

Experiment 1

Shainwat Shah

60004220126

BE Comp 22

Aim! To perform data exploration usg R.

Theory: Introduction to R.

R is a rich environment for statistical computing and has many capabilities to explore data in its base package.

In addition, R contains a ~~etc~~ collection of functions for simulating and summarizing the familiar one-parameter probability.

It is widely used by statisticians, data scientists and researchers in various fields to analyze data.

Perform statistical modeling, and create informative visualizations.

R's popularity stems from flexibility, extensive functionality and active community of users.

Some of the core features that are:-

- i) It is an interpreted language
- ii) It renders vectorized operations
- iii) It has a vibrant community of users and developers
- iv) It offers extensive graphic functionalities

v) The attach() command is used to make variables visible in the R environment.

Conclusion: In this experiment we used programming to explore a dataset of student characteristics and preferences.



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat Shah

Batch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### **Experiment No. 1**

**AIM:-** To Perform Data Exploration using R.

Tasks:

Exploring student's dataset for :

1. Displaying the first ten rows of the data frame.
2. What is your gender?
3. What is your height in inches?
4. Displaying the drinking preferences of the students and plotting their frequencies using a bar chart.
5. Displaying the summary of hours of sleep and plot histogram.
6. Plotting a boxplot of the hours of sleep for each level of gender.
7. Summary of haircut for each level of gender.
8. Constructing a histogram of DVDs and display the summary.
9. Constructing a frequency table of the individual values of DVDs that were observed.
10. Constructing parallel boxplots of the heights using the Gender variable.

Code:

Loading the dataset

```
library('LearnBayes')  
data(studentdata)
```

Performing Data Exploration:

```
1. print(studentdata[1:10,])  
2. table(studentdata$Gender)  
3. table(studentdata$Height)  
4. table(studentdata$Drink)  
barplot(table(studentdata$Drink),xlab="Drink",ylab="Count")
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

5. hours.of.sleep = studentdata\$WakeUp - studentdata\$ToSleep  
summary(hours.of.sleep) hist(hours.of.sleep,main="")
6. boxplot(hours.of.sleep~studentdata\$Gender,ylab="Hours of Sleep")
7. female.Haircut=studentdata\$Haircut[studentdata\$Gender=="female"]  
summary(female.Haircut)  
male.Haircut=studentdata\$Haircut[studentdata\$Gender=="male"]  
summary(male.Haircut)
8. hist(studentdata\$Dvds) print(summary(studentdata\$Dvds))
9. print(table(studentdata\$Dvds)) barplot(table(studentdata\$Dvds))
10. boxplot(studentdata\$Height~studentdata\$Gender)



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Computer Engineering

Output:

1.

Student	Height	Gender	Shoes	Number	Dvds	ToSleep	WakeUp	Haircut	Job	Drink
1	1	67 female	10	5	10	-2.5	5.5	60	30.0	water
2	2	64 female	20	7	5	1.5	8.0	0	20.0	pop
3	3	61 female	12	2	6	-1.5	7.5	48	0.0	milk
4	4	61 female	3	6	40	2.0	8.5	10	0.0	water
5	5	70 male	4	5	6	0.0	9.0	15	17.5	pop
6	6	63 female	NA	3	5	1.0	8.5	25	0.0	water
7	7	61 female	12	3	53	1.5	7.5	35	20.0	water
8	8	64 female	25	4	20	0.5	7.5	25	0.0	pop
9	9	66 female	30	3	40	-0.5	7.0	30	25.0	water
10	10	65 male	10	7	22	2.5	8.5	12	0.0	milk

2.

female	male
435	222

3.

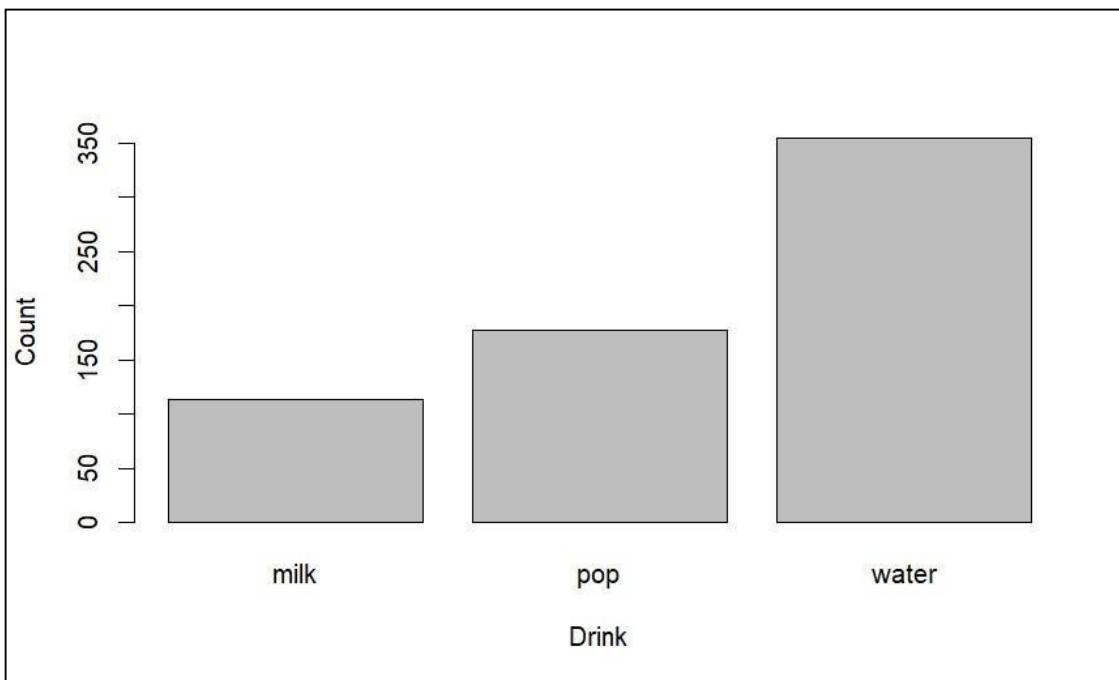
54	55	56	57.75	58	58.5	59	59.75	60	60.5	61	61.5	62
2	1	3	1	2	1	3	1	18	3	23	2	49
62.5	63	63.5	64	64.5	65	65.5	66	66.5	66.929	67	67.5	68
2	41	1	66	3	52	2	51	3	1	54	4	45
68.5	69	69.5	70	70.5	71	71.5	71.75	72	72.5	73	74	75
1	28	3	46	2	36	1	1	41	1	17	16	7
75.5	76	77	78	79	84							
1	5	2	2	2	1							

4.

milk	pop	water
113	178	355

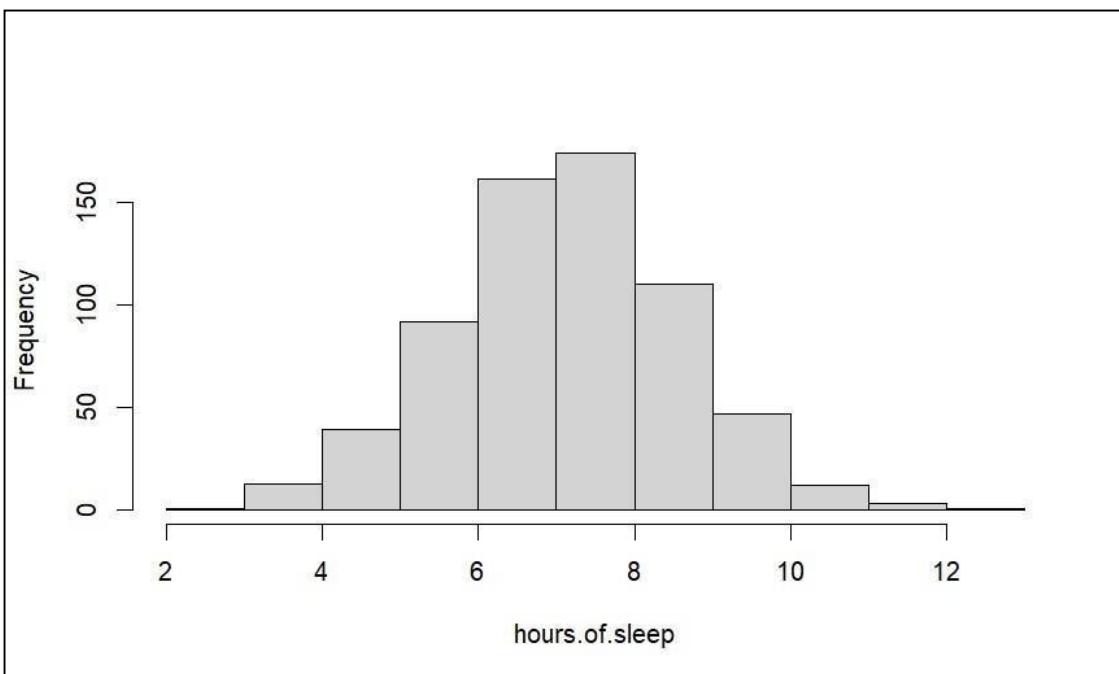


Department of Computer Engineering



5

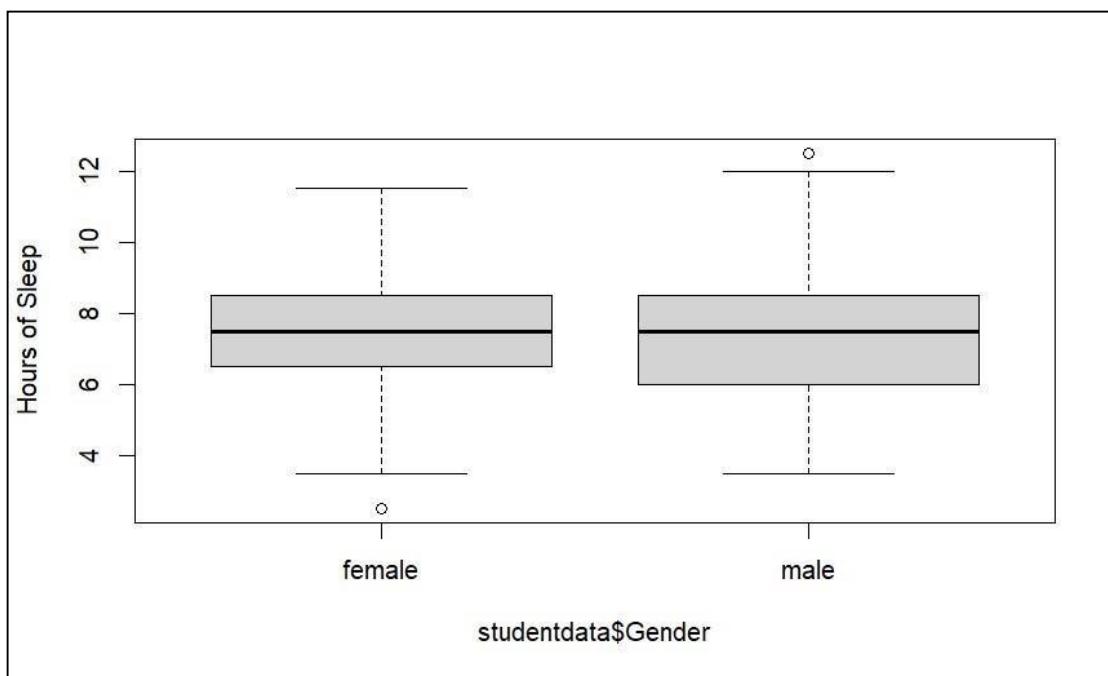
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
2.500	6.500	7.500	7.385	8.500	12.500	4



6.



Department of Computer Engineering



7

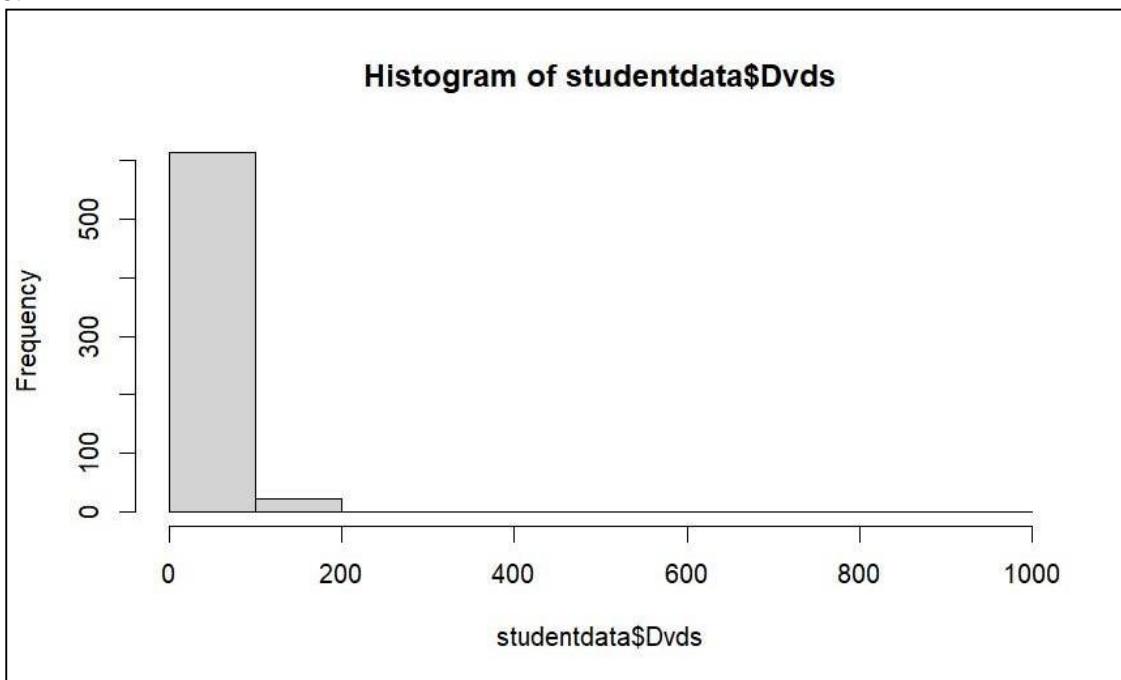
	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
female	0.00	15.00	25.00	34.08	45.00	180.00	19

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
male	0.00	0.00	12.00	10.54	15.00	75.00	1



Department of Computer Engineering

8.



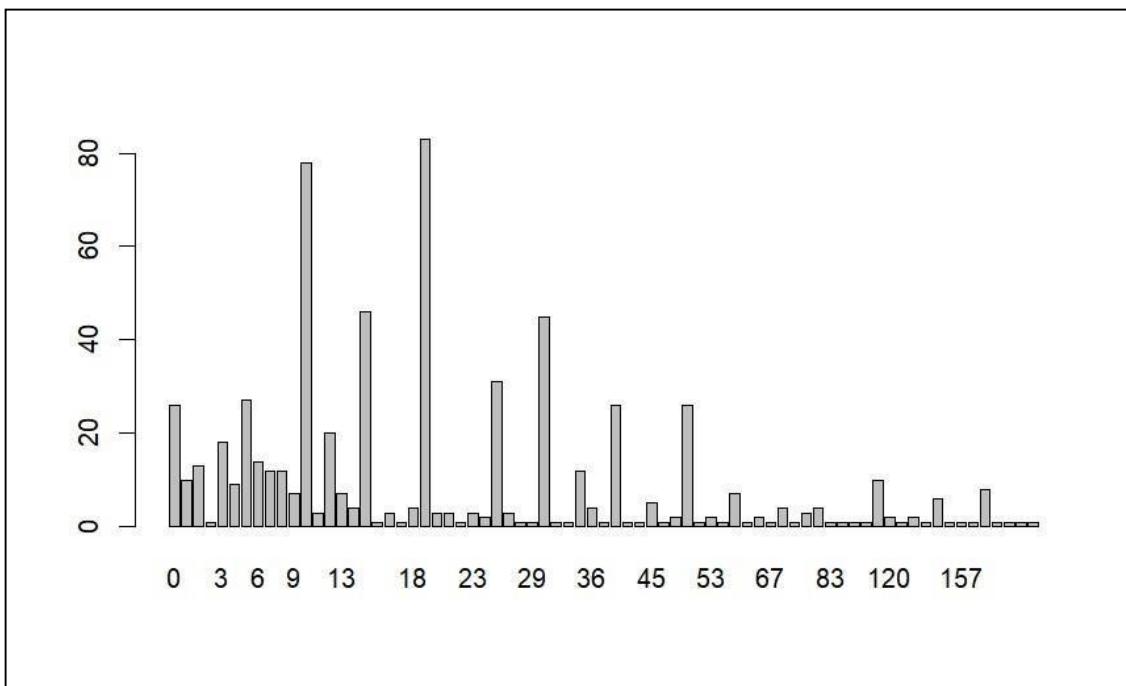
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.00	10.00	20.00	30.93	30.00	1000.00	16

9

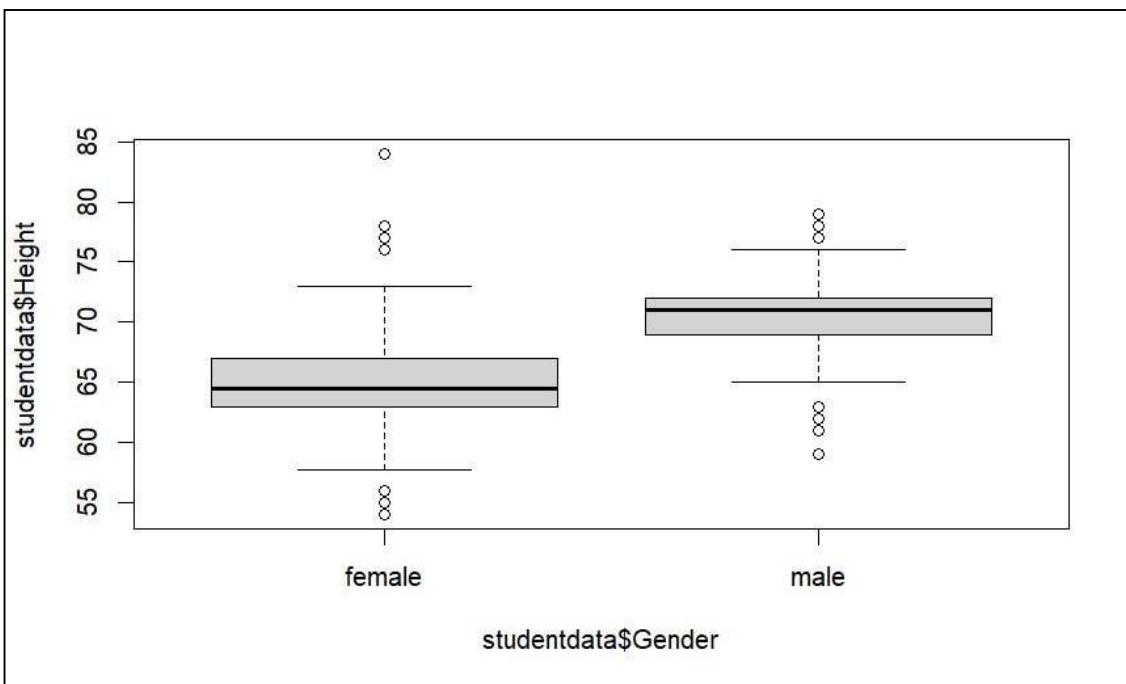
0	1	2	2.5	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
26	10	13	1	18	9	27	14	12	12	7	78	3	20	7	4	46	1	3
17.5	18	20	21	22	22.5	23	24	25	27.5	28	29	30	31	33	35	36	37	40
1	4	83	3	3	1	3	2	31	3	1	1	45	1	1	12	4	1	26
41	42	45	46	48	50	52	53	55	60	62	65	67	70	73	75	80	83	85
1	1	5	1	2	26	1	2	1	7	1	2	1	4	1	3	4	1	1
90	97	100	120	122	130	137	150	152	157	175	200	250	500	900	1000			
1	1	10	2	1	2	1	6	1	1	1	8	1	1	1	1			



Department of Computer Engineering



10.



Experiment 2

Shashwat Shab

60004220126

BE Comps CSE

Aim : To implement a Discrete prior , Beta , Rprior , histogram , predicted models using R.

Theory : Discrete prior -

A discrete prior is a probability distribution over a finite or countably infinite set of possible values. It represents the belief about a parameter before any data is observed. Discrete priors are commonly used for categorical variables.

Beta Prior - A beta prior is a probability distribution over the interval  $(0, 1)$ . It is commonly used as a prior for parameters that represent proportions or probabilities. The beta prior is characterized by 2 hyperparameters,

$$P(\theta | \alpha, \beta) = (\beta^{\alpha} \Gamma(\alpha + \beta))^{-1} \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

where  $\Gamma(x, p)$  is the beta function

The beta prior is a conjugate prior for the binomial distribution.

histogram prior and the prediction model are the statistical models.

Conclusion : In this experiment, we learned that how to implement different types of prior and posterior distributions for a binomial parameter  $p$  using R.



Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

**Experiment No. 2**

**Aim:**

To implement a Discrete Prior, Beta Prior, Histogram Prior, Prediction Model using R.

Code:

Loading the dataset

