Bayesian Learning Computer Lab 1

bisku859 and gowku593

4/18/2021

Assignment 1. Daniel Bernoulli

Let $y_1, ..., y_n \mid \theta \sim Bern(\theta)$, and assume that you have obtained a sample with s = 8 successes in n = 24 trials.

Assume a $Beta(\alpha_0, \beta_0)$ prior for θ and let $\alpha_0 = \beta_0 = 3$

(a)

Draw random numbers from the posterior $\theta \mid y \sim Beta(\alpha_0 + s, \beta_0 + f)$, where $y = y_1, ..., y_n$ and verify graphically that the posterior mean and standard deviation converges to the true values as the number of random draws grows large.

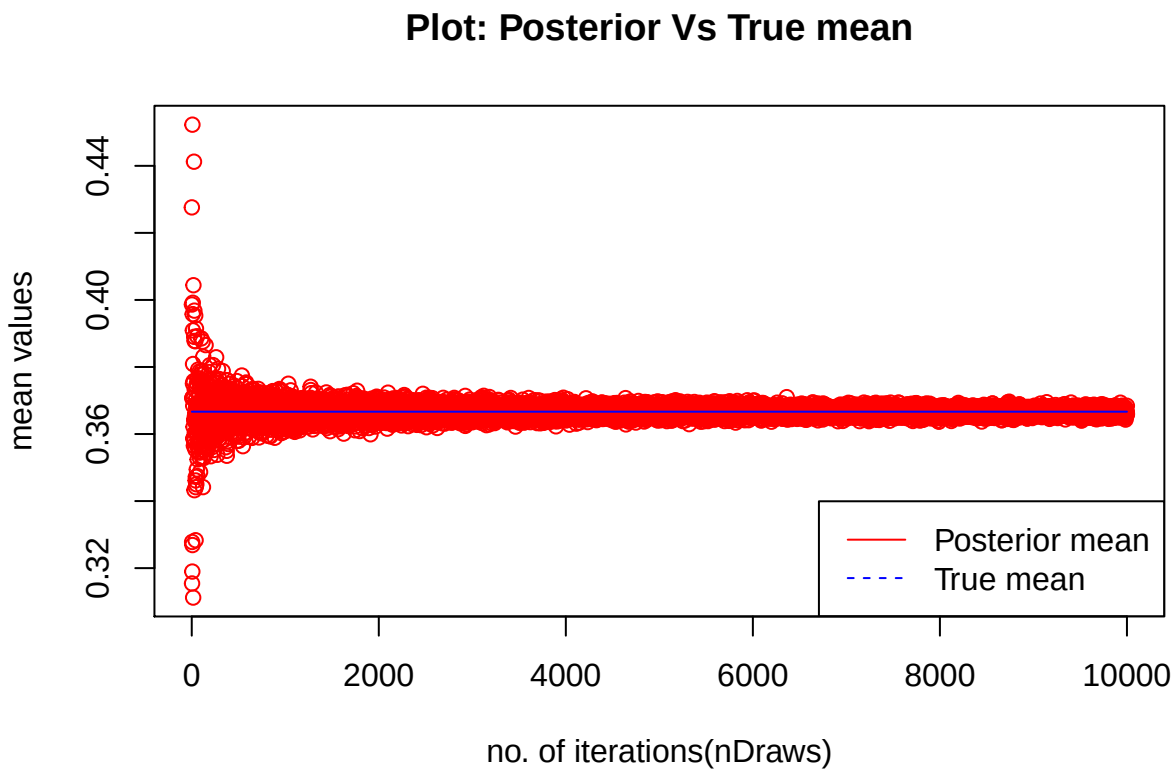
Solution :

Important Formulaes that have been used for Beta distribution:

Expected Value or Mean = $\frac{\alpha}{\alpha+\beta}$

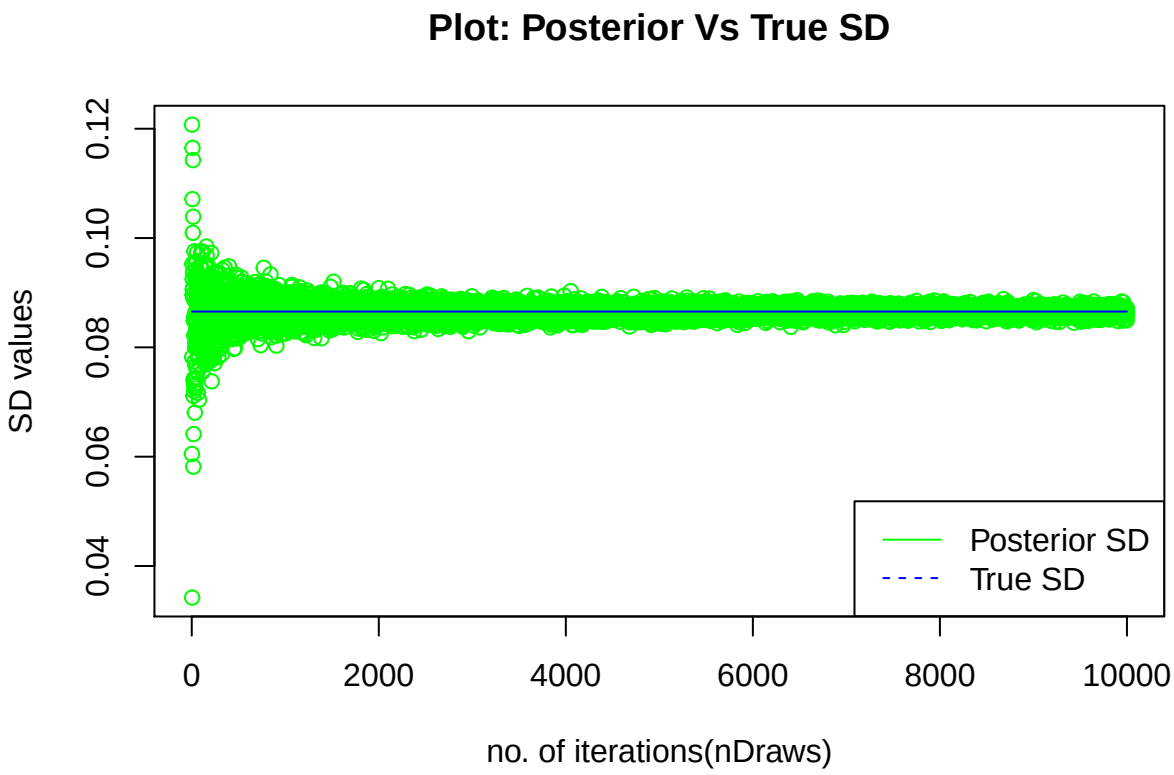
Standard Deviation = $\sqrt{(}(\alpha * \beta)/((\alpha + \beta)^2 * (\alpha + \beta + 1))$

Posterior mean Vs True mean



Looking at the above graph, we can say that the computed values(Posterior) of mean is converging to the true value (highlighted in blue) as the number of random Draws grows large.

