled data by AR and MH algo. (refer to appendix).

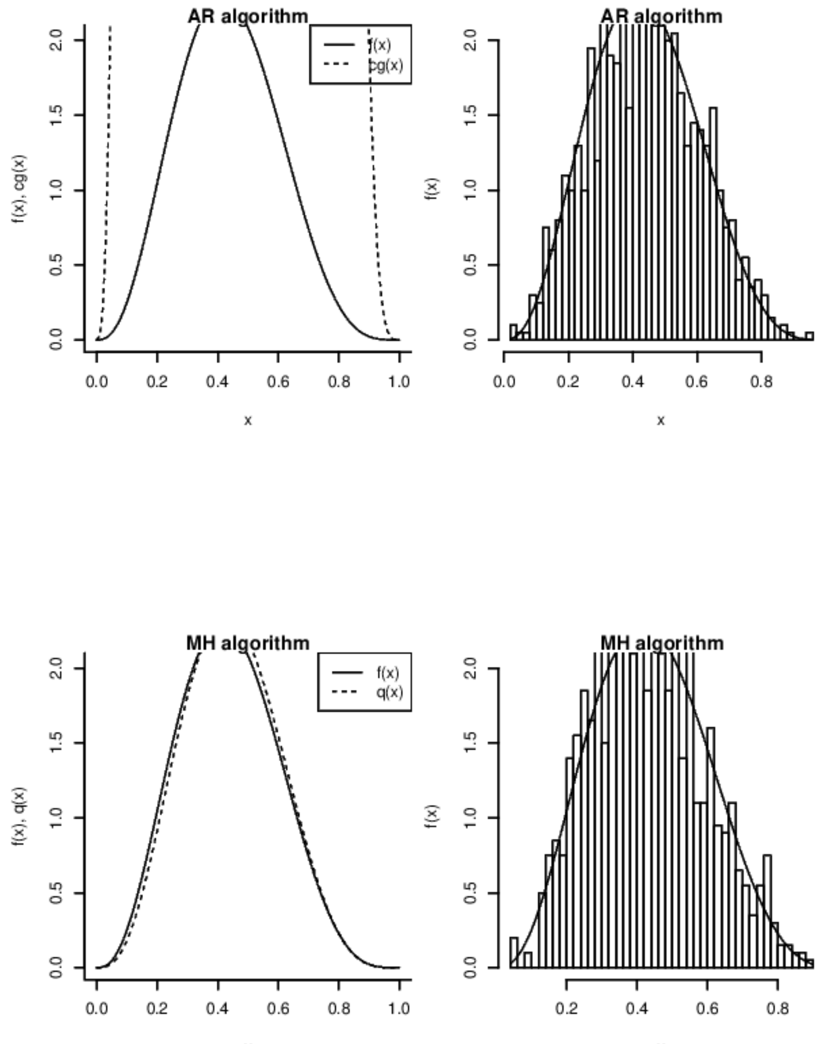

Figure 5: Results of AR and MH algoriothm.

□

5. **Solution:** As Given the model $y_i = \beta x_i + u_i$ where $u_i \sim N(0,1), i = 1, ..., n$, with gamma prior distribution $\beta \sim Ga(2,1), \beta > 0$. The likelihood of the above given model,

$$f(y_1, ..., y_n|\beta) = (\frac{1}{\sqrt{2\pi}})^n exp(-\frac{1}{2}\sum_{i=1}^{n}(y_i - \beta x_i)^2)$$

and prior on parameter β,

$$\pi(\beta) = \frac{1}{\Gamma(2)}\beta exp(-\beta)$$

Now using the Bayesian inference for Posterior distribution using Likelihood and Prior we get,

$$\pi(\beta|y_1, ..., y_n) \propto f(y_1, ..., y_n|\beta) \cdot \pi(\beta)$$

$$\boxed{\pi(\beta|y_1, ..., y_n) \propto \beta exp[-\beta]exp[-\frac{1}{2}\sum_{i=1}^{n}(y_i - \beta x_i)^2], \beta > 0}$$

As the above equation isn't in its standard form we can take log on both side,

$$log(\pi(\beta|y_1, ..., y_n)) = log(\beta) - \beta - \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta x_i)^2 + ...$$