```
library('LearnBayes')
```

Using a Discrete Prior

```
library('LearnBayes') p <- seq(0.05, 0.95, by =  
0.1) prior <- c(1, 5.2, 8, 7.2, 4.6, 2.1, 0.7, 0.1, 0,  
0) prior <- prior / sum(prior) plot(p, prior, type =  
"h", ylab="Prior Probability")
```

The posterior for p

```
data <- c(11, 16) post <-  
pdisc(p, prior, data)  
round(cbind(p, prior,  
post),2)
```

```
library(lattice)
```

```
PRIOR <- data.frame("prior", p, prior)
```

```
POST <- data.frame("posterior", p, post)
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Computer Engineering

```
names(PRIOR) <- c("Type", "P", "Probability") names(POST) <-  
c("Type", "P", "Probability") data <- rbind(PRIOR, POST) xyplot(Probability ~ P |  
Type, data=data, layout=c(1,2), type="h", lwd=3, col="black")
```

### Using a Beta Prior

```
quantile2 <- list(p=.9, x=.5)  
  
quantile1 <- list(p=.5, x=.3)  
  
(ab <- beta.select(quantile1,quantile2))
```

### Bayesian triplot

```
a <- ab[1] b <- ab[2] s <- 11 f <- 16 curve(dbeta(x, a + s, b + f), from=0, to=1,  
xlab="p", ylab="Density", lty=1, lwd=4) curve(dbeta(x, s + 1, f + 1),  
add=TRUE, lty=2, lwd=4) curve(dbeta(x, a, b), add=TRUE, lty=3, lwd=4)  
legend(.7, 4, c("Prior", "Likelihood", "Posterior"), lty=c(3, 2, 1), lwd=c(3, 3, 3))
```

### Posterior summaries

```
1 - pbeta(0.5, a + s, b + f)  
  
qbeta(c(0.05, 0.95), a + s, b + f)
```

### Simulating from posterior

```
ps <- rbeta(1000, a + s, b + f)  
hist(ps, xlab="p")
```



Shri Vile Parle Kelavani Mandal's

## DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

```
sum(ps >= 0.5) / 1000
```

```
quantile(ps, c(0.05, 0.95))
```

Using a Histogram Prior

```
midpt <- seq(0.05, 0.95, by = 0.1) prior <- c(1, 5.2, 8, 7.2, 4.6, 2.1, 0.7, 0.1, 0, 0)
```

```
prior <- prior / sum(prior) curve(histprior(x, midpt, prior), from=0, to=1,
```

```
ylab="Prior density", ylim=c(0, .3))
```

```
curve(histprior(x,midpt,prior) * dbeta(x, s + 1, f + 1), from=0, to=1, ylab="Posterior  
density")
```

```
p <- seq(0, 1, length=500) post <- histprior(p, midpt,  
prior) * dbeta(p, s + 1, f + 1) post <- post / sum(post)  
ps <- sample(p, replace = TRUE, prob = post) hist(ps,  
xlab="p", main="")
```

### Prediction Model

#### Discrete prior approach

```
p <- seq(0.05, 0.95, by=.1) prior <- c(1, 5.2,  
8, 7.2, 4.6, 2.1, 0.7, 0.1, 0, 0) prior <- prior /  
sum(prior) m <- 20 ys <- 0:20
```

```
pred <- pdiscp(p, prior, m, ys)  
cbind(0:20, pred)
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

#### Continuous prior approach

```
ab <- c(3.26, 7.19) m <-
20 ys <- 0:20 pred <-
pbetap(ab, m, ys)
```

#### Simulating predictive distribution