Posterior SD Vs True SD



Looking at the above graph, we can say that the computed values(Posterior) of SD is converging to the true value (highlighted in blue) as the number of random Draws grows large.

(b)

Use simulation (nDraws = 10000) to compute the posterior probability $Pr(\theta > 0.4 \mid y)$ and compare with the exact value

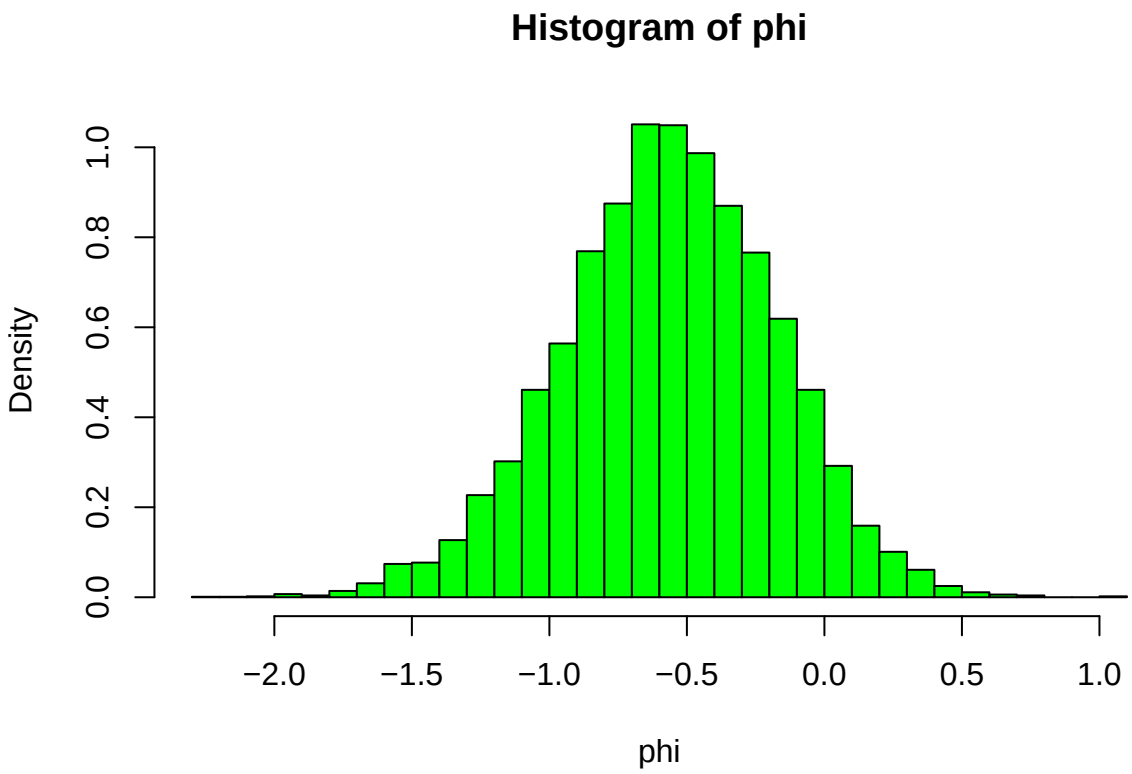
```
## The simulated probability is  0.3434
```

```
## The theoretical posterior probability is  0.3426654
```

The simulated probability is ver near to the theoritical posterior probability where theta values are greater than 0.4 given samples. This conveys that the estimations are meaningful.

(c)

Compute the posterior distribution of the log-odds $\phi = log\frac{\theta}{1-\theta}$ by simulation (nDraws = 10000).




```
##
## Call:
##  density.default(x = phi)
##
## Data: phi (10000 obs.);  Bandwidth 'bw' = 0.0553
##
##        x                 y
##  Min.   :-2.4184   Min.   :0.0000084
##  1st Qu.:-1.5078   1st Qu.:0.0044522
##  Median :-0.5971   Median :0.0741228
##  Mean   :-0.5971   Mean   :0.2742681
##  3rd Qu.: 0.3135   3rd Qu.:0.5159932
##  Max.   : 1.2241   Max.   :1.0604735
```

2. Log-normal distribution and the Gini coefficient.

(a)

Log Normal Distribution and Gini-Coeff :

The Density funtion for log normal distribution is given by :

$p(y \mid \mu, \sigma^2) = \frac{1}{y \times \sqrt{2\pi\sigma^2}} exp\left[-\frac{1}{2\sigma^2}(\log y - \mu)^2\right]$

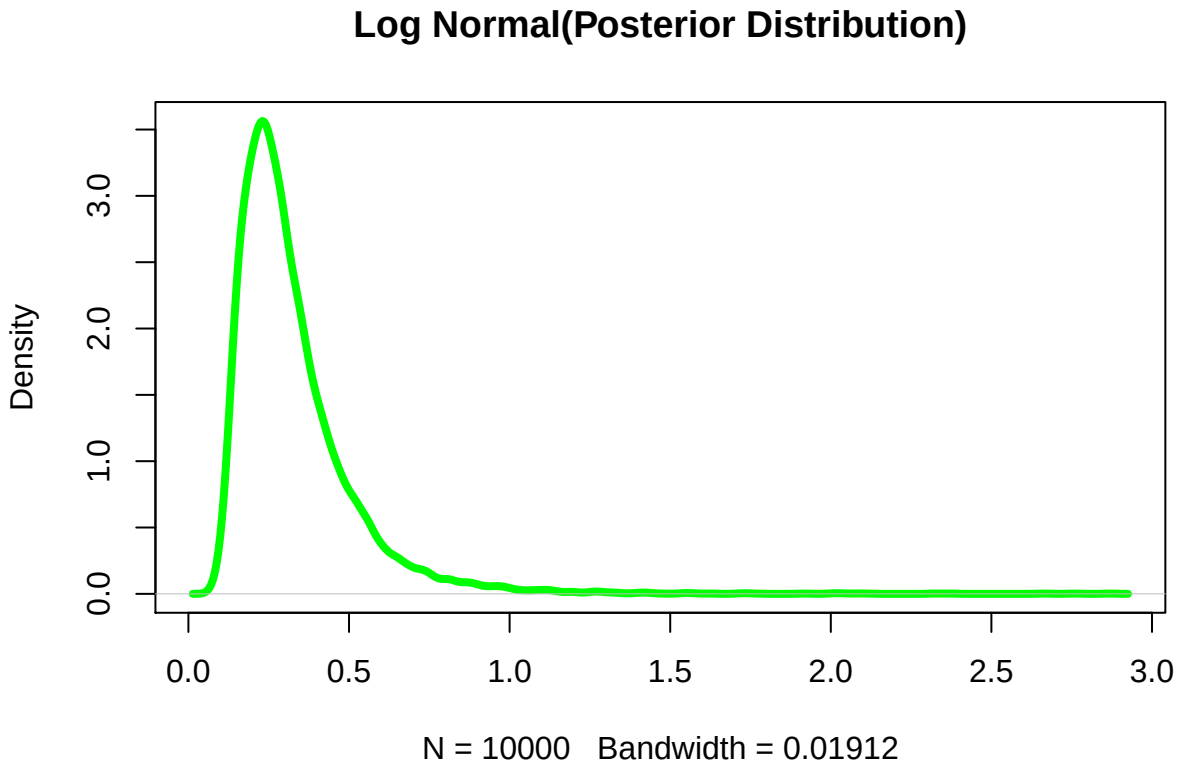
The posterior for σ^2 is the $Inv - \chi^2(n, \tau^2)$ distribution , where $\tau^2 = \frac{\sum_{i=1}^{n}(\log y_i - \mu)^2}{n}$

