Bayesian Learning Lab1

Umamaheswarababu Maddela (umama339) and Dinesh Sundaramoorthy(dinsu875)

2023-04-05

1. Daniel Bernoulli

1(a)

Given data

$y_1, \ldots y_n | \theta \sim Bern(\theta)$
$n = 70 \ trials$
$s = 22$
$f = n - s = 48$
$Prior \ \ p(\theta) \sim Beta(\alpha_0, \beta_0), \ \ \alpha_0 = \beta_0 = 8$
$Posterior \ \ p(\theta|y) \sim Beta(\alpha_0 + s, \beta_0 + f)$
$nDraws = 10000$

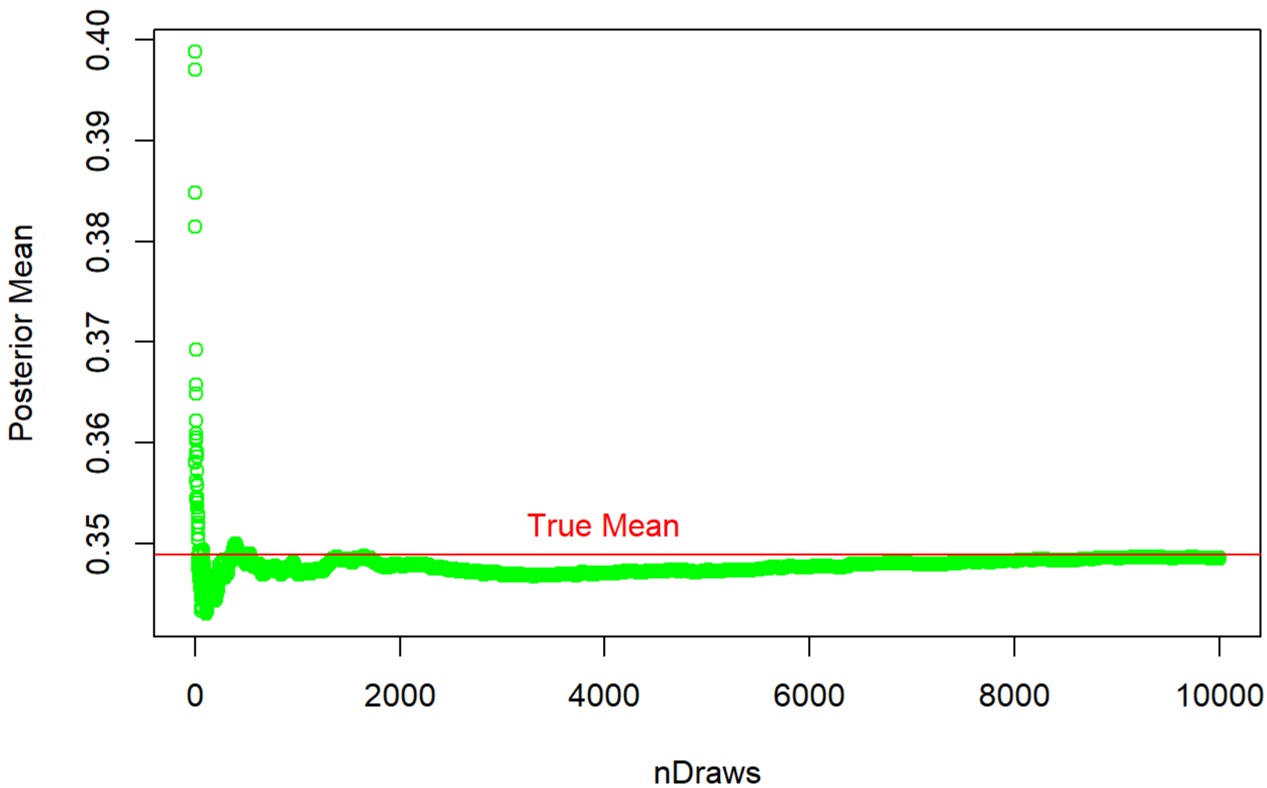
True posterior mean is given by $E[\theta] = \frac{\alpha_0+s}{\alpha_0+s+\beta_0+f}$

True posterior standard deviation is given by $SD[\theta] = \sqrt{\frac{(\alpha_0+s)(\beta_0+f)}{(\alpha_0+s+\beta_0+f+1)(\alpha_0+s+\beta_0+f)^2}}$

```
#1(a)
set.seed(1234567)
n <- 70
s <-  22
f <-  n-s
alpha_zero = 8
beta_zero = 8
nDraws=10000

#drawing 10000 random samples from posterior distribution
random_values <- rbeta(nDraws,alpha_zero+s,beta_zero+f)
means<-c()
sd_s<-c()
for (i in 1:nDraws){
  means<-c(means,mean(random_values[1:i]))
  sd_s <- c(sd_s, sd(random_values[1:i]))
}

#calculating true mean and true standard deviation
true_mean = (alpha_zero+s)/(alpha_zero+s+beta_zero+f)
true_sd = sqrt((alpha_zero+s)*(beta_zero+f)/((alpha_zero+s+beta_zero+f+1)*(alpha_zero+s+beta_zer
o+f)**2))
```