```
p <- rbeta(1000, 3.26, 7.19)

y <- rbinom(1000, 20, p)

table(y)

freq <- table(y) ys <- as.integer(names(freq)) predprob <- freq /
sum(freq) plot(ys, predprob, type="h", xlab="y", ylab="Predictive
Probability")

dist <- cbind(ys, predprob)
```

#### Construction of a prediction interval

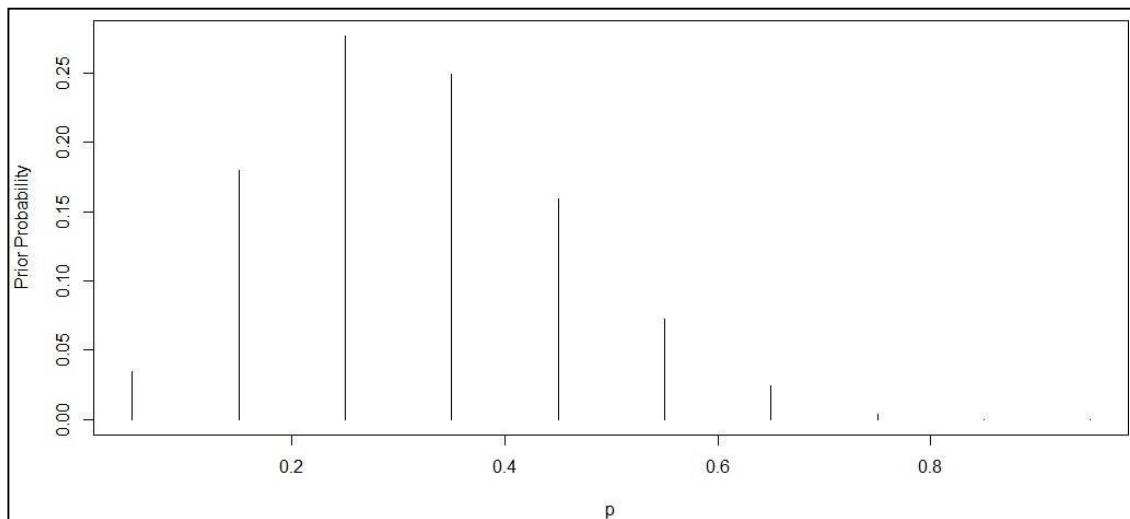
```
covprob <- .9
discint(dist, covprob)
```



Department of Computer Engineering

**Output:**

Using a Discrete Prior

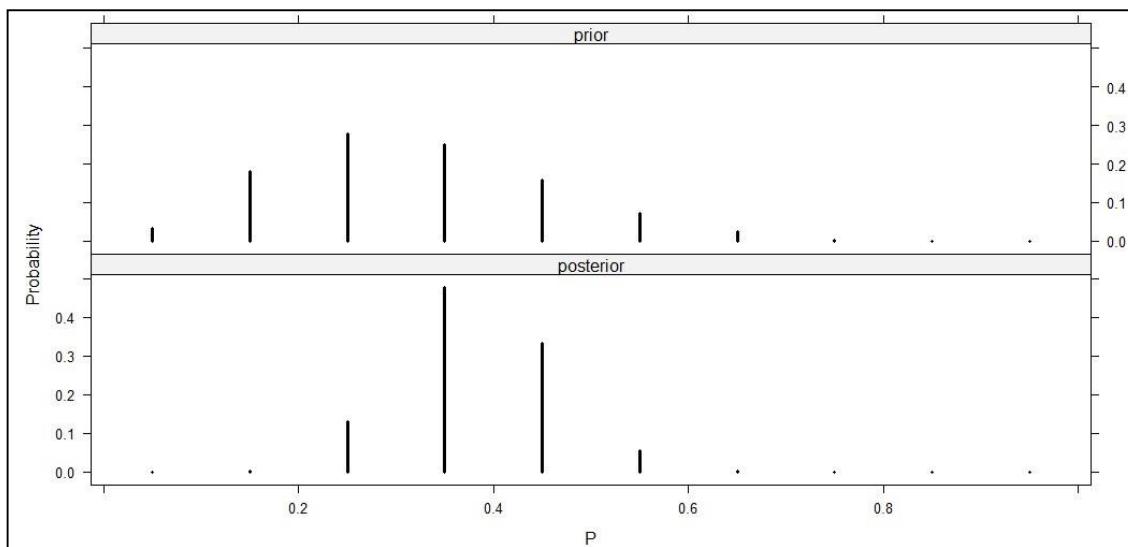


The posterior for p

	p	prior	post
[1, ]	0.05	0.03	0.00
[2, ]	0.15	0.18	0.00
[3, ]	0.25	0.28	0.13
[4, ]	0.35	0.25	0.48
[5, ]	0.45	0.16	0.33
[6, ]	0.55	0.07	0.06
[7, ]	0.65	0.02	0.00
[8, ]	0.75	0.00	0.00
[9, ]	0.85	0.00	0.00
[10, ]	0.95	0.00	0.00



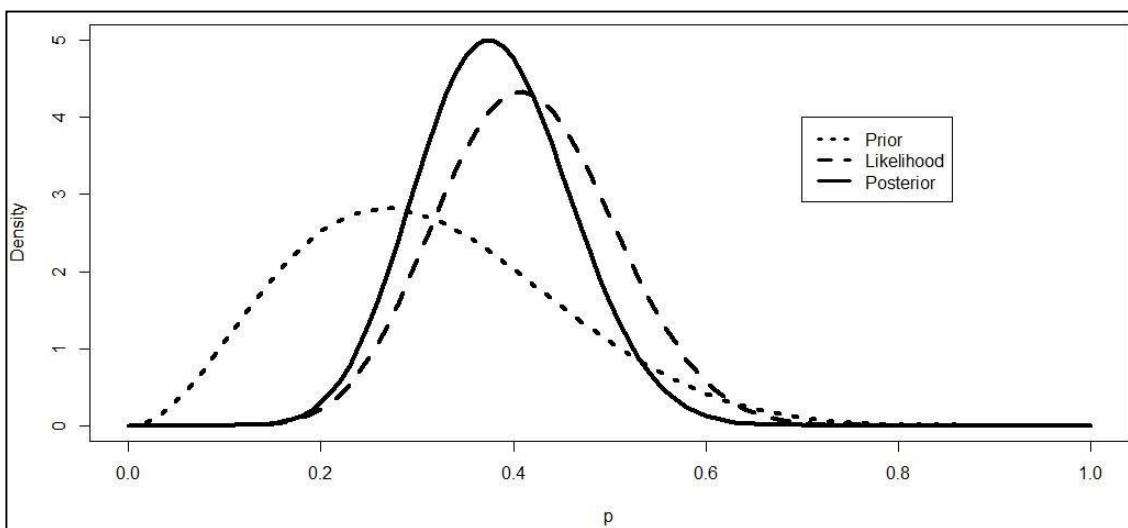
### Department of Computer Engineering



Using a Beta Prior

3.26 7.19

Bayesian triplot



Posterior summaries

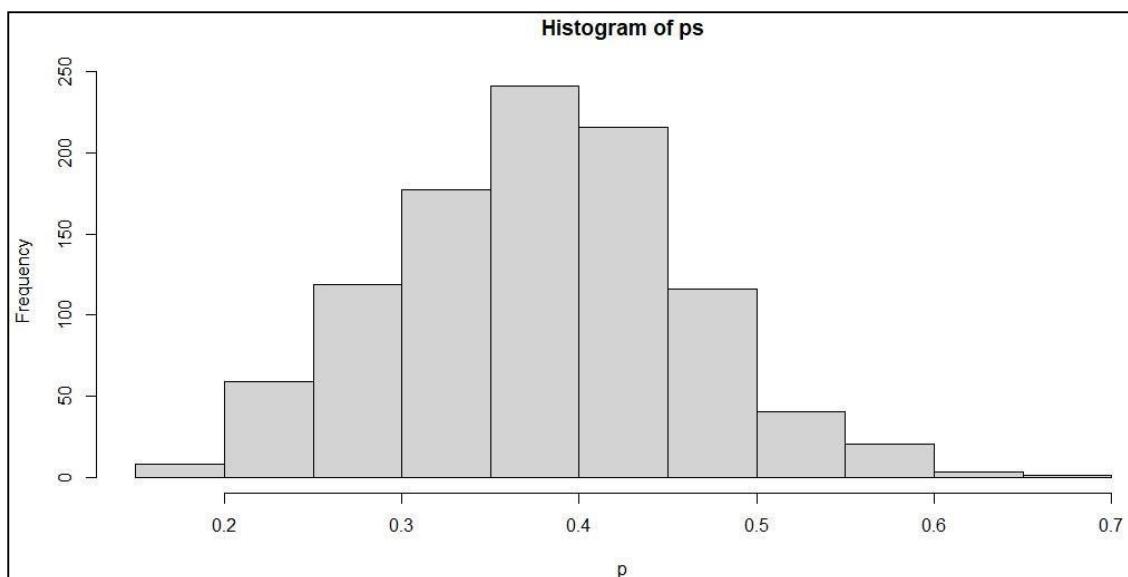
0.0690226

0.2555267 0.5133608

Simulating from posterior



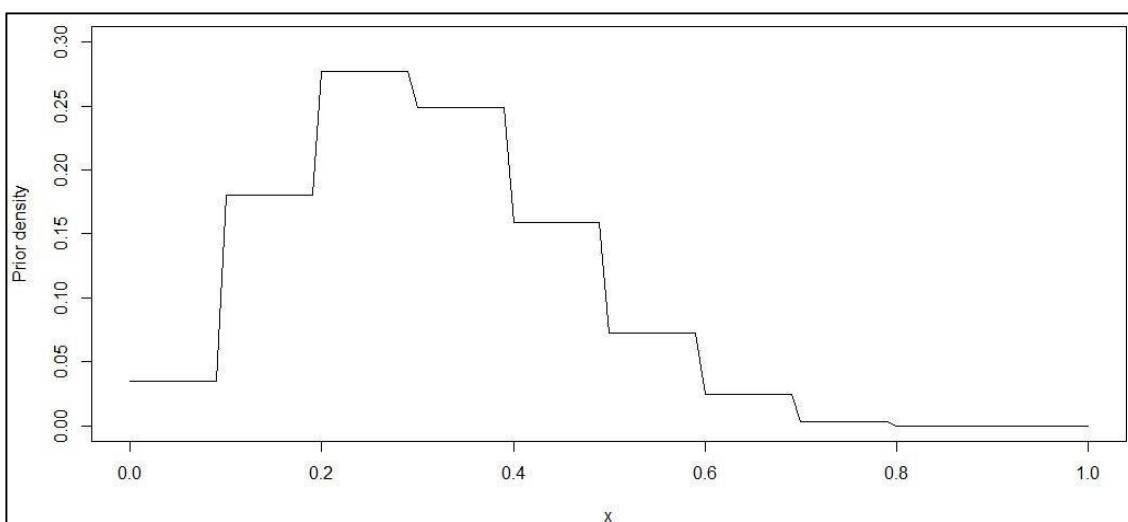
Department of Computer Engineering



0.064

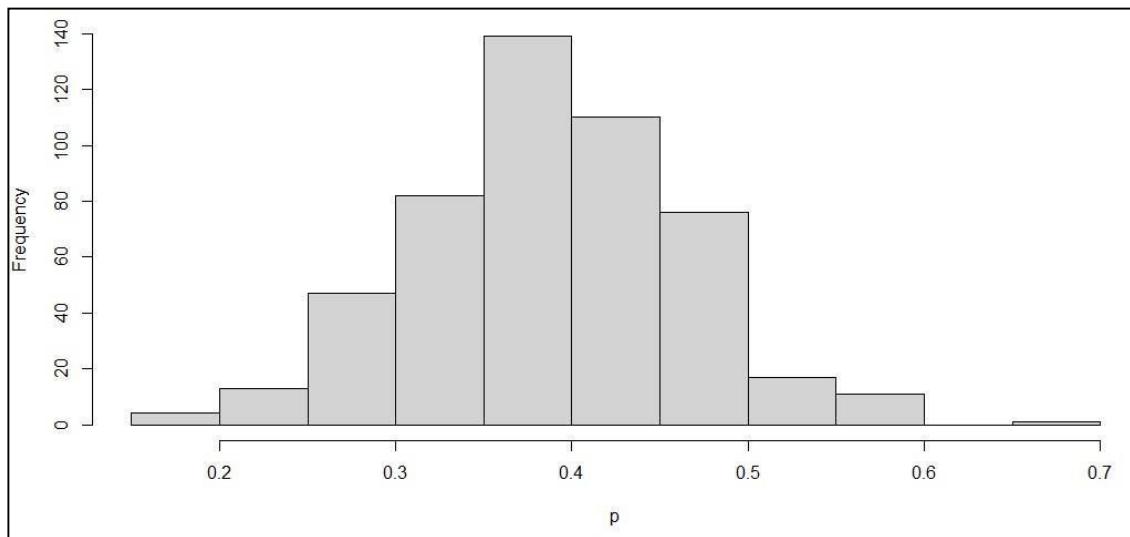
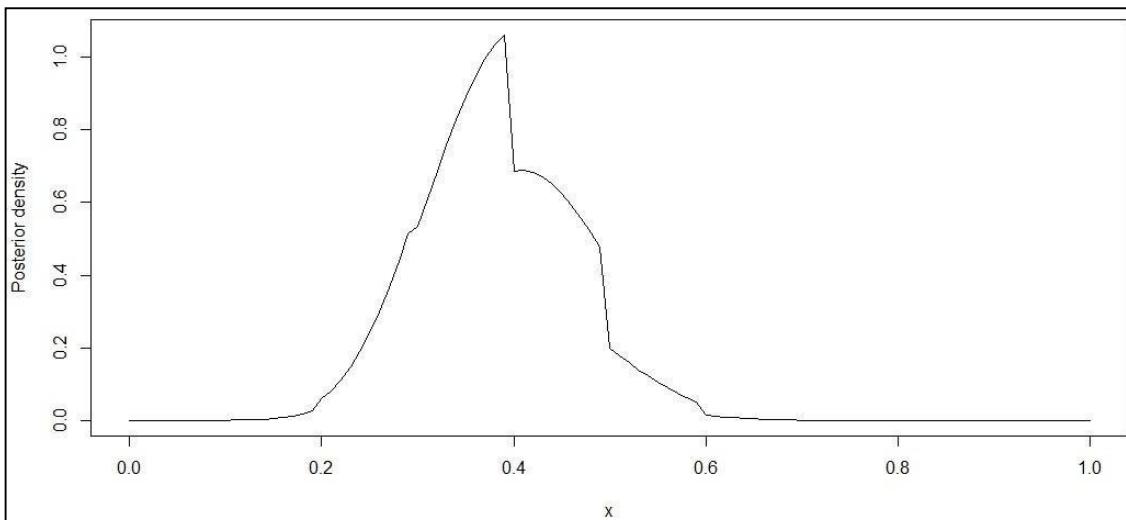
5%	95%
0.2373968	0.5094293

Using a Histogram Prior





Department of Computer Engineering





### Department of Computer Engineering

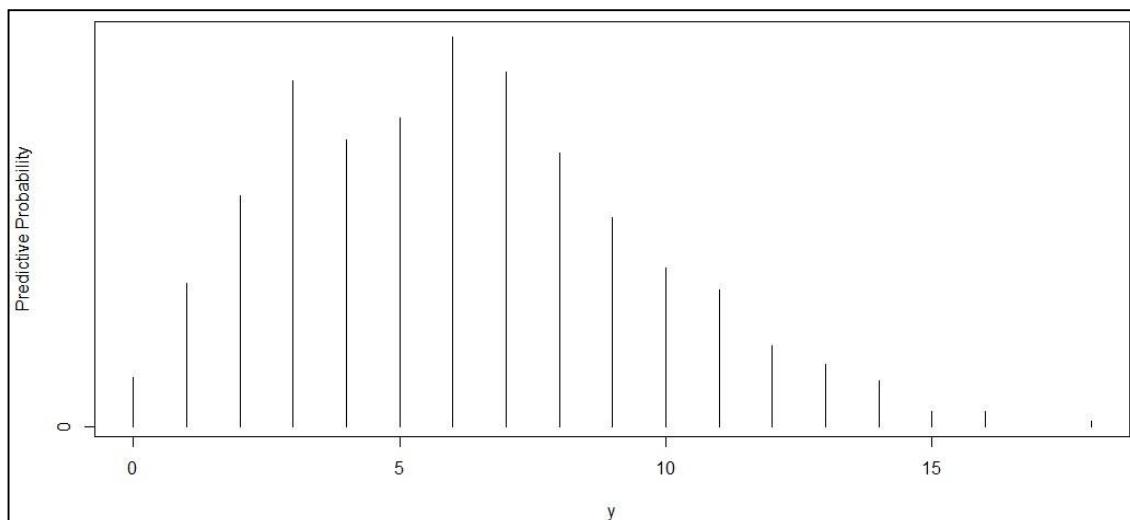
#### Prediction Model

Discrete prior approach

	pred
[1,]	0 2.030242e-02
[2,]	1 4.402694e-02
[3,]	2 6.894572e-02
[4,]	3 9.151046e-02
[5,]	4 1.064393e-01
[6,]	5 1.124487e-01
[7,]	6 1.104993e-01
[8,]	7 1.021397e-01
[9,]	8 8.932837e-02
[10,]	9 7.416372e-02
[11,]	10 5.851740e-02
[12,]	11 4.383668e-02
[13,]	12 3.107700e-02
[14,]	13 2.071698e-02
[15,]	14 1.284467e-02
[16,]	15 7.277453e-03
[17,]	16 3.667160e-03
[18,]	17 1.575535e-03
[19,]	18 5.381536e-04
[20,]	19 1.285179e-04
[21,]	20 1.584793e-05

Simulating predictive distribution

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18
	16	46	74	111	92	99	125	114	88	67	51	44	26	20	15	5	5	2





Department of Computer Engineering  
Academic Year 2023-2024

Construction of a prediction interval