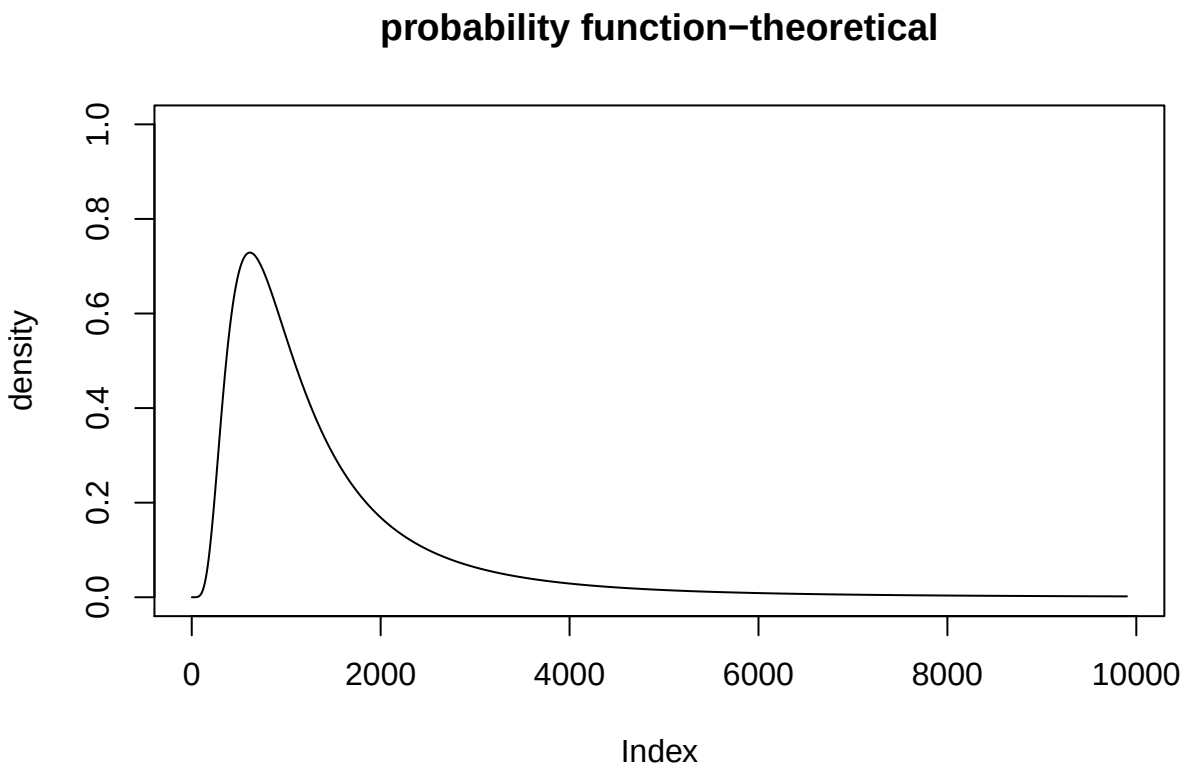
(a) Simulate 10000 draws from the posterior of σ^2 (assuming $\mu = 3.8$) and compare it with the theoretical $Inv - \chi^2(n, \tau^2)$ posterior distribution

Steps: 1. Draw $X \sim \chi^2(n)$ 2. Compute $\sigma^2 = \frac{n \times \tau^2}{X}$ (this is a draw from $)Inv - \chi^2(n, \tau^2)$ 3. Repeated until nDraws (10000) times

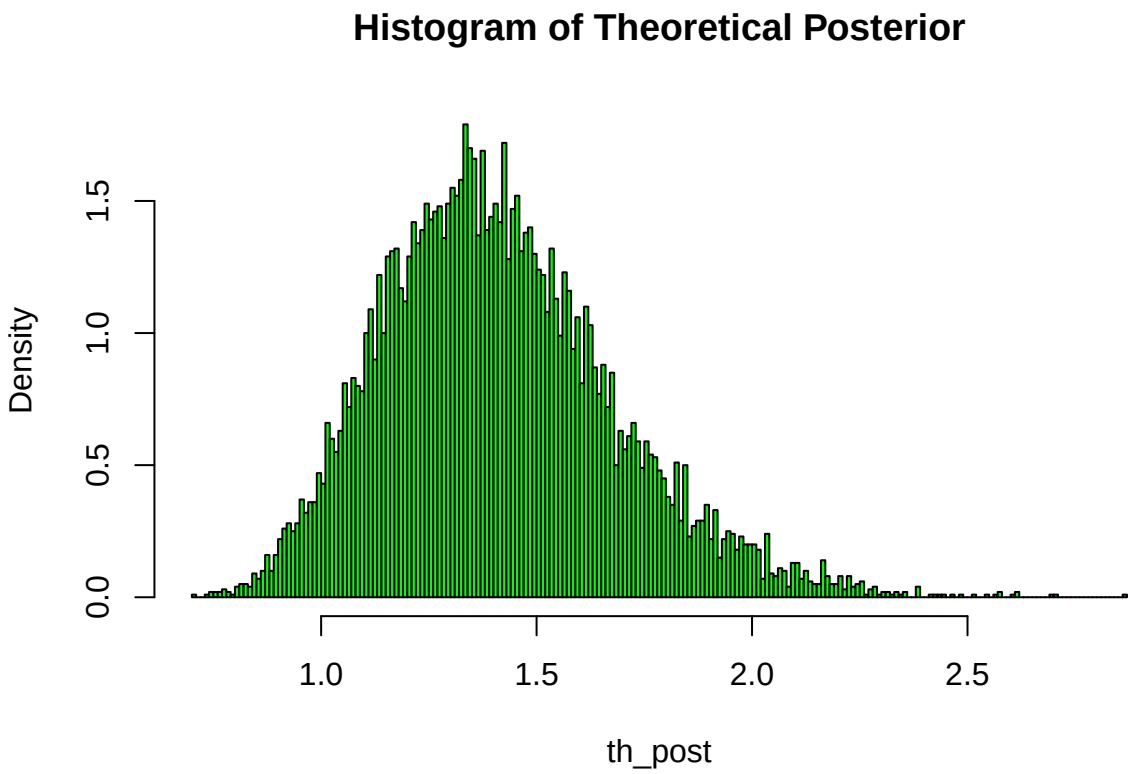
From the Scaled Inverse Chi Square distribution, mean and Sd can be calculated as