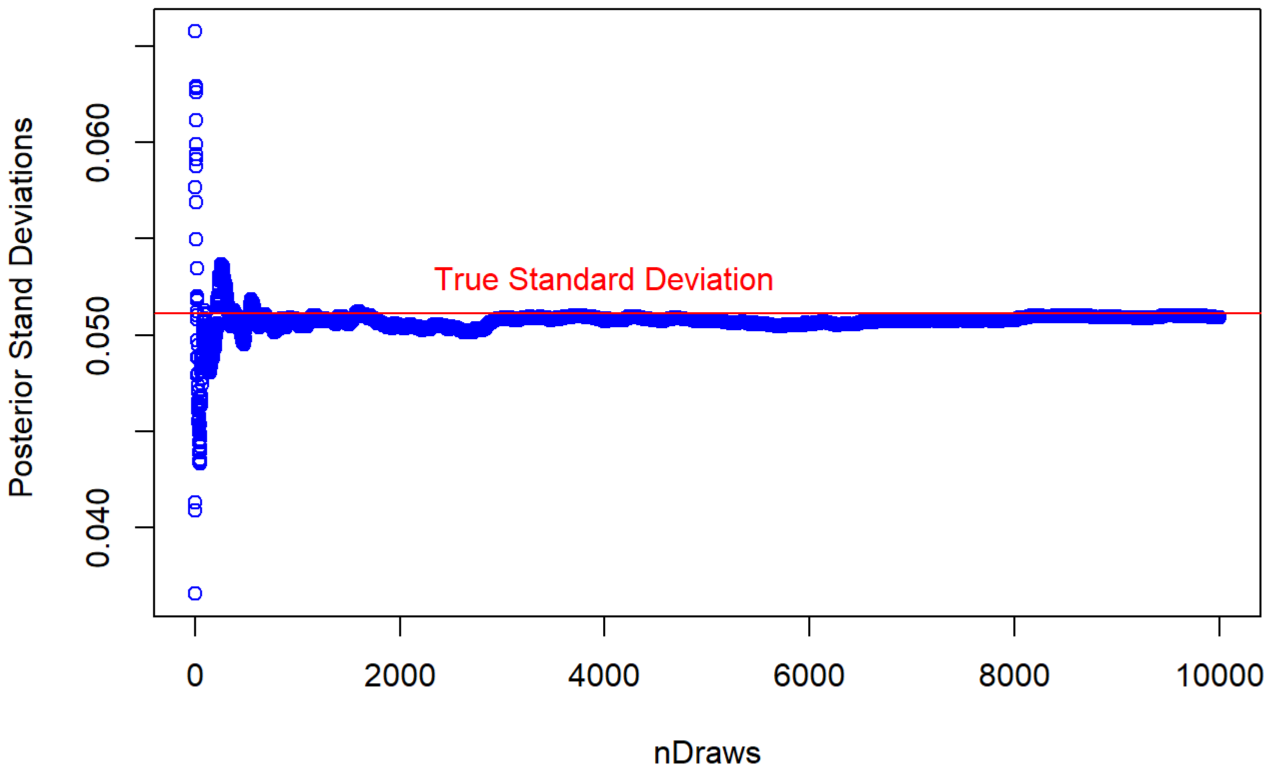
Plotting Means

```
plot(x=1:10000, y=means, xlab = "nDraws", ylab = "Posterior Mean", col="green")
abline(h=true_mean, col="red")
text(x=4000, y=0.352, labels="True Mean", col="red")
```



Plotting Standard deviations

```
plot(x=1:10000, y=sd_s, xlab = "nDraws", ylab=" Posterior Stand Deviations", col="blue")
abline(h=true_sd,col="red")
text(x=4000, y=0.053, labels="True Standard Deviation", col="red")
```



The above plots show that the posterior mean E [θ|y] and standard deviation SD [θ|y] converges to the true values as the number of random draws grows large.

1(b)

Computing posterior probability $Pr(\theta > 0.3|y)$

```
#1(b)
set.seed(1234567)

#Drawing 10000 random values from the posterior
random_samples <- rbeta(nDraws,alpha_zero+s,beta_zero+f)

#computing posterior probability Pr(ϑ > 0.3|y)
posterior_prob <- mean(random_samples>0.3)

print(posterior_prob)
```

```
## [1] 0.8296
```

Computing the exact value from the Beta posterior

```
#computing exact posterior probability from Beta posterior
exact_prob <- pbeta(0.3, alpha_zero+s, beta_zero+f, lower.tail = FALSE)
print(exact_prob)
```

```
## [1] 0.8285936
```

Both the values are almost same.