Taking derivative and equating to zero will give us the maxima of the non-standard $\pi(\beta|y_1, ..., y_n)$ i.e. the mean of the distribution (Say $\hat{\beta}$),

$$\frac{\partial log(\pi(\beta|y_1, ..., y_n))}{\partial\beta} = 1/\beta - 1 + \sum_{i=1}^{n}(y_i - \beta x_i)x_i$$

At $\beta = \hat{\beta}$,

$$\frac{\partial\pi(\beta|y_1, ..., y_n)}{\partial\beta} = 0$$

$$1/\hat{\beta} - 1 + \sum_{i=1}^{n}(y_i - \hat{\beta}x_i)x_i = 0$$

$$\boxed{\hat{\beta} = \frac{(\sum_i x_i y_i - 1) + \sqrt{((\sum_i x_i y_i - 1)^2 + 4\sum_i x_i^2)}}{2\sum_i x_i^2}}$$

As given the scale factor is the Negative Inverse of Second derivative at $\hat{\beta}$ So,

$$\boxed{(-\frac{\partial^2 log(\pi(\beta|y_1, ..., y_n))}{\partial\beta^2})^{-1} = (1/\hat{\beta}^2 + \sum_i x_i^2)^{-1}}$$

Thereby we found both the mean and Scale factor for the Student-t distribution which we need to take the Proposal density for MH algorithm with $\pi(\beta|y_1, ..., y_n)$ as the target density. There by Assuming Student-t to be g(.) with degree of freedom of 5(as given) and truncated to $(0, \inf)$ and Target density as f(.) we have MH algo as, (Note here proposal density is independent of past value g(x,y) = g(y) i.e. RANDOM WALK.)

$$f(x) \propto x \cdot exp(-x) \cdot exp(-\frac{1}{2}\sum_{i=1}^{n}(y_i - x \cdot x_i)^2)$$

$$g(x) \propto (1 + \frac{(x-\hat{\beta})^2}{5(1/\hat{\beta}^2 + \sum_i x_i^2)^{-1}})^3$$

In the below Algorithm we can remove the $\propto$ to $=$ as ratio is taken for each distribution with itself there by cancelling out the constant.

Algorithm 3: MH Algorithm Pseudo Code

Result: Sample from $f(x)$ Target Density
Let $g(x)$ be proposal Density and $x_0 = 1$
for i := 1 to (No. of Samples + Burnin) do begin
  $y \sim g(.)$
  $u_1 \sim U(0,1)$
  if $u_1 \leq min(1, \frac{f(y)}{f(x_{i-1})} \cdot \frac{g(x_{i-1})}{g(y)})$
    $x_i = y$
    $i = i + 1$
end
Return x_i from i = (Burnin + 1) to (No. of Samples + Burnin)

The above Algorithm is implemented in appendix for n=50 and can be changed to n=500 for the given Sample Size of 5000 and Burnin of 1000. Thereby giving 4000 affective sample from The Target distribution. The below plots of histogram give the approximation of target density,