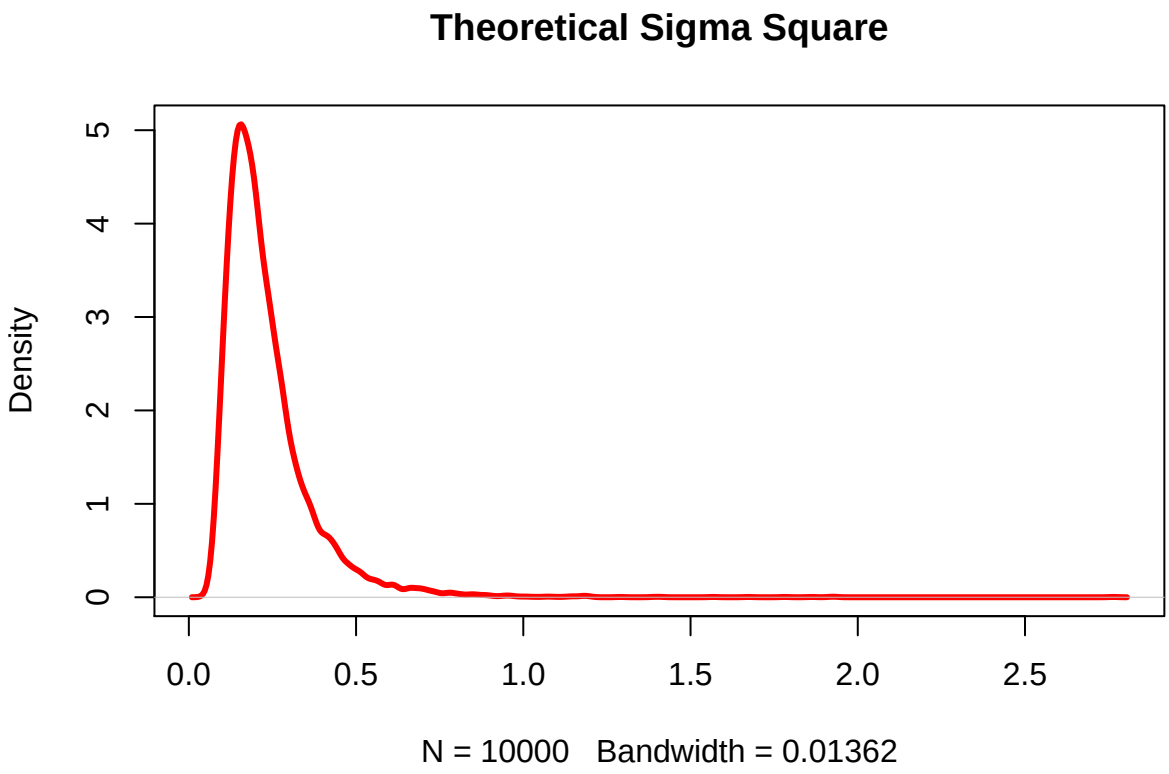
$mean = \frac{n}{n-2} \times \tau^2$ for n>2 $sd = \sqrt{\frac{2n^2}{(n-2)^2(n-4)} \times \tau^4}$ for n>4