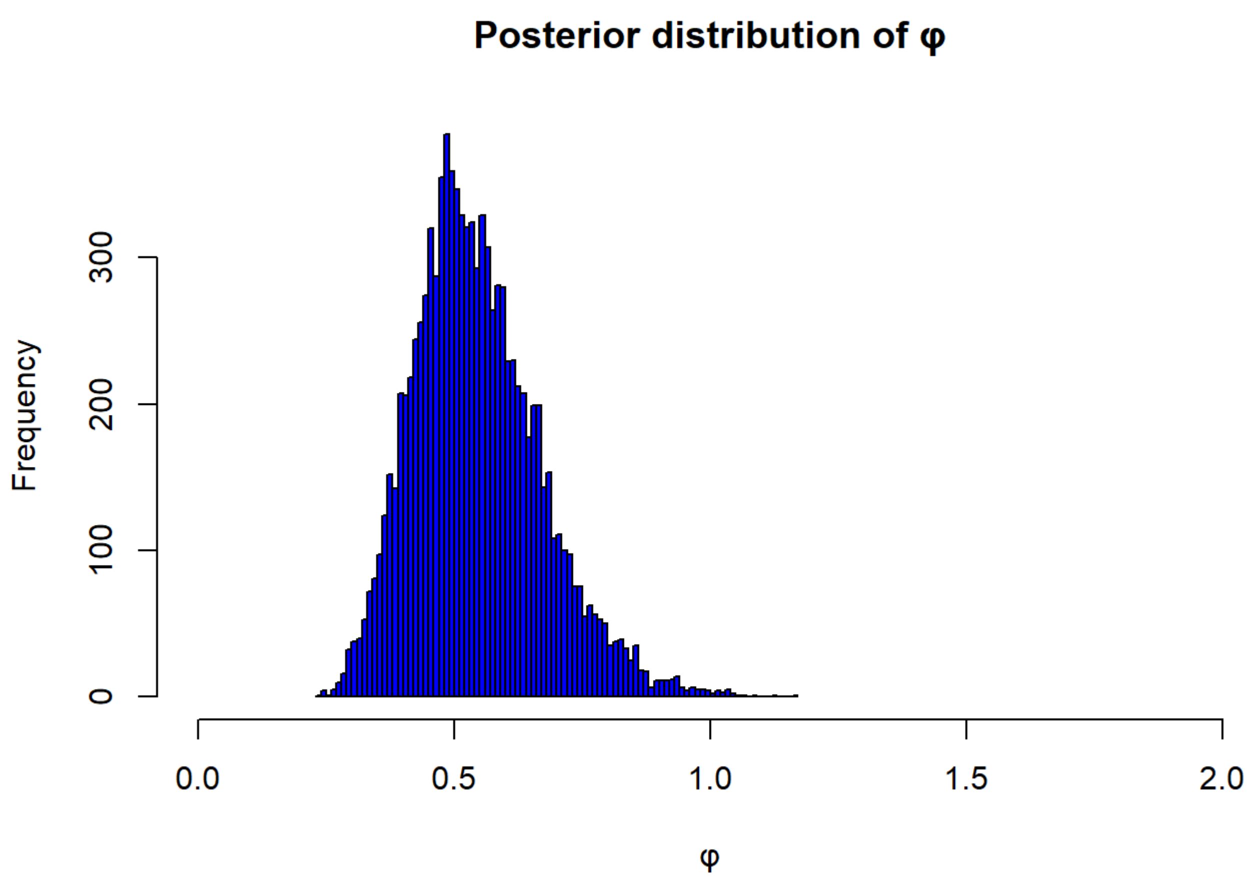
1(c)

Draw 10000 random values from the posterior of the odds $\phi = \frac{\theta}{1-\theta}$

```
#1(c)
theta <-random_samples # copying random samples from 1(b)
phi_odds <- theta/(1-theta)
```

Plotting the posterior distribution of ϕ

```
# plotting the posterior distribution of φ
hist(phi_odds, breaks = 100, col = "blue", main = "Posterior distribution of ϕ",xlab = "ϕ",xlim
= c(0, 2))
```



2.Log-normal distribution and the Gini coefficient.