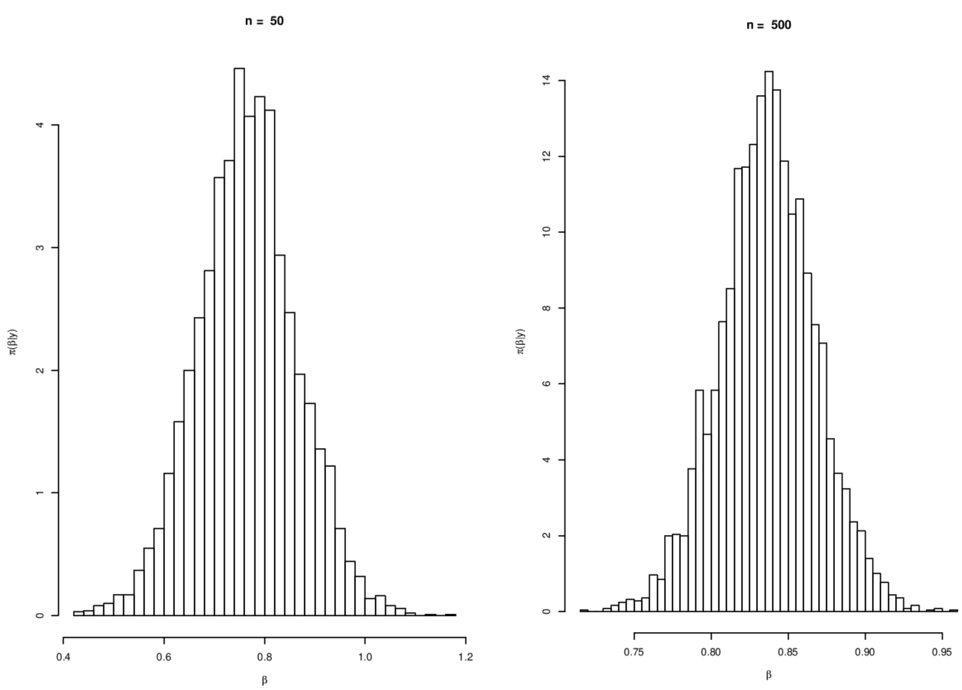

Figure 6: Sample from Target Density for n=50 and n=500

Above Graph demonstrate the affect of n of plot as it decreases the Variance and increase density at True $\hat{\beta}$ i.e. mean of that sample of that corresponding n samples. It can also be confirmed from the detailed statistics output of code on command line. □

APPENDIX

1. **Code for Question 1**: (This Code is written in MATLAB.)

```
clc;
clear all;

%% Varying with beta
hold on;
for b = 1:5
    n = 10;
    a = 1;
    y = zeros(1,10);
    x = linspace(10,19,10);
    for N = 10:19
        answer = factorial(N-1)*gamma(N-n+b);
        answer = answer/(N*factorial(N-n)*gamma(N+a+b));
        y(N-9) = answer;
    end
    plot(x,y);
    if b == 1
        str = {strcat('beta = ' , num2str(1))};
    else
        str = [str , strcat('beta = ' , num2str(b))];
    end
end
xlabel('N')
ylabel('f(N|n)')
title('Marginal Distribution (a=1,n=10) with varying beta','Fontsize',12);
legend(str{:});
saveas(gcf,'beta-hyperparameter.pdf')
hold off;

%% Varying with alpha
clf("reset")
hold on;
A = linspace(1,5,5);
for i = 1:5
    n = 10;
    b = 1;
    a = A(i);
    y = zeros(1,10);
    x = linspace(10,19,10);
    for N = 10:19
        answer = factorial(N-1)*gamma(N-n+b);
        answer = answer/(N*factorial(N-n)*gamma(N+a+b));
        y(N-9) = answer;
    end
    plot(x,y);
    if i == 1
        str = {strcat('Alpha = ' , num2str(a))};
    else
        str = [str , strcat('Alpha = ' , num2str(a))];
```

```
    end
end
xlabel('N')
ylabel('f(N|n)')
title('Marginal Distribution (b=1,n=10) with varying Alpha','Fontsize',12)
legend(str{:});
saveas(gcf,'alpha-hyperparameter.pdf')
hold off;

%% With Varying n
clf("reset")
hold on;
b = 0.1;
a = 0.1;
for i = 1:5
    y = zeros(1,10);
    x = linspace(10,19,10);
    for N = 10:19
        answer = factorial(N-1)*gamma(N-i+b);
        answer = answer/(N*factorial(N-i)*gamma(N+a+b));
        y(N-9) = answer;
    end
    plot(x,y);
    if i == 1
        str = {strcat('n = ' , num2str(i))};
    else
        str = [str , strcat('n = ' , num2str(i))];
    end
end
xlabel('N')
ylabel('f(N|n)')
title('Marginal Distribution (b=0.1,a=0.1) with varying n','Fontsize',12)
legend(str{:});
saveas(gcf,'n-parameter.pdf')
hold off;
```

2. **Code for Question 2**: (This Code is written in MATLAB.)

```
clc;
clear all;

n = 100
x1 = 0 : 0.01 : 1;
y1 = betapdf(x1,21,91);


n = 1000
x = 0 : 0.01 : 1;
y = betapdf(x,192,820);
plot(x1,y1,'Color',[0.2,0.4,0.8]);
hold on
plot(x,y,'Color',[0.9,0.1,0.1]);
legend('Beta(21,91)','Beta(192,820)')
hold off
xlabel('x');
ylabel('f(x|a,b)');
title('Beta Distribution','Fontsize',12);
```