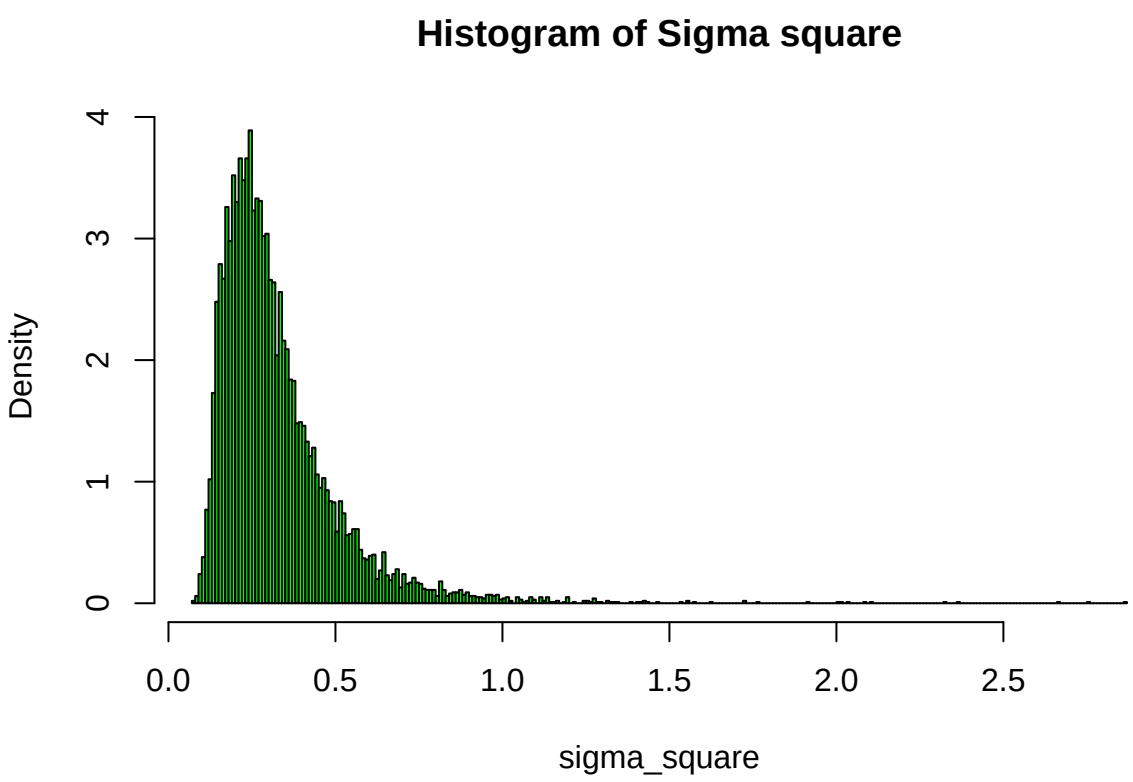


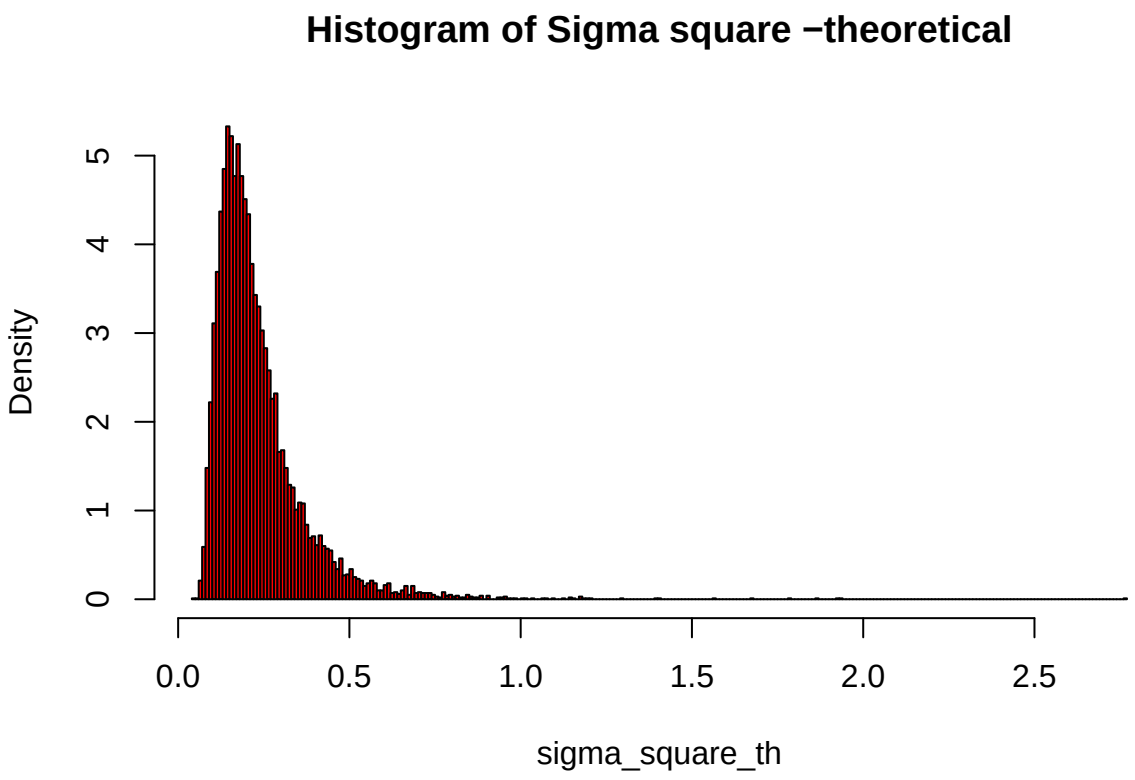



```
##  The computed value of mean 0.3263818

##  The computed value of sd 0.185362

##  The theoretical value of mean 0.3263046

##  The theoretical value of sd 0.188392
```


Histogram of Theoretical Posterior
Density
0.0
0.5
1.0
1.5
1.0
1.5
2.0
2.5
th_post


Theoretical Sigma Square
Density
0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
N = 10000 Bandwidth = 0.01362


Histogram of Sigma square
Density
0
1
2
3
4
0.0
0.5
1.0
1.5
2.0
2.5
sigma_square



We can see that the theoritical values of mean and sd is very much near to the computed values of mean and sd respectively for inverse chi square distribution for given mu and number of iterations.

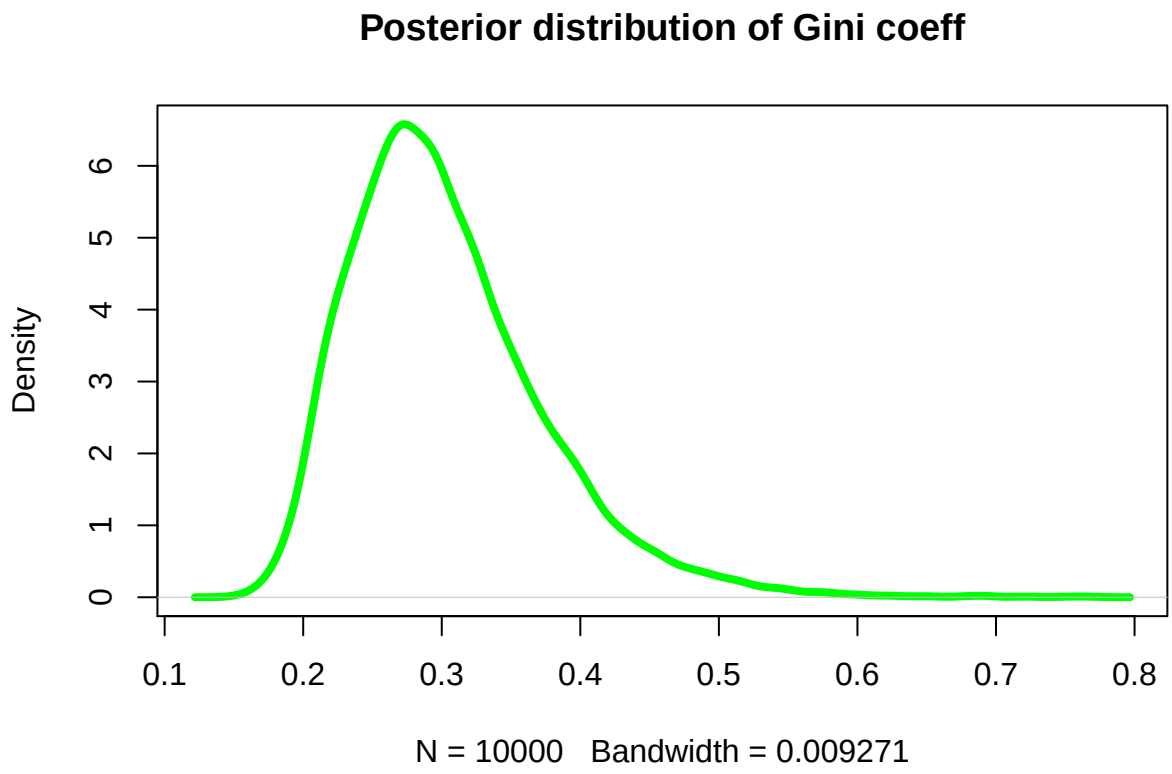
The plotted graphs shows that the theoritical and computed values are similar and hence we can say that the computed values seems correct.

(b)