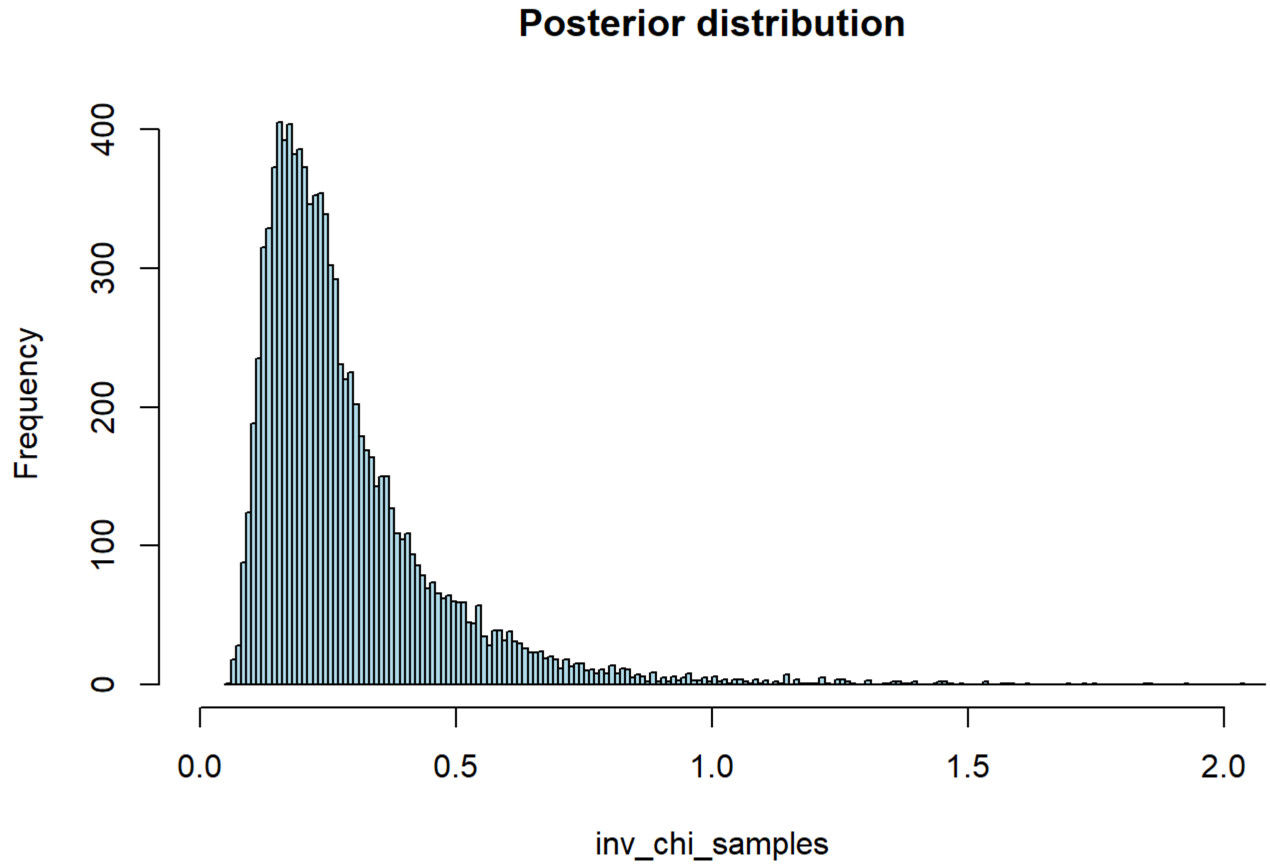
2(a)

Drawing 10000 random values from the posterior of σ^2 by assuming μ = 3.6

```
#2(a)
library(LaplacesDemon)
observations <- c(33, 24, 48, 32, 55, 74, 23, 17)
library(LaplacesDemon)
n = length(observations)
mu= 3.6
log_y = log(observations)
tao_sq = (sum((log_y - mu)^2))/n

#drawing 10000 samples from posterior distribution
inv_chi_samples<- rinvchisq(10000, n, tao_sq)
```

Plotting posterior distribution

```
#plotting posterior distribution
hist(inv_chi_samples, breaks = 300, col = "lightblue", main = "Posterior distribution", xlim=c
(0,2))
```



2(b)