```
$prob
 11
0.911

$set
 1  2   3   4   5   6   7   8   9 10 11
 1  2   3   4   5   6   7   8   9 10 11
```

### Experiment 3

Shashwat Shah

60004220126

BE compu C22

Aim: To apply markov chain Monte Carlo (MCMC) method to estimate the area of unit circle (hit-miss)

Theory: Markov Chain Monte Carlo (MCMC) method

Markov Chain Monte Carlo (MCMC) is a computational method for approximating probability distributions by generating MCMC methods are widely used in Bayesian computing to sample from posterior.

There are several different MCMC algorithms but they all show the same general structure. Each algorithm starts with an initial state, which may or may not be from the desired distribution. The algorithm then generates a sequence of new states by randomly selecting a new state from a candidate distribution. The candidate distribution is typically designed to be close to the desired distribution but it may not be exactly the same the new state is accepted.

MCMC methods are powerful tool for Bayesian computing and they have been used to solve problems in statistics, science and engineering

Conclusion - In this experiment, we learnt how to use the MCMC method to estimate area of unit circle & thus approximate values of  $\pi$ . We used 2 loops code to approximate random points. ~~title~~



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### Experiment No. 3

#### **Aim:**

To apply Markov Chain Monte Carlo (MCMC) method to estimate the area of Unit Circle.  
[Hit-and-miss example: to calculate the area of a circle][The area of a circle is computed as  $\pi R^2$ ]

Code:

Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')
sns.set_context('paper')
```

Part 1

```
radius = 1
```

```
N = 100000 #Use 10000 points
```

```
X = np.random.uniform(low=-radius, high=radius, size=N) # Random numbers from -1 to 1
```

```
Y = np.random.uniform(low=-radius, high=radius, size=N)
```

```
# calculate the distance from the center
```

```
R = np.sqrt(X**2+Y**2);
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



## Department of Computer Engineering

```
box_area =(2.0*radius)**2    # This is the big
box_is_point_inside = R<radius
N_inside=np.sum(is_point_inside) circle_area =
box_area*N_inside/N

plt.scatter(X,Y, c=is_point_inside, s=5.0, edgecolors='none', cmap=plt.cm.Paired)
plt.axis('equal')

print "Area of the circle = ", circle_area print
"pi = ", circle_area/radius**2
```

### Part 2 (Alternative)

```
# Dan's think-through

N = 100000.0 # number of points
radius = 1.0 # radius of circle

# generate points in circumscribing square
X = np.random.uniform(low=-radius, high=radius, size=int(N))
Y = np.random.uniform(low=-radius, high=radius, size=int(N))

origin_dist_sqr = X*X + Y*Y is_in_circle
= origin_dist_sqr < radius ** 2 in_circle
= np.sum(is_in_circle)
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

```
square_area = (2 * radius) ** 2 circle_area = in_circle / N *
square_area pi_approx = circle_area/radius ** 2 #
manipulate A = pi*r**2

plt.scatter(X, Y, c=is_in_circle, edgecolors='none', cmap=plt.cm.Blues)
plt.axis('equal')

print 'Circle area:', circle_area print 'Pi approximation with %4s trials is
%.6s' % (str(int(N)), pi_approx)
```

**Output:**

Part 1



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

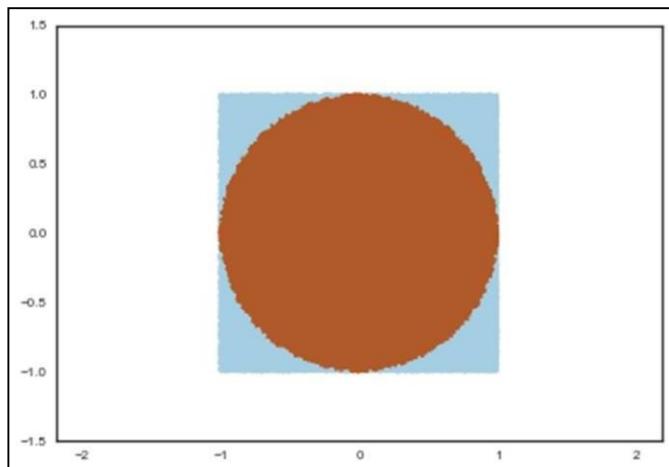
NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

Area of the circle = 3.14436

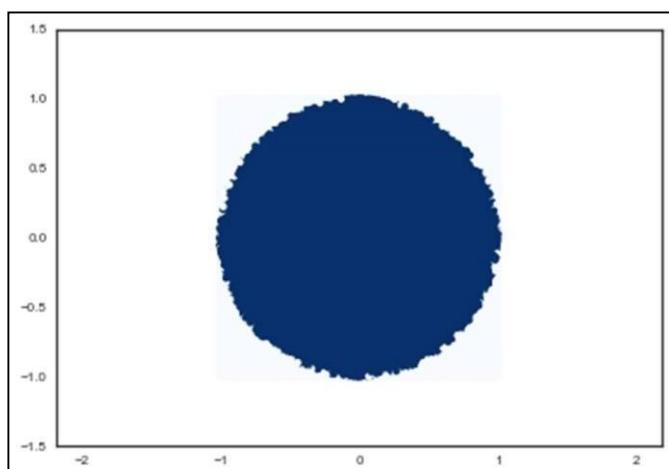
$\pi$  = 3.14436



Part 2 (Alternative)

Circle area: 3.14884

Pi approximation with 100000 trials is 3.1488



## Experiment 4

Shashwat Singh

60004220126

BE Comp C2

Aim: To implement a model for a bioassay experiment.

Consider data from getoor et 2003, where we obtain  
a dose level (in  $\text{log g/ml}$ ), the number of animals  
and the number of deaths for each of the 4 groups.

Theory: Multiparameter models.

These are statistical methods that involve multiple parameters at unknown quantities. These models are commonly used in various fields, including robotics, machine learning and scientific research.

In Bayesian computing, multiparameter models are analyzed using Bayesian methods which incorporate prior belief about the parameters into the analysis and updates this belief in a comprehensive framework for handling uncertainty in multiparameter models.

Logistic Regression: A statistical model used for binary classification where the goal is to predict a binary outcome yes or no. success or failure. Based on a set of independent variables.

A Bioassay Experiment:

These are used to determine the dose-response relationship for a biological agent such as drug, toxin. These experiments typically involve exposing a group of organisms to different doses of the agent & measuring the biological response such as life or death.

Conclusion: In this experiment, we implemented a Bayesian model for a binary exposure using R. The model fitted the data well, capturing the uncertainty in the parameter estimate. The results demonstrated the utility of Bayesian methods in interpreting biosafety data and the influence



Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

**Experiment No. 4**

**Aim:**

Implement a model for A Bioassay Experiment: consider data from Gelman et al. (2003), where one observes a dose level (in log g/ml), the number of animals, and the number of deaths for each of four groups.