$$G = 2\phi(\frac{\sigma}{\sqrt{2}}) - 1$$

ϕ is the cummulative distribution function for the standard normal distribution

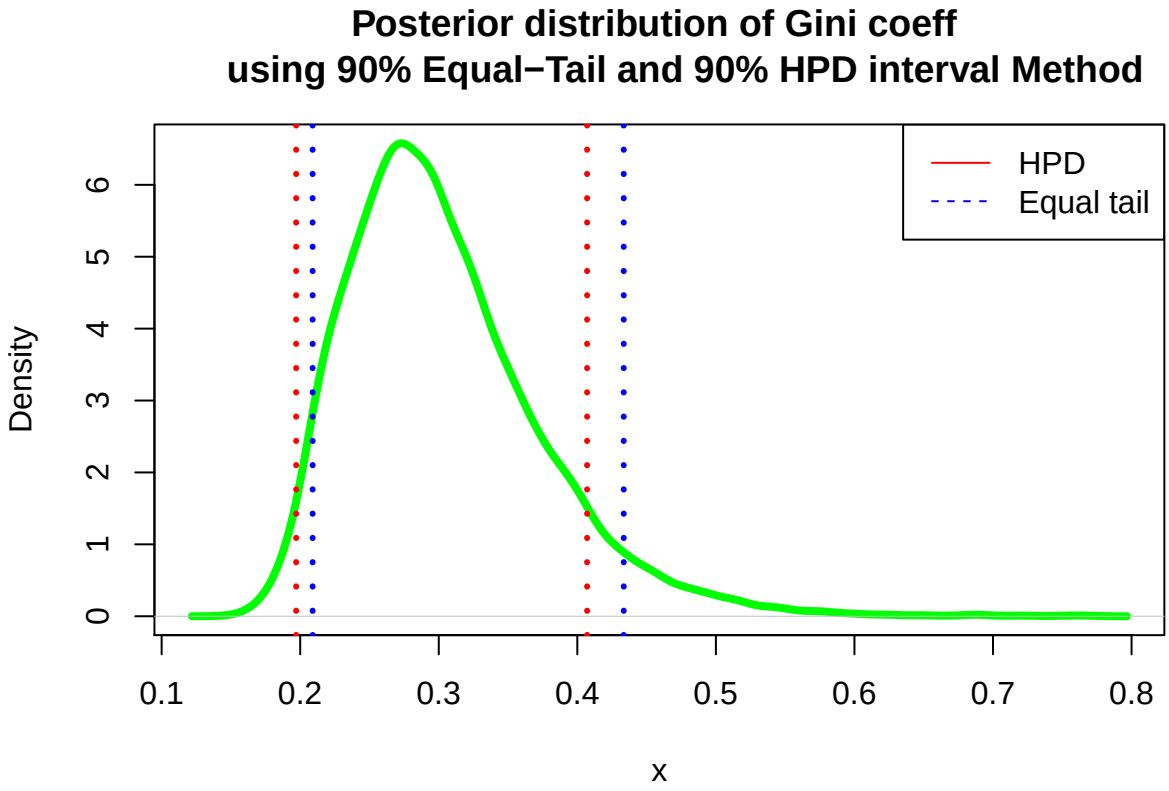
Herein, We can use pnorm() for calculating CDF where q is the values of $\frac{\sigma}{\sqrt{2}}$



(c)

```
## [1] 0.1970842
```

```
## [1] 0.407076
```



HPD (Highest Probability Density) is basically the interval band which captures the top most density of the distribution. This eventually captures values of all possible highest value including the mode. In the above graph the distribution is left skewed and band in red captures top 90% (density area) of the values, normal to x.

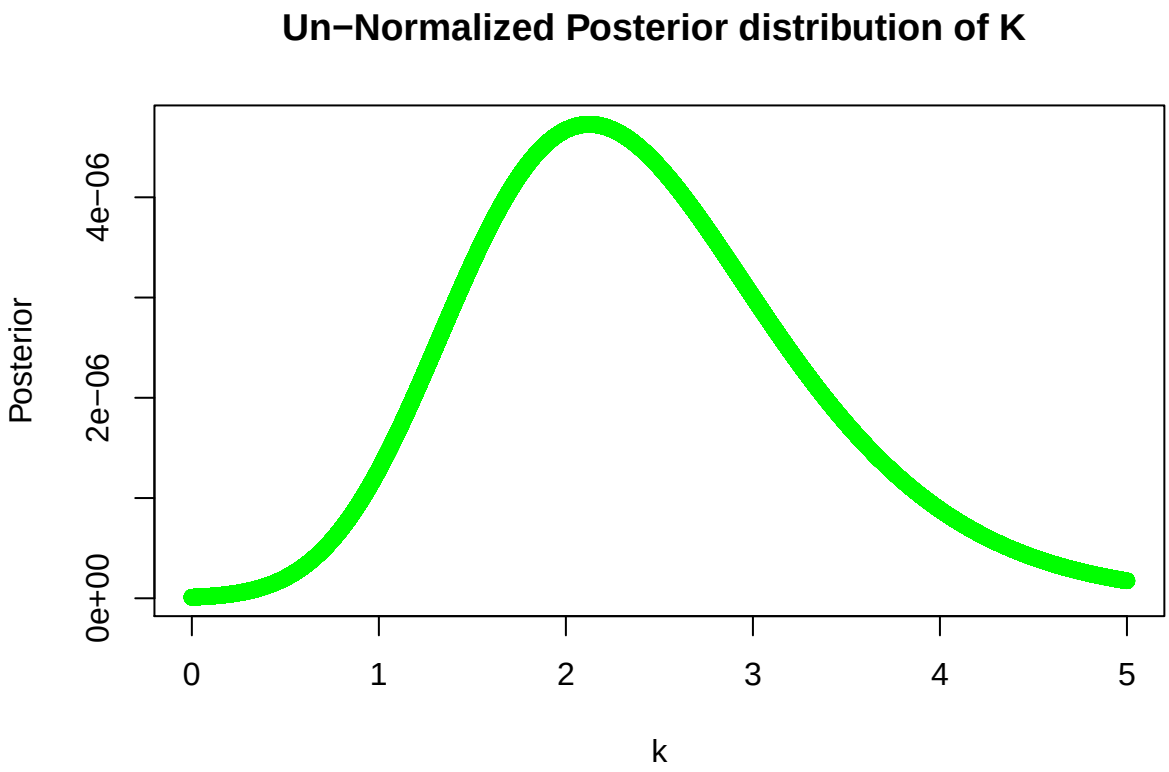
The equal-tail interval basically clips the equal value (density area) towards both side of tails . In the above graph , it clips 5 % value on both left and right side of tail to give us the band in blue color.

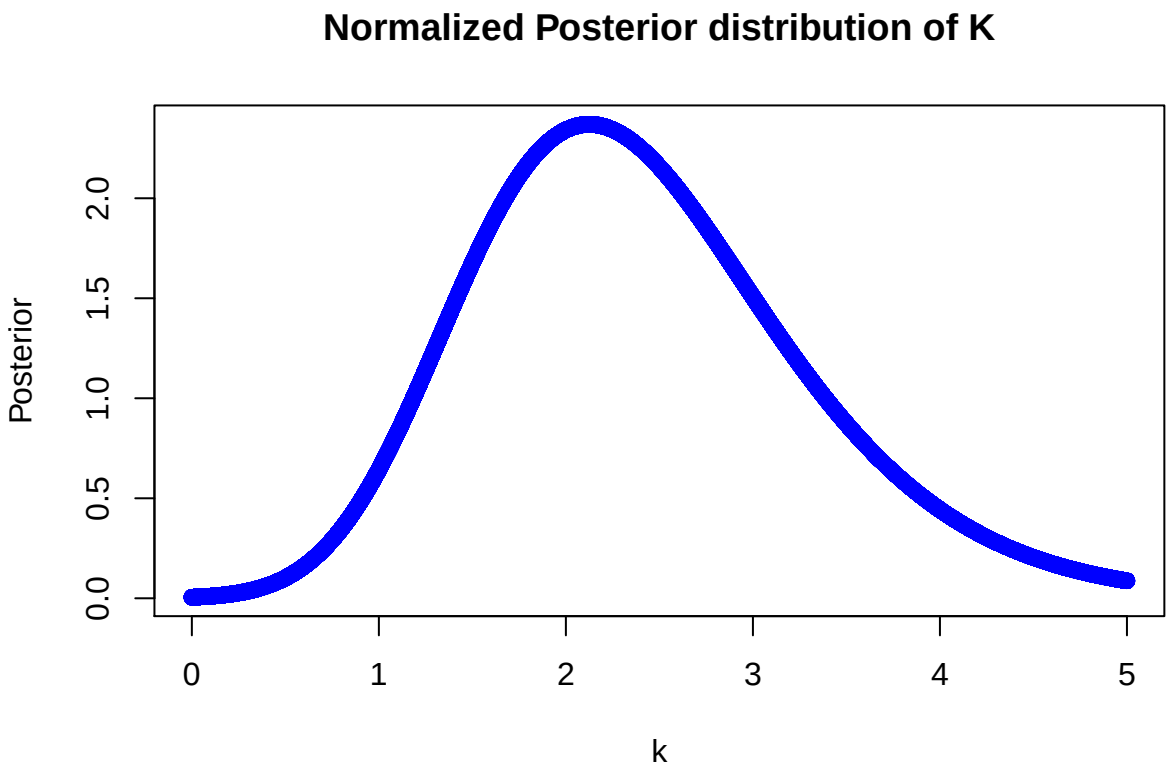
3. Bayesian inference for the concentration parameter in the von Mises distribution.