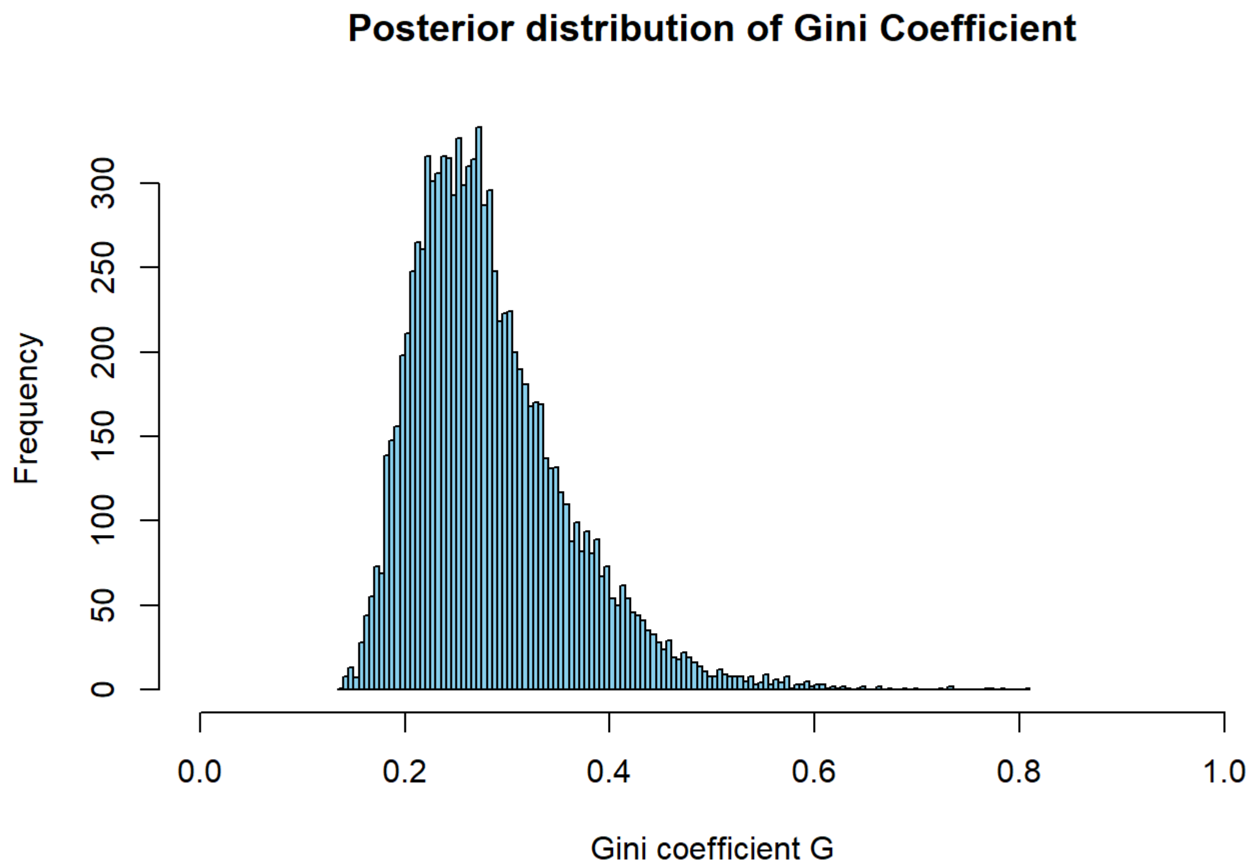
Gini coefficient $G = 2\Phi(\sigma/\sqrt{2}) - 1$ where $\Phi(\sqrt{\frac{\sigma^2}{2}})$ is cumulative distribution function (CDF) for the standard normal distribution with mean zero and unit variance in which \$ ^2 \$ is the posterior distribution $Inv - \chi^2(n, \tau^2)$

Computing posterior distribution of Gini Coefficient

```
#2(b)
# Calculating CDF for the standard normal distribution
phi <- pnorm(sqrt(inv_chi_samples/2), mean=0, sd=1)

#compute the posterior distribution of the Gini coefficient G
G <- 2* phi  - 1
```

Plotting posterior distribution of Gini Coefficient

```
#plotting posterior distribution
hist(G, breaks = 200, col = "skyblue", main = "Posterior distribution of Gini Coefficient", xlim
=c(0,1), xlab="Gini coefficient G")
```



2(c)

$Credible\ interval$: A credible interval is a range of values that contains a certain proportion (95% in this case) of the possible values for a parameter, given the data and a specified prior distribution.