3. **Code for Question 4**: (This Code is written in R.)

```
rm(list = ls())
graphics.off()
set.seed(234)
alpha <- 3.7
beta  <- 4.8

G <- 1000
burnin <- 100
modeBeta <- (alpha - 1)/(alpha + beta -2)
c <- 100 # To be on the safe side.
oneOverc <- format(1/c, digits = 6)
storeAR <- 0.5
ctrAR <- 0
storeMH <- rep(0, G)
storeMH[1] <- 0.5

#  AR algorithm.
while (length(storeAR) <= G) {
   ctrAR <- ctrAR + 1
   y <- rbeta(1,4,5)
   u <- runif(1)

   if (u <= dbeta(y, 3.7, 4.8)/c*1/dbeta(y, 4, 5) ) { storeAR <- c(storeAR, y) }
   }

trueMean <- format(alpha/(alpha + beta), 3)
cat(paste("True Mean = alpha/(alpha + beta) = ", trueMean, "\n", "\n", sep = ""))

meanAR <- format(mean(storeAR), 3)
cat(paste("Mean estimated by AR = ", meanAR, "\n", "\n", sep = ""))
accRateAR <- format(G/ctrAR, digits = 3) # G divided by total number of trials needed to get
    G trials.


#  MH algorithm.
for (g in 2:(G + burnin))
   {
   y <- rbeta(1,4,5)
   u <- runif(1)
   x <- storeMH[g-1]
   alphaxy <- dbeta(y, 3.7, 4.8)/dbeta(x, 3.7, 4.8) * dbeta(x, 4, 5)/dbeta(y, 4, 5)
   if (u <= alphaxy) {storeMH[g] <- y} else {storeMH[g] <- x}
   }
storeMH <- storeMH[-(1:burnin)] # Eliminate burnin sample.
meanMH <- format(mean(storeMH), 3)
cat(paste("Mean estimated by MH = ", meanMH, "\n", "\n", sep = ""))

ARHist <- hist(storeAR, 40)$intensities
MHHist <- hist(storeMH, 40)$intensities
maxY <- max(c(ARHist, MHHist))
yLimAR <- max(max(maxY), max(dbeta(x, 3.7, 4.8), from = 0, to = 1))
yLimMH <- max(max(maxY), max(dbeta(x, 3.7, 4.8), from = 0, to = 1))
```

```r
   # To calculate acceptance rate for MH.
accMH <- 0
for (g in 2:G) { if (storeMH[g] != storeMH[g - 1]) accMH <- accMH + 1 }
accRateMH <- format(accMH/G, 3)

cat(paste("AR acceptance rate = ", accRateAR, " (1/c = ", oneOverc, ")", "\n", "\n", sep =
    ""))
cat(paste("MH acceptance rate = ", accRateMH, "\n", "\n", sep = ""))

xx <- seq(from = 0, to = 1, length = 100)

plot(1:G, cumsum(storeMH[1:1000])/(1:G), type='l', xlab = "G", ylab = "Running average for
    MH")
abline(h = trueMean, col = "red")

postscript(file = "./q4.pdf", horizontal = F)
op <- par()
op <- par(mfrow = c(2, 2), pty = "s", bty = "l", mar=c(2,4, 1 ,0.5), oma=c(1.5, 2, 1, 2))

# AR f(x) and cg(x)
curve(dbeta(x, 3.7, 4.8), main = "AR algorithm", ylim = c(0, yLimAR), ylab = "f(x), cg(x)")
curve(c*dbeta(x, 4, 5), add = T, lty=2)
legend(x = "topright", lty = c(1, 2), c("f(x)", "cg(x)"))

# AR hist of output and true.
hist(storeAR, 40, freq = FALSE, ylab = "f(x)", xlab = "x", main = "AR algorithm", ylim = c(0,
    yLimAR))
curve(dbeta(x, 3.7, 4.8), add = T)

curve(dbeta(x, 3.7, 4.8), main = "MH algorithm", ylim = c(0, yLimMH), ylab = "f(x), q(x)")
curve(dbeta(x, 4, 5), add = T,lty=2)
legend(x = "topright", lty = c(1, 2), c("f(x)", "q(x)"))

hist(storeMH, 40, freq = FALSE, ylab = "f(x)", xlab = "x", main = "MH algorithm", ylim = c(0,
    yLimMH))
curve(dbeta(x, 3.7, 4.8), add = T)
par(op)
dev.off()
```