Code:

Importing Libraries

```
library("LearnBayes")
```

Bayesian fitting of a logistic model using data from a dose-response experiment.

```
x <- c(-0.86, -0.3, -0.05,
      0.73) n <- c(5, 5, 5, 5) y <-
      c(0, 1, 3, 5) data <- cbind(x,
      n, y)

glmdata <- cbind(y, n - y) results <-
      glm(glmdata ~ x, family = binomial)
summary(results)

a1.b1 <- beta.select(list(p=.5, x=.2),
      list(p=.9, x=.5)) a2.b2
<- beta.select(list(p=.5, x=.8),
```



Shri Vile Parle Kelavani Mandal's

**DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Computer Engineering

```
list(p=.9, x=.98))

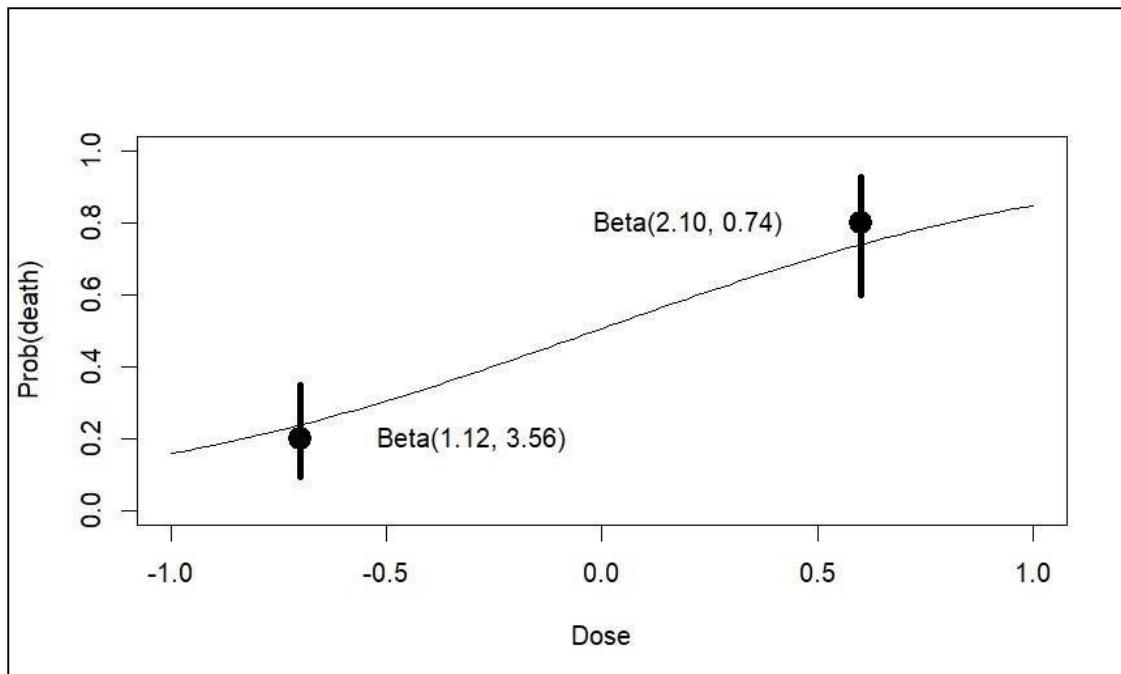
prior <- rbind(c(-0.7, 4.68, 1.12),
c(0.6, 2.10, 0.74)) data.new <-
rbind(data, prior)

plot(c(-1,1), c(0, 1), type="n",
xlab="Dose", ylab="Prob(death)") lines(
0.7 * c(1, 1), qbeta(c(.25, .75),
a1.b1[1], a1.b1[2]), lwd=4) lines(0.6 *
c(1, 1), qbeta(c(.25, .75), a2.b2[1],
a2.b2[2]), lwd=4) points(c(-0.7, 0.6),
qbeta(.5, c(a1.b1[1], a2.b2[1]),
c(a1.b1[2], a2.b2[2])), pch=19,
cex=2) text(-0.3, .2, "Beta(1.12, 3.56)")
text(.2, .8, "Beta(2.10, 0.74)") response <-
rbind(a1.b1, a2.b2) x <- c(-0.7, 0.6) fit <-
glm(response ~ x, family = binomial)

curve(exp(fit$coef[1] + fit$coef[2] * x) /
(1 + exp(fit$coef[1] + fit$coef[2] * x)),
add=T)
```

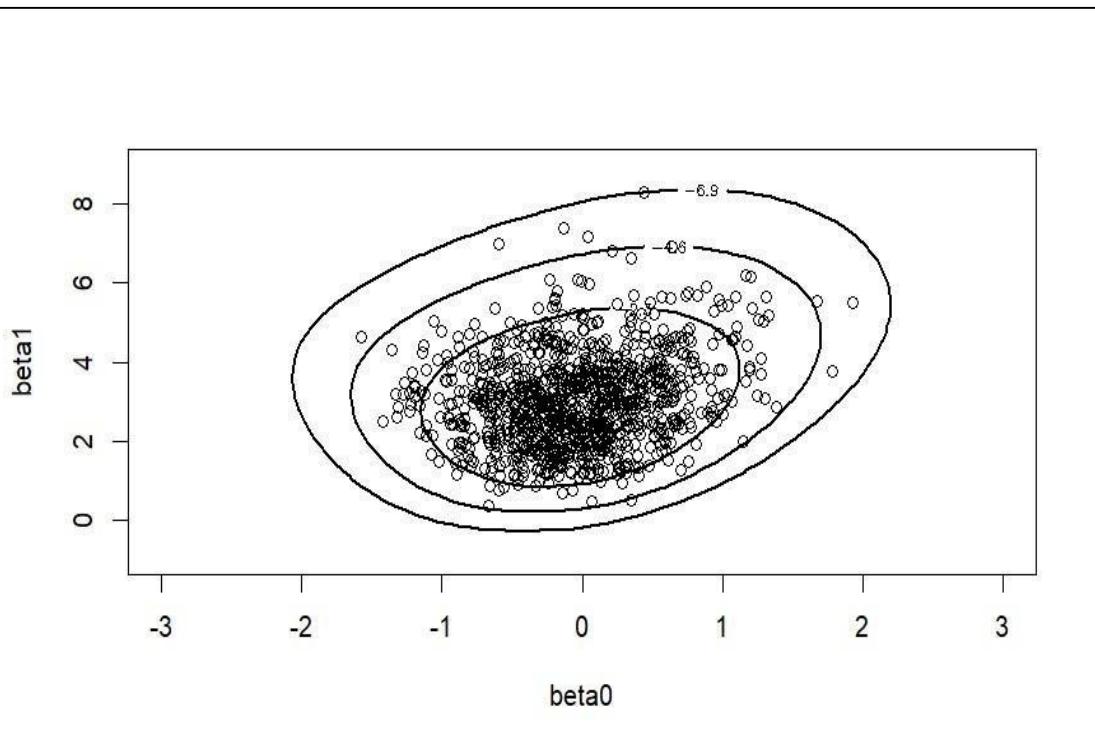
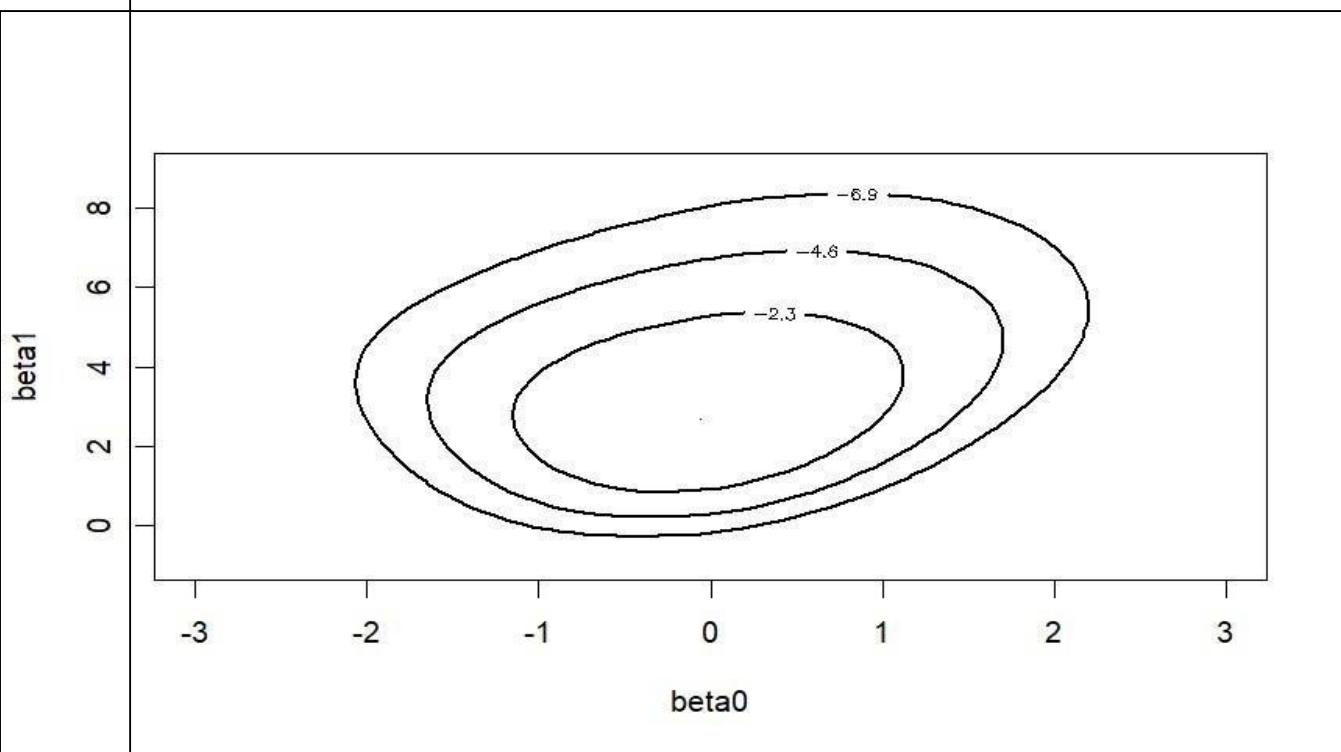
## Output:

```
Call:  
glm(formula = glmdata ~ x, family = binomial)  
  
Coefficients:  
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)  0.8466    1.0191   0.831   0.406  
x            7.7488    4.8728   1.590   0.112  
  
(Dispersion parameter for binomial family taken to be 1)  
  
Null deviance: 15.791412 on 3 degrees of freedom  
Residual deviance: 0.054742 on 2 degrees of freedom  
AIC: 7.9648  
  
Number of Fisher Scoring iterations: 7
```



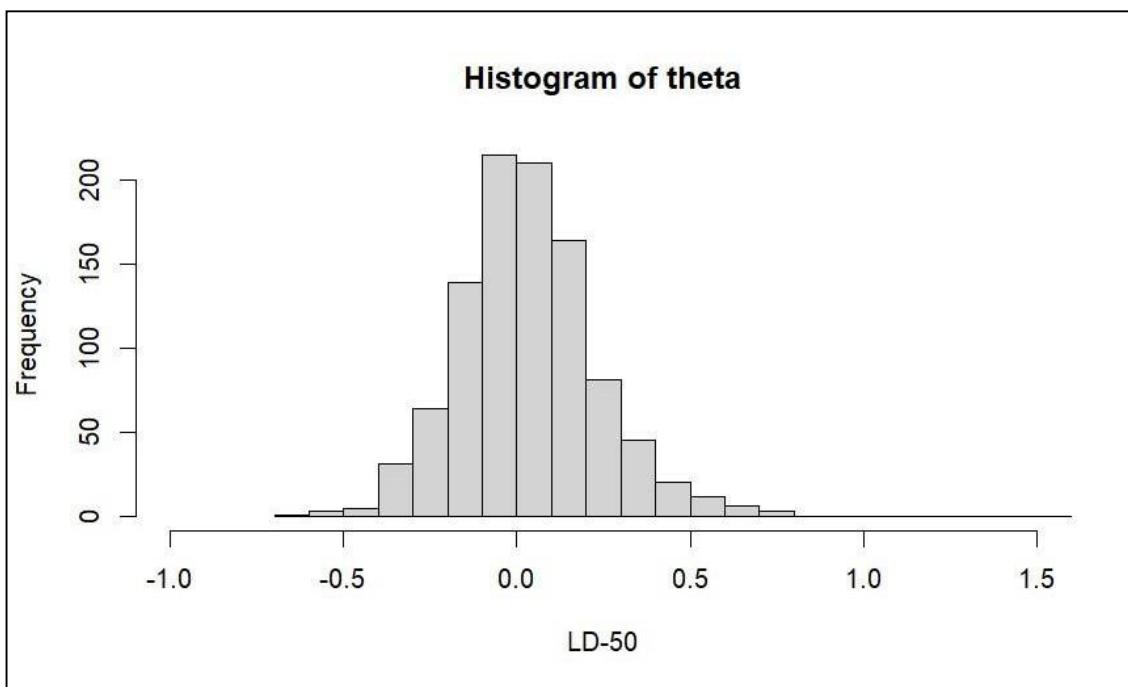
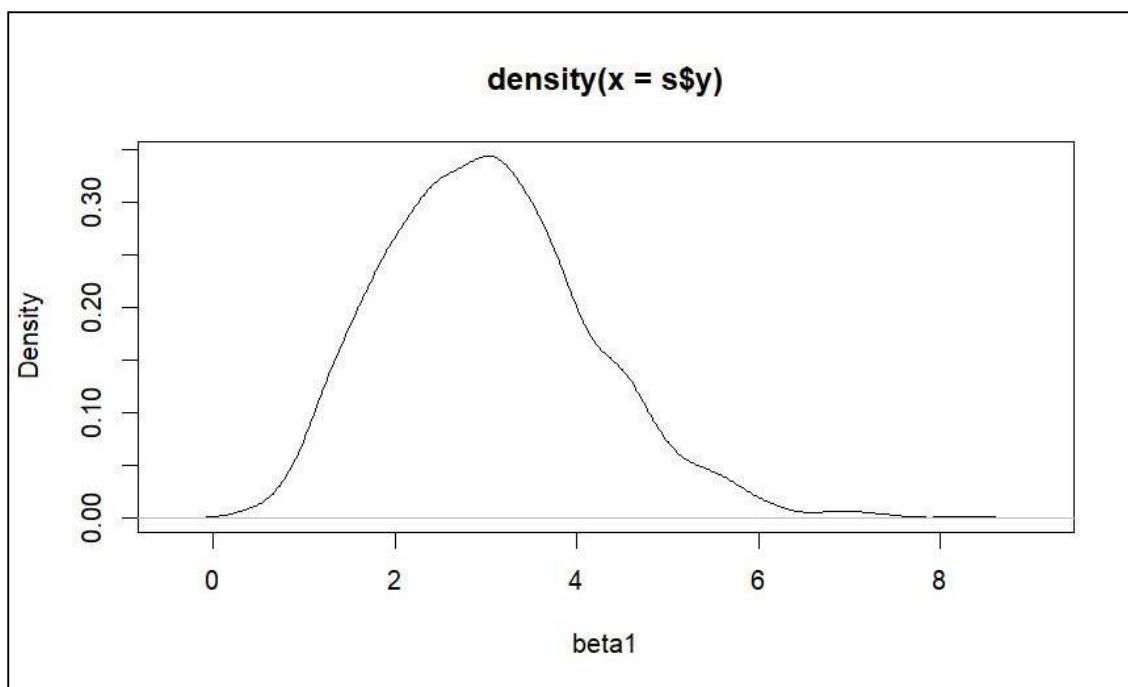


Department of Computer Engineering





Department of Computer Engineering



2.5%	97.5%
-0.3251418	0.4707840

## Experiment 5

Shashwat Shah

6000422016

BE Comm C22

Aim: To implement logistic modeling. A motor department is interested in exploring the relationship between students scores on the Act test, a Stanley College entrance exam, and their success (getting A or B) in a business.