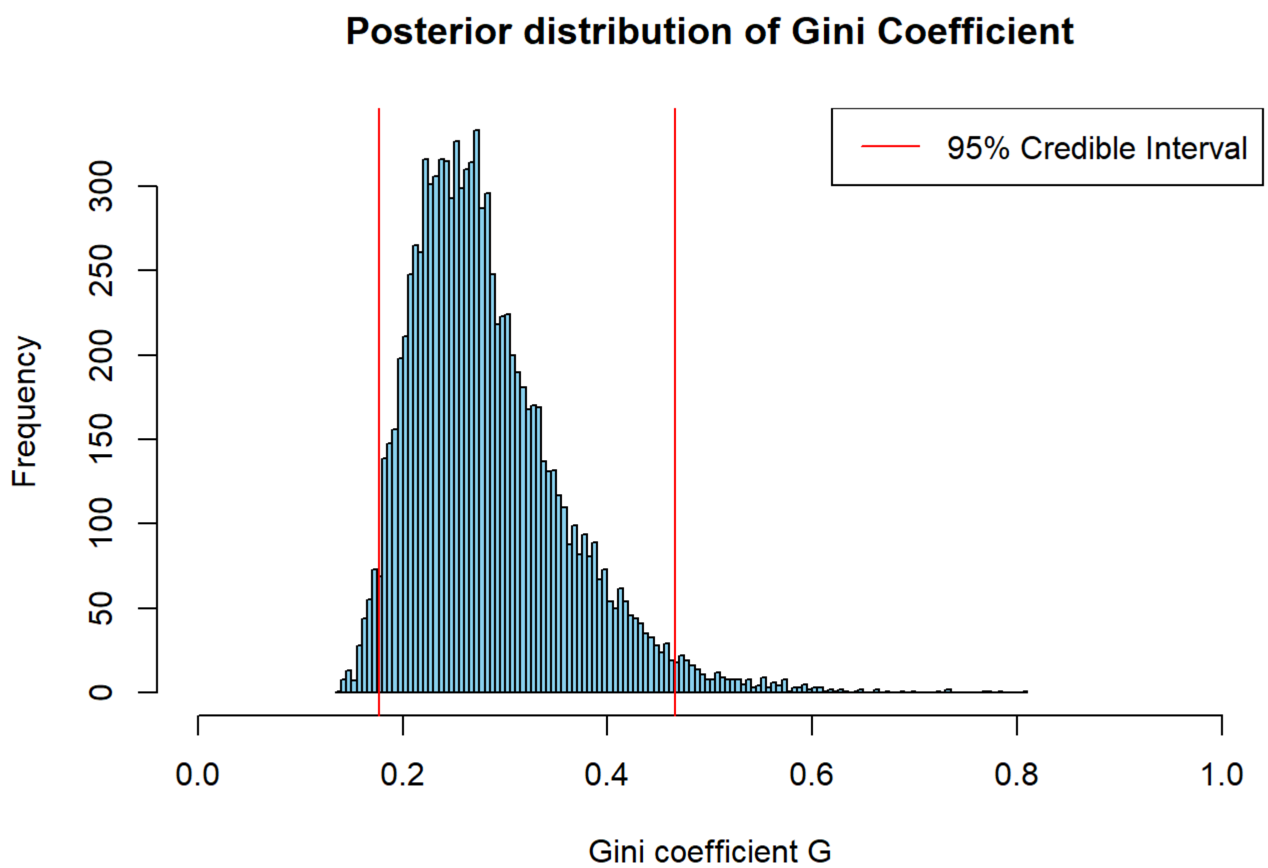
$Calculating\ Credible\ interval$: Calculate the quantiles of the posterior distribution that contain the desired proportion of the distribution, such as the 2.5th and 97.5th percentiles for a 95% credible interval.

```
#2(c)
# Computing the lower bound and upper bound of the credible interval
lower <- quantile(G, probs = 0.025)
upper <- quantile(G, probs = 0.975)

# Print the credible interval
cat("The 95% equal tail credible interval for G is (", lower, ",", upper, ")")
```

```
## The 95% equal tail credible interval for G is ( 0.176404 , 0.4654876 )
```

Plotting credible interval

```r
# Create a histogram of the posterior distribution of Gini coefficient
hist(G, breaks = 200, col = "skyblue", main = "Posterior distribution of Gini Coefficient", xlim
=c(0,1), xlab="Gini coefficient G")

# Add vertical lines at the lower and upper bounds of the credible interval
abline(v = lower, col = "red")
abline(v = upper, col = "red")
legend("topright", legend = c("95% Credible Interval"),
       lty = 1, col = "red")
```



2(d)

$Highest\ Posterior\ Density\ Interval\ (HPDI)$: The HPD interval is defined as the shortest interval on the posterior distribution that contains a specified probability mass (95% in this case).