This exercise is concerned with directional data. The point is to show you that the posterior distribution for somewhat weird models can be obtained by plotting it over a grid of values. The data points are observed wind directions at a given location on ten different days. The data are recorded in degrees,where North is located at zero degrees (see Figure 1 on the next page, where the angles are measured clockwise). To fit with Wikipedias description of probability distributions for circular data we convert the data into radians $-\pi \leq y \leq \pi$. The 10 observations in radians are (-2.44, 2.14, 2.54, 1.83, 2.02, 2.33,-2.79, 2.23, 2.07, 2.02).

Assume that these data points are independent observations following the von Mises distribution $p(y \mid \mu, k) = \frac{exp[k.cos(y-\mu)]}{2\pi I_0(k)}$ for $-\pi \leq y \leq \pi$

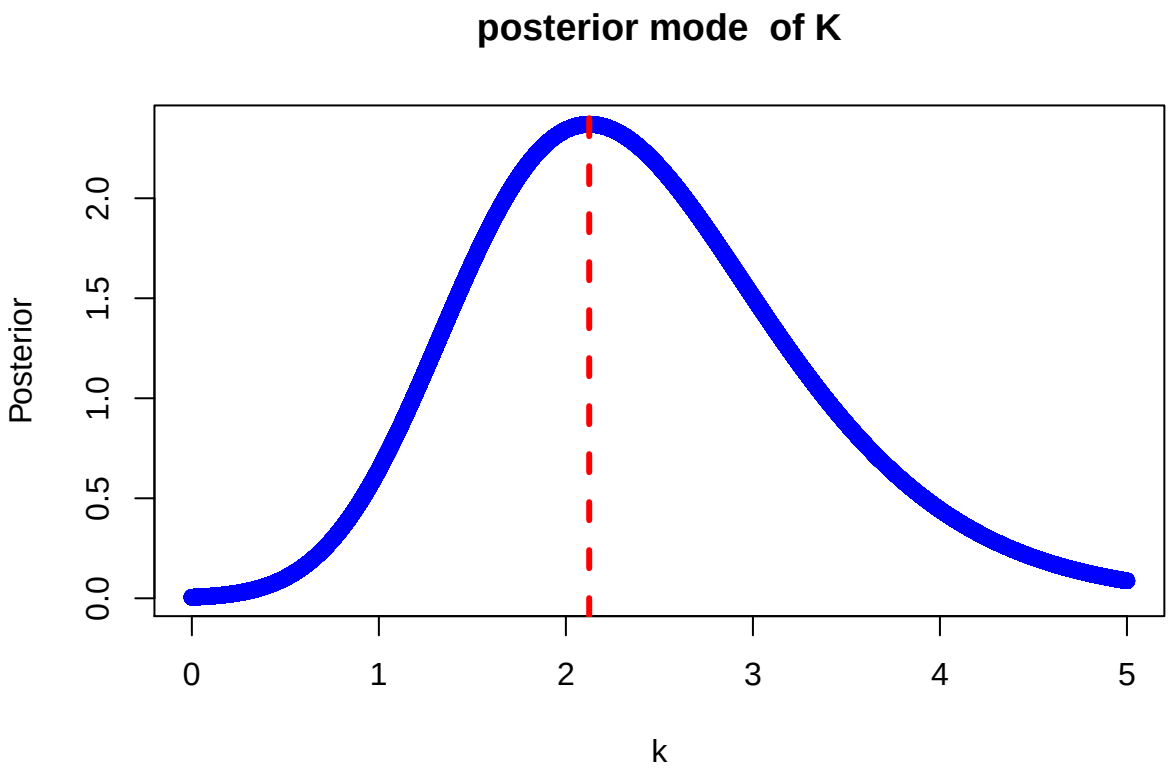
(a) Plot the posterior distribution of k for the wind direction data over a fine grid of k values.


Un−Normalized Posterior distribution of K
Posterior
0e+00
2e−06
4e−06
0
1
2
3
4
5
k



b. Find the (approximate) posterior mode of k from the information in a).

Posterior mode is the maximum posterior probability (MAP) estimate, that equals the mode of the posterior distribution.




```
## The value for posterior mode of k is : 2.12475
```

Contribution :

Biswas contributed majorly with Assignment # 3, report writing and overall trouble-shooting. Gowtham contributed majorly with Assignment # 1 and 2. Both team members discussed on solution approach and expected outcomes of all the assignments.

Note : We have referred lecture notes, our group's previous submission, R -documentation and Wikipedia

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(Hmisc)
library("tidyverse")
library(dplyr)
library(LaplacesDemon)

set.seed(12345) # have taken set seed to reproduce the result
#Given that:
```

```r
n<-24
s<-8
#Calculated f below
f<-n-s
alpha_0 <-3
beta_0<-3
nDraws=10000
alp<-alpha_0+s  # 3+8
bet<-beta_0+f   #3+16


#Given
#Beta(11,19) , density function for posterior

#rbeta(nDraws,shape1 = alpha_0, shape2 = beta_0,ncp = 0)

mean_fun<-function(malpha,mbeta){
  #Expected value or mean
  res<-(malpha/(malpha+mbeta))
  return(res)
}
true_mean<-mean_fun(malpha=alp,mbeta=bet)

sd_fun<-function(salpha,sbeta){
  res<-sqrt((salpha*sbeta)/(((salpha+sbeta)^2)*(salpha+sbeta+1)))
}

true_sd<-sd_fun(salpha=alp,sbeta=bet)

random_draw<-function(N,kalpha,kbeta){
  #using rbeta to generate random numbers from beta density
   ran<-rbeta(N,kalpha,kbeta)
   # need mean & std deviation as output
   res<-list(S.No=N,mean=mean(ran),std_dev=sd(ran))
   return(res)
  }
#random_draw(10000,11,19)

output<-data.frame()
for(i in 1:nDraws){
  out<-random_draw(i,alp,bet)
  output<-rbind(output,out)
}
table<-cbind(output,true_mean,true_sd)
plot(x=1:nDraws,y=table$mean,col="red",ylab="mean values",
     xlab = "no. of iterations(nDraws)", main="Plot: Posterior Vs True mean")
lines(table$true_mean,col="blue",type="l")
legend("bottomright", c("Posterior mean", "True mean"),col = c("red", "blue"), lty = c(1, 2))

plot(x=1:nDraws,y=table$std_dev,col="green",
     ylab="SD values",xlab = "no. of iterations(nDraws)",
     main="Plot: Posterior Vs True SD")
lines(table$true_sd,col="blue",type="l")
```