Command Line Call for above Script : $ Rscript q4.r

Command Line Output :

```
True Mean = alpha/(alpha + beta) = 0.4352941

Mean estimated by AR = 0.438792

Mean estimated by MH = 0.4295362

Warning message:
In max(c(ARHist, MHHist)) : no non-missing arguments to max; returning -Inf
AR acceptance rate = 0.0103 (1/c = 0.01)

MH acceptance rate = 0.958

pdf
```

4. **Code for Question 5 for n=50**: (This Code is written in R.)

```
# source("AnswerEx7_7n50.r")

rm(list = ls())
graphics.off()
library(tmvtnorm)
library(coda)

# Create data.
set.seed(123)  # To get same values for output.

n <- 50
bet <- rgamma(1, 2, 1)
x <- rnorm(n, 0, 1)
u <- rnorm(n, 0, 1)


y <- bet * x + u

sigxy <- sum(x * y)
sigxx <- sum(x * x)
bethat <- (sigxy - 1 + sqrt( (sigxy - 1)^2 + 4 * sigxx))/(2 * sigxx)

csf <- 1  # Tuning parameter for scale factor.
sf <- csf /(1/bethat^2 + sigxx)

# MH algorithm
G <- 5000
burnin <- 1000

betStore <- rep(0, burnin + G)
betStore[1] <- 1

betPost <- function(bet, y, x) {
   bet * exp(-bet) * exp(-(1/2) * sum((y - bet * x) * (y - bet * x)))
   }

   # Next line uses package tmvtnorm to generate burnin + G candidates.
bet.can <-  rtmvt(burnin + G, mean = bethat, sigma = sf, df = 5, lower = 0, upper = Inf)
for (g in 2:(burnin + G)) {
   bet.cur <- betStore[g-1]
   postRatg <- betPost(bet.can[g], y, x)/betPost(bet.cur, y, x)
   kernRatg <- dt(bet.cur, df = 5)/dt(bet.can[g], df = 5)
   alphaxyg <- postRatg * kernRatg
   v <- runif(1)
   if (v < alphaxyg) betStore[g] <- bet.can[g] else betStore[g] <- betStore[g - 1]
   }

   # Compute acceptance rate.
acc <- 0
betStore <- betStore[-(1:burnin)]
betStore.mcmc <- as.mcmc(betStore)
for (g in 2:G) { if ( betStore[g] != betStore[g-1]) acc <- acc + 1 }
accRate <- format(acc/(G-1), digits = 3)
```

```
    # Outputs
cat(paste("MH acceptance rate = ", accRate, "\n"), sep = "")
cat(paste("True beta = ",format(bethat, digits = 4), "\n"), sep = "")
cat(paste("Summary of MH run, n = ", n, "\n"))
print(summary(betStore.mcmc))

     # Graph

postscript(file = "./q5_50.pdf", horizontal = F)
hist(betStore, 40, freq = F, main = paste("n = ", n), xlab = expression(beta), ylab =
    expression(paste(pi, "(", beta, "|y)")))
dev.off()
```

Command Line Call for above script : $ Rscript q5-50.r

Command Line Output:

```
Loading required package: mvtnorm
Loading required package: Matrix
Loading required package: stats4
Loading required package: gmm
Loading required package: sandwich
MH acceptance rate = 0.743
True beta = 0.757
Summary of MH run, n = 50

Iterations = 1:5000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 5000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

         Mean             SD       Naive SE Time-series SE
     0.765625       0.098516       0.001393       0.001654

2. Quantiles for each variable:

  2.5%    25%    50%    75%  97.5%
0.5736 0.7012 0.7650 0.8275 0.9595

null device
          1
```