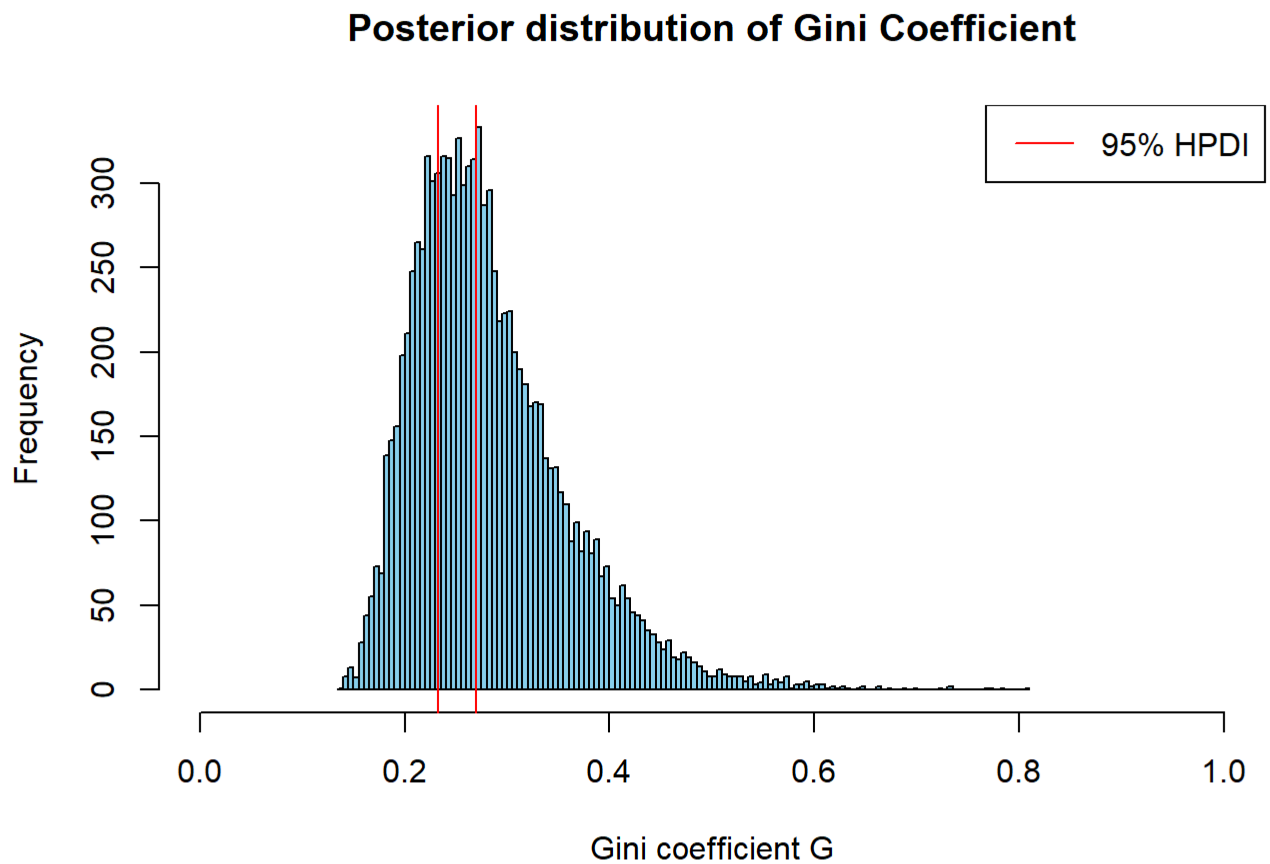
$Computing\ Highest\ Posterior\ Density\ Interval\ (HPDI)$:

```r
#2(d)
# Estimating the posterior density of Gini coefficient using kernel density
densty <- density(G)

alpha <- 0.95 #95% of HPDI
sorted_densty <- sort(densty$y)
threshold <- sorted_densty[round(alpha * length(sorted_densty))]

hpdi_lower <- min(densty$x[densty$y >= threshold]) #lower bound
hpdi_upper <- max(densty$x[densty$y >= threshold]) #upper bound

# Print the HPDI
cat("The 95% HPDI for G is (", hpdi_lower, ",", hpdi_upper, ")\n")
```

```
## The 95% HPDI for G is ( 0.2323471 , 0.2695589 )
```

Plotting HPDI

```r
# Create a histogram of the posterior distribution of Gini coefficient
hist(G, breaks = 200, col = "skyblue", main = "Posterior distribution of Gini Coefficient", xlim
=c(0,1), xlab="Gini coefficient G")

# Add vertical lines at the lower bound and upper bound of the HDPI interval
abline(v = hpdi_lower, col = "red")
abline(v = hpdi_upper, col = "red")
legend("topright", legend = c("95% HPDI"),
       lty = 1, col = "red")
```



The equal tail credible interval for Gini Coefficient was (0.1759587, 0.4642831), while the HPDI is (0.2297412 , 0.2677448). The HPDI is narrower than the credible interval, and its location is shifted slightly to the left. This shows that the most probable range of values for G is slightly lower than the midpoint of the credible interval.

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

3(a)

The posterior distribution is given by

$$p(k|y,\mu) \propto p(y|\mu,k).\,p(k)$$

where $p(k|y,\mu)$ is likelihood function and $p(k)$ is prior distribution of k.

$$p(y|\mu,k) = \prod_{i=1}^{10} \frac{exp[k.\,cos(y_i-\mu)]}{2\pi I_0(k)}$$

where $I_0(k)$ is the modified Bessel functionof order zero and μ=2.4 (given)