### Theory: Bayesian Logistic Models

Bayesian logistic models is a statistical approach used to model the relationship between a binary outcome variable and one or more predictor variables.

Model description.

#### a) Logistic Regression model.

The model specifies the relationship between the log-odds of the probability of success ( $\pi_i$ ) and the predictor variable,

$$\log \left( \frac{\pi_i}{1 - \pi_i} \right) = \beta_0 + \beta_1 x_i, \text{ where}$$

$\beta_0$  is the intercept and  $\beta_1$  is the coefficient associated with the predictor variable.

$$\pi_i = \frac{e^{(\beta_0 + \beta_1 x_i)}}{1 + e^{(\beta_0 + \beta_1 x_i)}}$$

Conclusion - In conclusion, we learnt how to use Bayesian methods to model binary data and how to assess the model.



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### **Experiment No. 5**

#### **Aim:**

Implement Logistic Modeling: A math department is interested in exploring the relationship between students' scores on the ACT test, a standard college entrance exam, and their success (getting an A or a B) in a business calculus class. Data were obtained for a sample of students.

Code:

Importing Libraries:

```
library(brms)
library(ggplot2)
```

Generate Data for given Scenario:

```
set.seed(123) n <- 100 #
Number of students
ACT_scores <- rnorm(n, mean = 25, sd = 5)
Success <- ifelse(ACT_scores + rnorm(n) > 25, "A", "B")
calculus_data <- data.frame(ACT_scores, Success)
```

Data Exploration and Visualization:

```
ggplot(calculus_data, aes(x = ACT_scores, fill = Success)) +
+   geom_histogram(binwidth = 2, position = "identity", alpha = 0.7) +
+   labs(title = "Distribution of ACT Scores by Success",
+        x = "ACT Scores", y = "Frequency",
+        fill = "Success")
```

Logistic Regression Modeling using Bayesian Approach:



## Department of Computer Engineering

```
calculus_data$Success_binary <- as.numeric(calculus_data$Success == "A") model <-
  brm(Success_binary ~ ACT_scores, data = calculus_data, family = bernoulli())
summary(model)
```

### Posterior Predictive Checks:

```
posterior_preds <- posterior_predict(model)

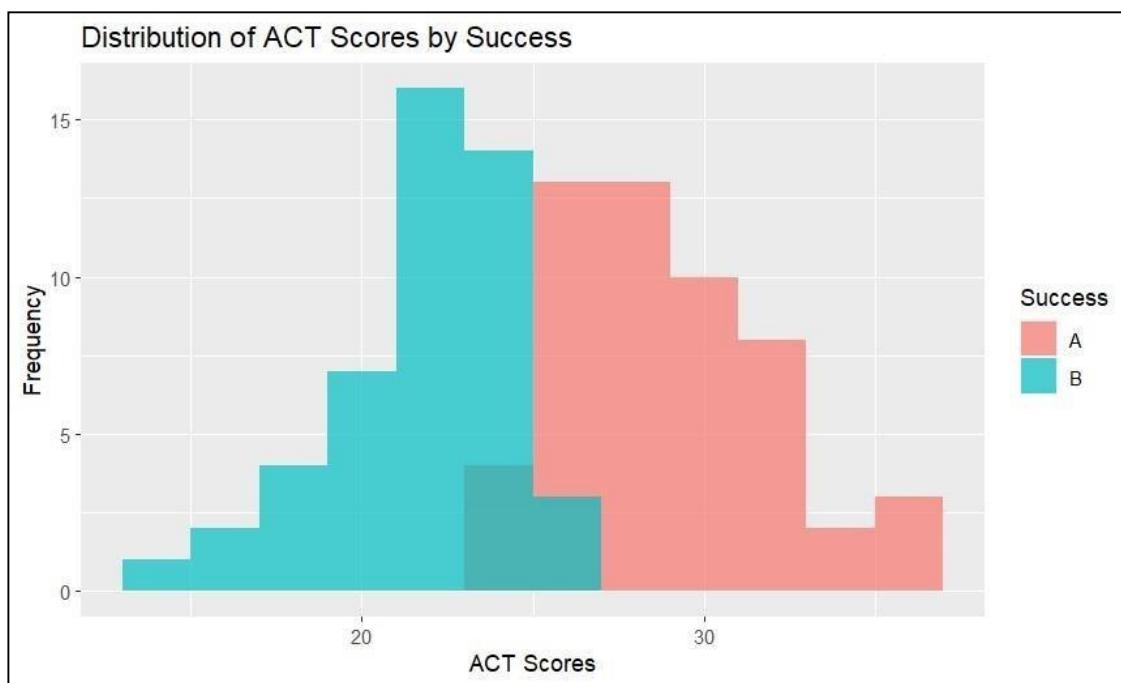
ggplot(calculus_data, aes(x = ACT_scores, y = colMeans(posterior_preds), color =
  Success)) +
  +   geom_point() +
  +   geom_line(aes(y = colMeans(posterior_preds) - 1.96 * sd(posterior_preds)), linetype =
  "dashed", alpha = 0.5) +
  +   geom_line(aes(y = colMeans(posterior_preds) + 1.96 * sd(posterior_preds)), linetype =
  "dashed", alpha = 0.5) +
  +   labs(title = "Posterior Predictive Checks",
  +       x = "ACT Scores", y = "Probability of Success",
  +       color = "Success")
```



Department of Computer Engineering

Output:

Data Exploration and Visualization:





## Department of Computer Engineering

### Logistic Regression Modeling using Bayesian Approach:

Compiling Stan program...  
Start sampling

```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.4e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.34 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration: 1 / 2000 [  0%] (Warmup)
Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 0.046 seconds (warm-up)
Chain 1:           0.045 seconds (Sampling)
Chain 1:           0.091 seconds (Total)
Chain 1:
```

```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 9e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration: 1 / 2000 [  0%] (Warmup)
Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.044 seconds (warm-up)
Chain 2:           0.047 seconds (Sampling)
Chain 2:           0.091 seconds (Total)
Chain 2:
```



Shri Vile Parle Kelavani Mandal's

## DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



### Department of Computer Engineering

```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)
Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.05 seconds (Warm-up)
Chain 3:           0.038 seconds (Sampling)
Chain 3:           0.088 seconds (Total)
Chain 3:
```

```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 7e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.07 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)
Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 0.041 seconds (Warm-up)
Chain 4:           0.049 seconds (Sampling)
Chain 4:           0.09 seconds (Total)
Chain 4:
```



## Department of Computer Engineering

### Model Summary:

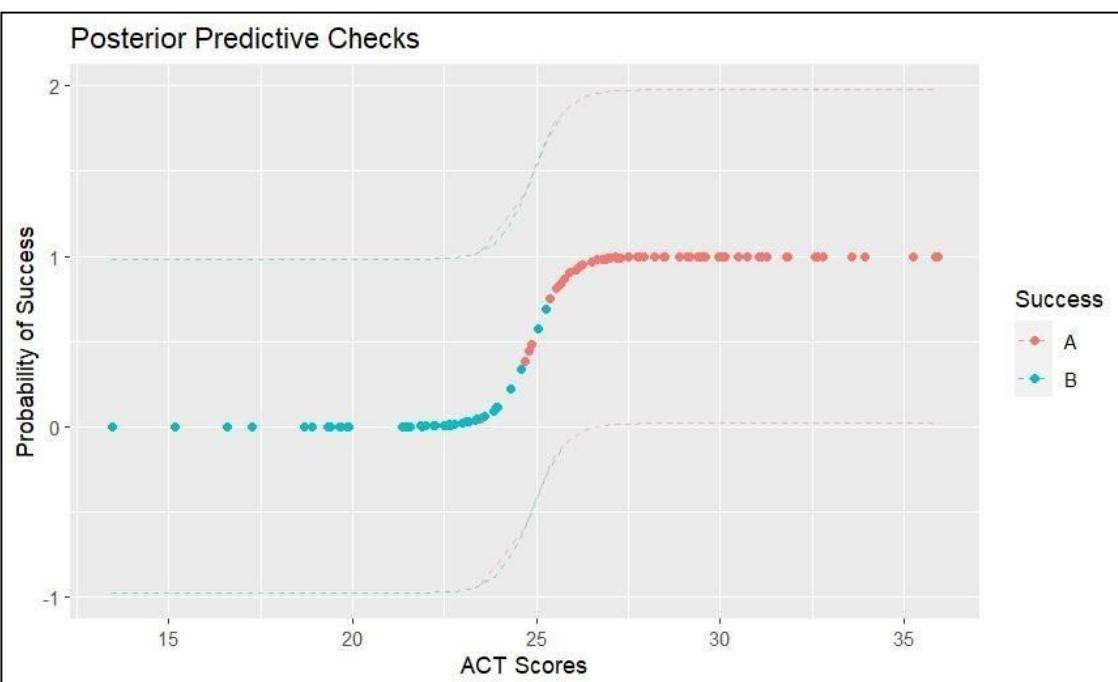
```
Family: bernoulli
Links: mu = logit
Formula: Success_binary ~ ACT_scores
Data: calculus_data (Number of observations: 100)
Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
      total post-warmup draws = 4000
```

#### Population-Level Effects:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-59.83	15.57	-95.02	-34.91	1.00	1497	1834
ACT_scores	2.40	0.62	1.41	3.82	1.00	1480	1741

Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

### Posterior Predictive Checks:



Experiment 6

Shashwat Shah

600004220128

BE Comps C22

Aim : To implement a Beta Binomial Model for over dispersion  
using cancer mortality data

### Theory, Overdispersion

This refers to a situation where the variability in the observed data is greater than what is expected from a given statistical model. In the context of binomial distributions overdispersion occurs when the observed variance of the data exceeds the variance predicted by the assumed distribution.

### Beta Binomial Model,

The Beta-binomial Model is a probabilistic model used to address overdispersion in binomially distributed data. It extends the standard binomial distribution by introducing an associated with a random probability of success drawn from a beta distribution. The beta-binomial distribution allows for a more flexible representation of variability in the data. The probability mass function (PMF)

$$P(y_i | \phi_i) = \frac{m_i c_{y_i}}{\Phi(\alpha, \beta)} B\left(\frac{y_i + \alpha}{n_i}, \frac{n_i - y_i + \beta}{\beta}\right)$$

Conclusion: In this experiment, we implemented a beta  
binomial model for overdispersion using career



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### Experiment No. 6

#### **Aim:**

Implement a Beta-Binomial Model for Over dispersion using Cancer mortality data.