```r
legend("bottomright", c("Posterior SD", "True SD"),col = c("green", "blue"), lty = c(1, 2))

set.seed(123) # have taken set seed to reproduce the result
#generating random draws
random_draw2<- function(N,kalpha,kbeta){
 ran<-rbeta(N,kalpha,kbeta)
 return(ran)
}
nDraws=10000
theta_values<-random_draw2(N=nDraws,kalpha=alp,kbeta=bet)

#Simulated value
posterior_prob<- sum(theta_values>0.4)/nDraws  # simulated
# Probability of theta > 0.4
exact_prob <-pbeta(q=0.4,shape1 =alp,shape2 = bet) # exact value
cat("The simulated probability is ",posterior_prob,"\n")
cat("The theoretical posterior probability is ",1-exact_prob)
#percentage cal

phi<-log(theta_values/(1-theta_values))
hist(phi,col="green",breaks=30,probability = TRUE)
density(phi)

observation<- c(38,20,49,58,31,70,18,56,25,78) # given
n<-length(observation)
mean_obs<-sum(observation)/n

mu<-3.8 # given
nDraws=10000 #given

tau_square<- sum((log(observation)- mu)^2)/(n) # formula
sigma_square<-NULL
#Calculation as per formula
i=1
while (i<=nDraws) {
  X=rchisq(n=1,df=n)
  sigma_square[i]<-(n)*tau_square/X
  i=i+1
}

plot(density(sigma_square),col="green",
     main = "Log Normal(Posterior Distribution)",lwd = 4)
#plot(tau_square,col="red",main = "Log Normal(Theoritical Distribution)",lwd = 15,type="l")
sek <- seq(0.1,10,0.001)
#length(sek)
plot(dinvchisq(sek,5,1), ylim = c(0,1), type = "l", main = "probability function-theoretical",
     ylab = "density")


mean_sigma<-mean(sigma_square)
sd_sigma<-sd(sigma_square)
cat(" The computed value of mean",mean_sigma,"\n")
cat(" The computed value of sd",sd_sigma,"\n")
```

```r
# Theoretical Values
n<-length(observation)
th_mean<- n*tau_square/(n-2)
cat(" The theoretical value of mean",th_mean,"\n")
num<-2*(n^2)*(tau_square^2)
dem<-(n-2)^2*(n-4)
th_sd<-sqrt(num/dem)
cat(" The theoretical value of sd",th_sd,"\n")


sigma_square_th<-c()
th_post<-c()
i=1
while(i<=nDraws){
  X=rchisq(n=1,df=n)
  sigma_square_th[i]<-(n)*(th_sd)/X
  th_post[i]<-rlnorm(1,sdlog =sqrt(sigma_square/n),meanlog=th_mean)
  i=i+1
}
hist(th_post,breaks = 200,probability = TRUE,
     main = "Histogram of Theoretical Posterior",col="green")
# hist(sigma_square_th,probability = TRUE)
plot(density(sigma_square_th),col="red",main="Theoretical Sigma Square",lwd="3")
hist(sigma_square,breaks = 200,probability = TRUE,
     main = "Histogram of Sigma square",col="green")
hist(sigma_square_th,breaks = 200,probability = TRUE,
     main = "Histogram of Sigma square -theoretical",col="red")


#As per given formula
gini<-2*pnorm(q=sqrt(sigma_square)/sqrt(2))-1
plot(density(gini),main="Posterior distribution of Gini coeff",col="green",lwd=4)
# using stated density function
G_density<-density(gini)
table<-data.frame(x=G_density$x,y=G_density$y)
sum_y<-sum(table$y)
table_y_decend<-table %>% arrange(desc(y))

#Normalizing
store_y<-cumsum(table_y_decend$y)/(sum(table_y_decend$y))

index_0.9<-which(store_y>0.9)[1] # last value
value_0.9<-table_y_decend$y[index_0.9]

values_gr_0.9<-which(table_y_decend$y>value_0.9)
values_x<-table_y_decend$x[values_gr_0.9]

min(values_x)
max(values_x)


# two tail
store_y_unsorted<-cumsum(table$y)/(sum(table$y))
```

```r
index_0.95_u<-which(store_y_unsorted>0.95)[1] # last value
value_0.95_u<-table$x[index_0.95_u]


index_0.05_u<-which(store_y_unsorted>=0.05)[1]
value_0.05_u<-table$x[index_0.05_u]

plot(density(gini),main="Posterior distribution of Gini coeff
     using 90% Equal-Tail and 90% HPD interval Method",col="green",lwd=4,xlab = "x")
abline(v=min(values_x), col="red", lwd=3,lty=3)
abline(v=max(values_x), col="red", lwd=3,lty=3)
abline(v=value_0.95_u, col="blue", lwd=3,lty=3)
abline(v=value_0.05_u, col="blue", lwd=3,lty=3)
legend("topright", c("HPD", "Equal tail"),col = c("red", "blue"), lty = c(1, 2))




observ<- c(-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02)
n<-length(observ)
mu<-2.39 # given
#let k be nDraws (number of iterations)
k<-seq(0,5,0.00005)
my_fun<- function(k,observ){
  p<-exp(k*cos(observ-mu))/(2*pi*besselI(k,0)) # k>0
  return(p)
}

probab<-function(k){
  prob<-sapply(observ,my_fun,k=k)
  posterior<- prod(prob)*dexp(k)
  return(posterior)
}
post<-c()
for (i in 1:length(k)){
  post[i]<-probab(k[i])
}

res<-list(k=k,posterior=post)

plot(res$k,res$posterior,col="green",
     main="Un-Normalized Posterior distribution of K ",
     xlab = "k",ylab="Posterior")

#normalized constant
norm_const<-sum((1/length(res$posterior))*res$posterior)

plot(x=k,y=res$posterior/norm_const,xlab = "k",ylab = "Posterior",col="blue",
     main="Normalized Posterior distribution of K")
```

```r
max_index<- which.max(res$posterior)
posterior_mode<-res$posterior[max_index]
plot(x=k,y=res$posterior/norm_const,xlab = "k",ylab = "Posterior",col="blue",
     main="posterior mode  of K")
abline(v=k[max_index], col="red", lwd=3, lty=2)
cat("The value for posterior mode of k is :",k[max_index])
```