$$p(k) \sim Exponential(\lambda = 0.5)$$

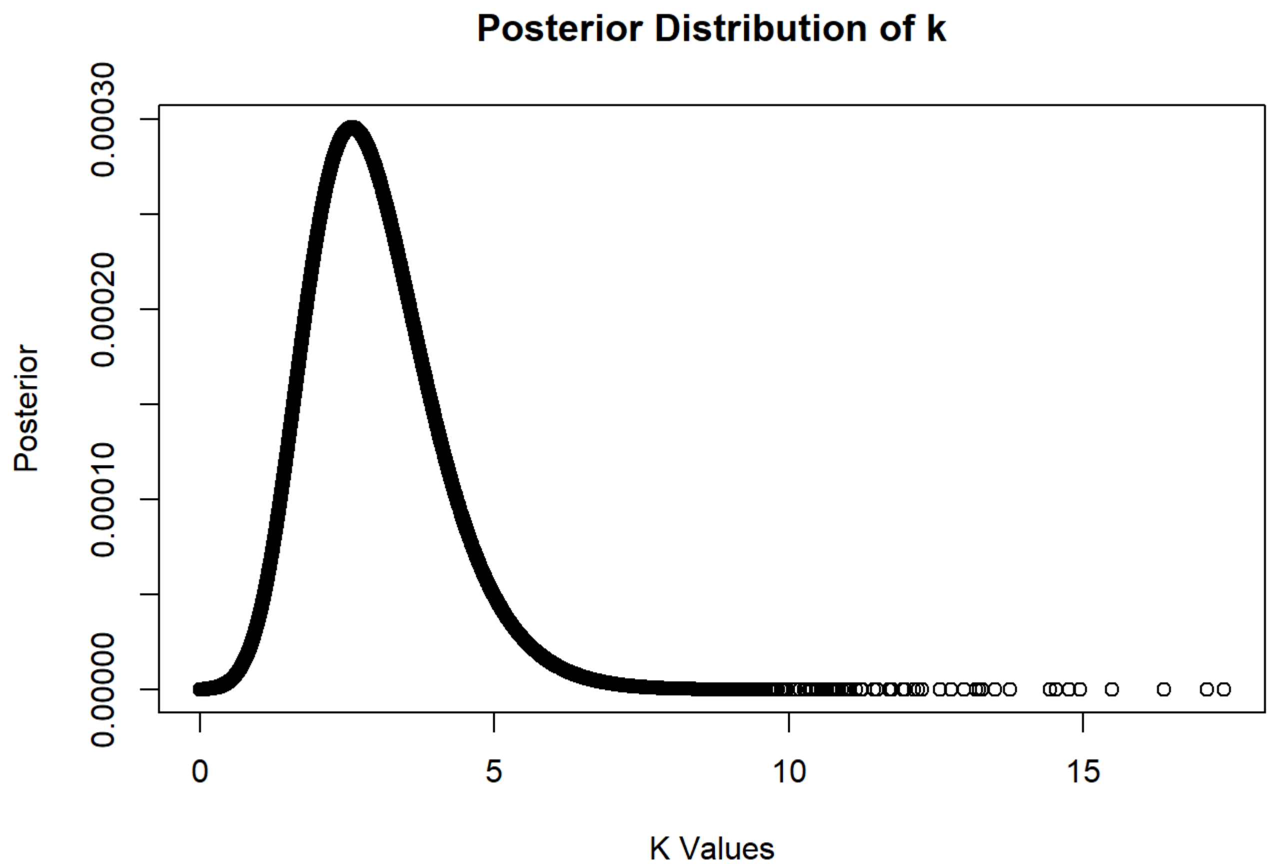
$$p(k) = \lambda.\, exp(-\lambda.\, k)\ ,\ \ \lambda = 0.5$$

The un-normalized posterior distribution of k is given by

$$\begin{aligned} p(k|y_1 \ldots y_{10}) &\propto p(y_1 \ldots y_{10}|\mu, k) * p(k) \\ &= \prod_{i=1}^{10} \frac{exp[k.\, cos(y_i - \mu)]}{2\pi I_0(k)} * \lambda.\, exp(-\lambda k) \\ &= \frac{exp\left[k.\sum_{1=1}^{10} cos(y_i - \mu)\right]}{(2\pi I_0(k))^{10}} * \lambda.\, exp(-\lambda k) \\ p(k|y_1 \ldots y_{10}) &= \frac{0.5 * exp\left[k(-0.5 + \sum_{i=0}^{10} cos(y_i - \mu)\right]}{(2\pi I_0(k))^{10}} \end{aligned}$$

Now lets plot the posterior distribution of κ for the wind direction data over a fine grid of κ values.

```
#3(a)
set.seed(1234567)

#observations in radians
observ_rad <- c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)

mu <- 2.4 #known
n <- length(observ_rad)
nDraws = 10000

# drawing prior k values
k_values <- rexp(10000, rate = 0.5) #given lambda= 0.5
expression1<- -0.5+sum(cos(observ_rad-mu))

#unnormalized posterior
posterior_dist <- 0.5 * exp(k_values * expression1) / (2*pi*besselI(x=k_values,nu=0))^10

posterior_norm <- posterior_dist/ sum(posterior_dist) #normalized

plot(x= k_values, y=posterior_norm, xlab="K Values", ylab="Posterior", main="Posterior Distribut
ion of k")
```



3(b)

The posterior mode of k is the value of k that corresponds to the maximum value of the posterior density function.

```
#3(b)
mode = k_values[which.max(posterior_norm)]
cat("The posterior mode of k is ",mode )
```

```
## The posterior mode of k is  2.586644
```