Code:

Importing Libraries:

```
library(LearnBayes)
```

Contour plot of parameters  $\eta$  and K in the beta-binomial model:

```
mycontour(betabinexch0,  
          c(.0001, .003, 1, 20000),  
          cancermortality,  
          xlab="eta", ylab="K")
```

Contour plot of transformed parameters logit( $\eta$ ) and log K in the beta-binomial model:

```
mycontour(betabinexch,  
          c(-8, -4.5, 3, 16.5),  
          cancermortality, xlab="logit  
          eta", ylab="log K")
```

Using 'laplace' for beta-binomial modelling:

```
fit <- laplace(betabinexch,  
                c(-7, 6),  
                cancermortality)
```

```
fit
```

Contour plot of normal approximation of logit( $\eta$ ) and log K in the beta-binomial model



Department of Computer Engineering

```
npar <- list(m=fit$mode, v=fit$var)

mycontour(lbinorm,
           c(-8, -4.5, 3, 16.5),
           npar,      xlab="logit eta",
           ylab="log K")

se <- sqrt(diag(fit$var))

fit$mode - 1.645 * se

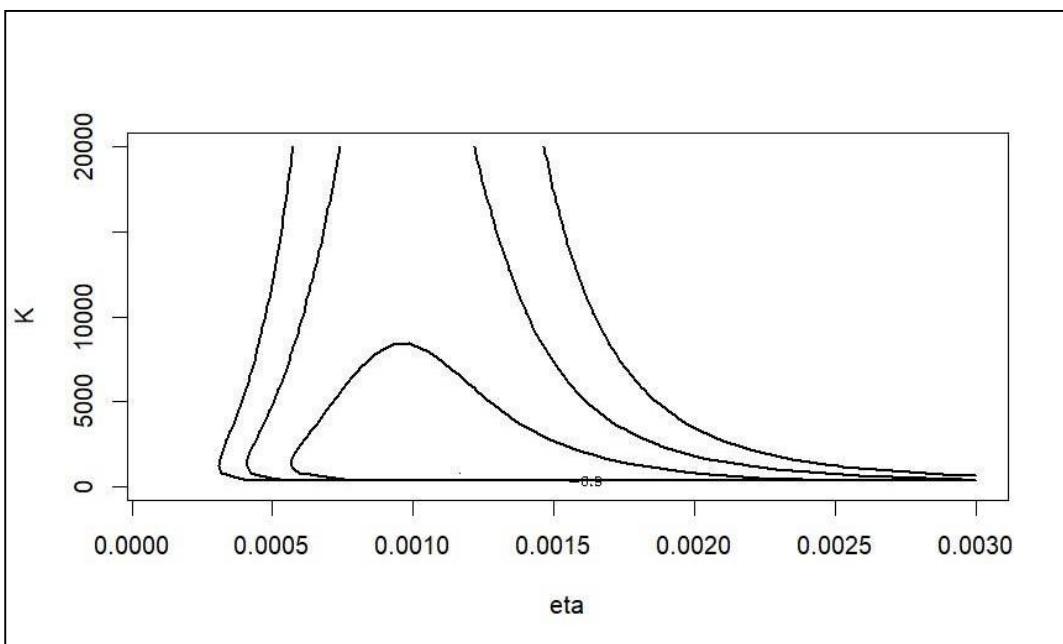
fit$mode + 1.645 * se
```

**Output:**

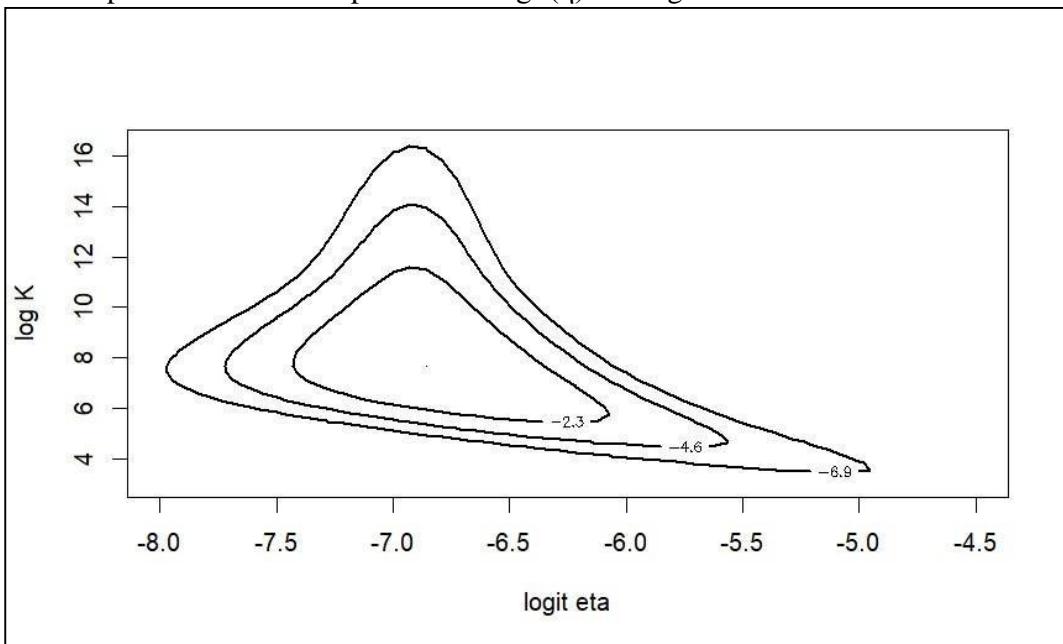
Contour plot of parameters  $\eta$  and  $K$  in the beta-binomial model:



Department of Computer Engineering



Contour plot of transformed parameters  $\text{logit}(\eta)$  and  $\log K$  in the beta-binomial model:



Using 'laplace' for beta-binomial modelling:



Department of Computer Engineering

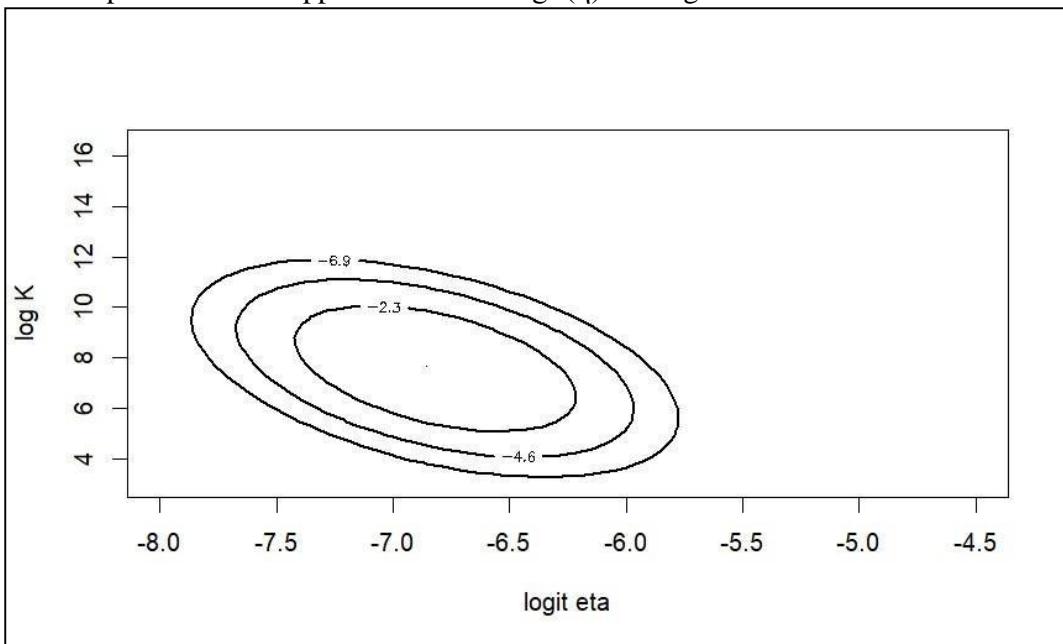
```
$mode
[1] -6.819793 7.576111

$var
[,1]      [,2]
[1,] 0.07896568 -0.1485087
[2,] -0.14850874  1.3483208

$int
[1] -570.7743

$converge
[1] TRUE
```

Contour plot of normal approximation of  $\text{logit}(\eta)$  and  $\log K$  in the beta-binomial model



[1] -7.282052 5.665982

[1] -6.357535 9.486239

## Experiment 7

Shashwat Shah

6000422012C

BE compu C22

Aim: Estimation for the two-parameter exponential distribution using simulated values from the posterior, find the posterior mean and posterior standard deviation

Theory: Sampling Importance Resampling (SIR) algorithm

This algorithm is a Monte Carlo method used for approximating complex distributions by drawing sample from an importance distribution & then resampling to assign higher weights to more relevant samples. The steps are:-

→ Sampling - Generate a set of  $n$  samples ( $x_i$ ) from an importance distribution  $q(x)$  compute the importance distribution & then proportionals to target

$$r(x) = w_i = \frac{p(x)}{q(x)}$$

→ Normalization - Normalize the weights

$$\hat{w}_i = \frac{w_i}{\sum_{i=1}^n w_i}$$

→ Resampling - Draw  $n$  samples with replacement from the set of  $x_i$  with probabilities given by the randomized weights

- 1) Two Parameter Exponential Distribution
- 2) Posterior Distribution
- 3) SIR Simulation
- 4) Posterior Mean and standard deviation

$$\text{Posterior Standard deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Posterior mean})^2}{n}}$$

Conclusion: In this experiment, we successfully implemented the SIR algorithm for estimating the parameters of the exponential distribution.



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### Experiment No. 7

#### **Aim:**

Estimation for the two-parameter exponential distribution: Using your simulated values from the posterior, find the posterior mean and posterior standard deviation.

Code:

Import Libraries:

```
library(MASS) # for multivariate t-distribution
```

Define the parameters for the exponential distribution:

```
lambda <- 1 # rate parameter for the exponential distribution
theta <- 1 # scale parameter for the exponential distribution
```

Define the proposal density function (multivariate t-distribution):

```
proposal_density <- function(x, df, mu, Sigma) {
  dmvt(x, df = df, mu = mu, Sigma = Sigma)
}
```

SIR Step 1: Generate a sample from the proposal density:

```
n <- 1000 # number of samples df <- 2 # degrees of
freedom for the t-distribution mu <- c(lambda, theta) #
mean vector Sigma <- diag(c(1, 1)) # covariance matrix
proposal_sample <- rmvnorm(n, mean = mu, sigma =
Sigma)
```



## Department of Computer Engineering

SIR Step 2: Calculate the weights for each sample:

```
weights <- dexp(proposal_sample[, 1], rate = lambda) * dexp(proposal_sample[, 2], rate =  
1/theta) weights <- weights / sum(weights) # normalize  
the weights
```

SIR Step 3: Perform the resampling step:

```
resampled_indices <- sample(1:n, size = n, replace = TRUE, prob = weights)  
resampled_sample <- proposal_sample[resampled_indices, ]
```

Calculate the posterior mean and standard deviation:

```
posterior_mean <- colMeans(resampled_sample) posterior_sd <-  
apply(resampled_sample, 2, sd) print(paste("Posterior mean  
(lambda): ", posterior_mean[1])) print(paste("Posterior mean (theta):  
", posterior_mean[2])) print(paste("Posterior standard deviation  
(lambda): ", posterior_sd[1])) print(paste("Posterior standard  
deviation (theta): ", posterior_sd[2]))
```

Visualize posterior sample:

```
plot(resampled_sample[, 1], resampled_sample[, 2], xlab = "Lambda", ylab = "Theta",  
main  
= "Posterior sample")
```



### Department of Computer Engineering

#### Output:

Calculate posterior mean and standard deviation:

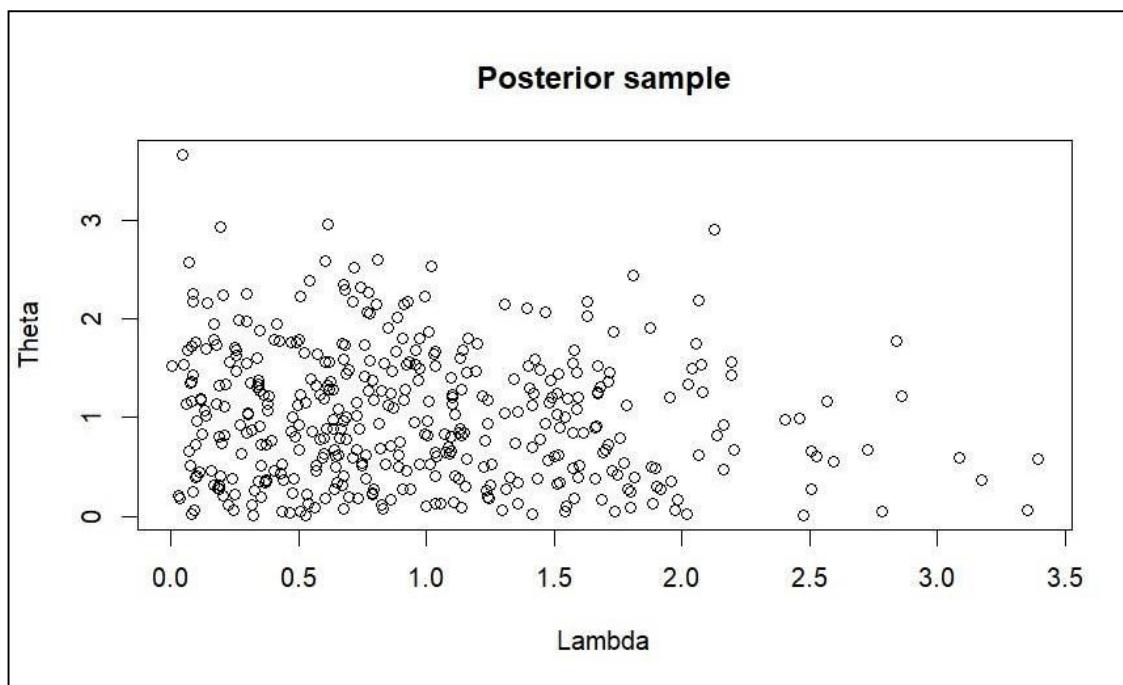
```
[1] "Posterior mean (lambda): 0.743075390295787"
```

```
[1] "Posterior mean (theta): 0.76260856601354"
```

```
[1] "Posterior standard deviation (lambda): 0.587616872814157"
```

```
[1] "Posterior standard deviation (theta): 0.611449920236162"
```

Visualize the Posterior Sample:



Experiment 8

Shashwat Sheh

6000422016

BE Sem 12

Aim: To implement a program for learning about a normal population from grouped data.

Theory: MCMC techniques

The markov chain monte carlo (MCMC) serves as a pivotal technique in Bayesian computing, offering a systematic approach to sample from intricate probability.

This iteration process proves invaluable in Bayesian analysis, particularly when dealing with high dimensional or complex distributions.

Metropolis Random walk algorithm.

The Metropolis random walk algorithm is a cornerstone in Bayesian computing, specifically within the realm of markov chain monte carlo (MCMC) methods.

- 1) Initialization
- 2) Proposal distribution
- 3) Generate a Proposed State
- 4) Acceptance Probability
- 5) Accept or Reject
- 6) Repeat.

Conclusion - In this exercise, we used the Learn Bayes library in to perform Bayesian inference on a normal population from grouped data.



## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### **Experiment No. 8**

#### **Aim:**

Implement a program for learning about a normal population from grouped data, using height and frequency data from student's dataset.

Code:

Import libraries:

Library(LearnBayes)

Observe normally distributed data in grouped form. Consider the posterior of  $(\mu, \log(\sigma))$ :

```
d <- list(int.lo=c(-Inf, seq(66, 74, by=2)),
```

```
int.hi=c(seq(66, 74, by=2), Inf),
```

```
f=c(14, 30, 49, 70, 33, 15))
```

```
y <- c(rep(65,14), rep(67,30), rep(69,49),
```

```
rep(71,70), rep(73,33), rep(75,15)) mean(y)
```

```
log(sd(y))
```

Obtain normal approximation to posterior:

```
start <- c(70, 1) fit <-
```

```
laplace(groupeddatapost, start, d)
```

```
fit
```



### Department of Computer Engineering

Use Metropolis (random walk) MCMC algorithm:

```
modal.sds <- sqrt(diag(fit$var))

proposal <- list(var=fit$var, scale=2)

fit2 <- rwmetrop(groupeddatapost,
                  proposal,
                  start,
                  10000, d)

fit2$accept

post.means <- apply(fit2$par, 2, mean)

post.sds <- apply(fit2$par, 2, sd)

cbind(c(fit$mode), modal.sds)

cbind(post.means, post.sds)

mycontour(groupeddatapost,
           c(69, 71, .6, 1.3), d,
           xlab="mu", ylab="log sigma")
points(fit2$par[5001:10000, 1],
       fit2$par[5001:10000, 2])
```



## Department of Computer Engineering

### Output:

Observe normally distributed data in grouped form. Consider the posterior of  $\mu, \log(\sigma)$ :

```
[1] 70.16588
```

```
[1] 0.9504117
```

Obtain normal approximation to posterior:

```
$mode
[1] 70.169880 0.973644

$var
[,1]      [,2]
[1,] 3.534713e-02 3.520776e-05
[2,] 3.520776e-05 3.146470e-03

$int
[1] -350.6305

$converge
[1] TRUE
```

Use Metropolis (random walk) MCMC algorithm:

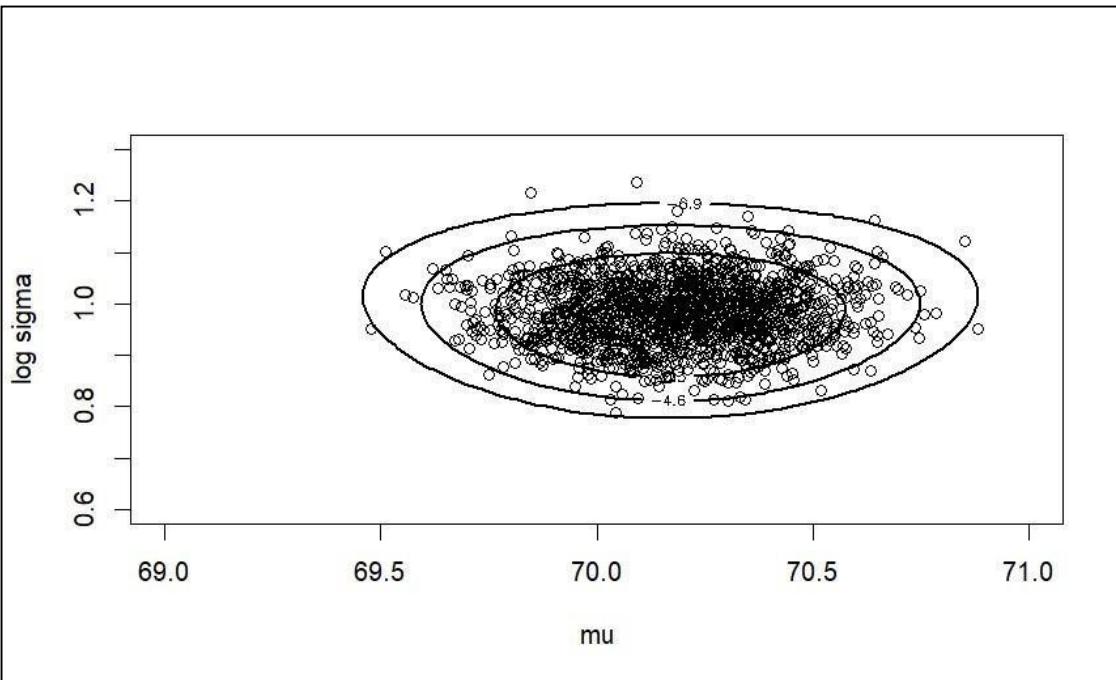
```
[1] 0.293
```

```
modal.sds
[1,] 70.169880 0.18800834
[2,] 0.973644 0.05609341
```

```
post.means post.sds
[1,] 70.1737319 0.19051458
[2,] 0.9815361 0.05686768
```



Department of Computer Engineering





## Department of Computer Engineering

SAP ID.: 60004220126

Name: Shashwat

ShahBatch: HN-3

Subject: Bayesian Computing Laboratory

Semester: VII

### **Experiment No. 9**

#### **Aim:**

Implement a model for Analysis of the Stanford Heart Transplant Data.

Code:

Importing libraries:

Library(LearnBayes)

Using a Pareto model for analyzing Stanford Heart Transplant data:

Laplace fit:

```
start <- c(0, 3, -1) laplacefit <-
laplace(transplantpost,
         start, stanfordheart)
laplacefit
proposal.list(var=laplacefit$var, scale=2)
```



### Department of Computer Engineering

```
tau <- exp(s$par[,1]) plot(density(tau),
main="TAU") lambda <- exp(s$par[,2])
plot(density(lambda),
main="LAMBDA") p <- exp(s$par[,3])
plot(density(p), main="P")

apply(exp(s$par), 2, quantile, c(.05, .5, .95))

par(mfrow=c(1,
1)) t <- seq(1, 240)
p5 <- 0*t p50 <- 0
* t p95 <- 0 * t for
(j in 1:240){
  S <- (lambda / (lambda + t[j])) ^
  p   q <- quantile(S, c(.05, .5, .95))
  p5[j] <- q[1]   p50[j] <- q[2]
  p95[j] <- q[3]
}
```

Estimating a patient's survival curve:

```
plot(t, p50, type="l",
ylim=c(0,1),
```



### Department of Computer Engineering

```
ylab="Prob(Survival)",  
xlab="time")  
lines(t, p5, lty=2)  
lines(t, p95, lty=2)
```



## Department of Computer Engineering

### Output:

Using a Pareto model for analyzing Stanford Heart Transplant data:

Laplace fit:

```
$mode
[1] -0.09210954 3.38385249 -0.72334008

$var
[,1]      [,2]      [,3]
[1,] 0.172788525 -0.009282308 -0.04995160
[2,] -0.009282308  0.214737054  0.09301323
[3,] -0.049951602  0.093013230  0.06891796

$int
[1] -376.2504

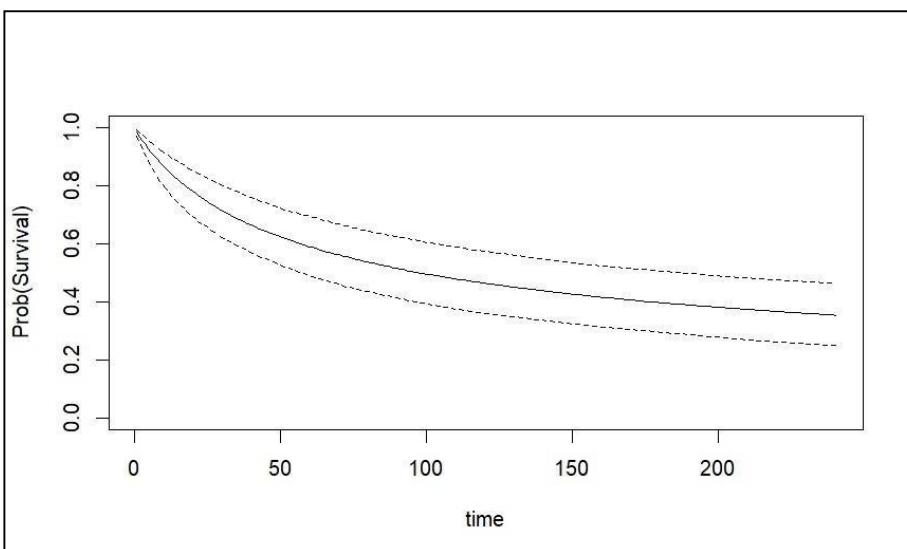
$converge
[1] TRUE
```

Random walk metropolis:

```
[1] 0.193
```

	[,1]	[,2]	[,3]
5%	0.4923842	12.81158	0.3063081
50%	0.9672965	28.82477	0.4648687
95%	2.0249481	63.96780	0.7425854

Estimating a patient's survival curve:



Exponent 9

Shashwat Shah

6000422026

BE Comp. C2

Aim: To implement a model for analysis of the Stanford heart transplant data.

Theory: In Bayesian computing, the pareto model, laplace fit & metropolis, random walk are concept that relate to statistical modeling inference and sampling.

### Pareto Model.

The Pareto distribution also known as 80/20 rule, is often used in Bayesian computing to model phenomena where a small no. of factors contribute to the majority.

It is particularly useful when dealing with sparse or heavy-tailed distributions where conditions model may not be appropriate.

### Metropolis Random Walk.

This is another MCMC used in Bayesian computing for sampling from complex probability distributions. In metropolis, random walk, a chain of samples is generated where each sample is obtained by taking a random move from previous sample and the move is accepted or rejected based on a certain criteria.

Conclusion: In this experiment, we applied the Bayesian Methods to the Stanford heart transplant data using Pareto distribution model.

Experiment 10

Shashwat Shah

60004220126

BE Comp C22

Aim: Study of R to interface with WinBUGS, a stand-alone software program for the windows operating system,

Theory: Introduction to Win BUGS

The BUGS project is focused on the development of software to facilitate Bayesian fitting of complex statistical models using MCMC algorithm. WinBUGS is a program for sampling from a general posterior distribution of a Bayesian model - by use of Gibbs sampling.

The Bayesian model for this task can be developed using the following WinBUGS model script

model {

$y \sim \text{dbin}(p, n)$

$p \sim \text{dtbeta}(\alpha, \beta)$

}

An R interface to WinBUGS - R/WINBUGS is an R interface to WinBUGS which allows the use of WinBUGS by use of R interface. There are 4 necessary inputs that are similar to the inputs requested within the WinBUGS program.

① Model

③ Parameters

② Data

④ Initial Values.

Conclusion - In this study, we examined the integration of R with WinBUGS, a widely used software for the Bayesian computing using Markov Chain Markov model, (MCMC) methods. In conclusion the study found that employing R as a, interface to WinBUGS offers a convenient and adaptable approach to perform analysis using the MCMC methods.