Bayesian Learning Lab-1

Vinay Bengaluru(vinbe289), Tejashree R Mastamardi(tejma768)

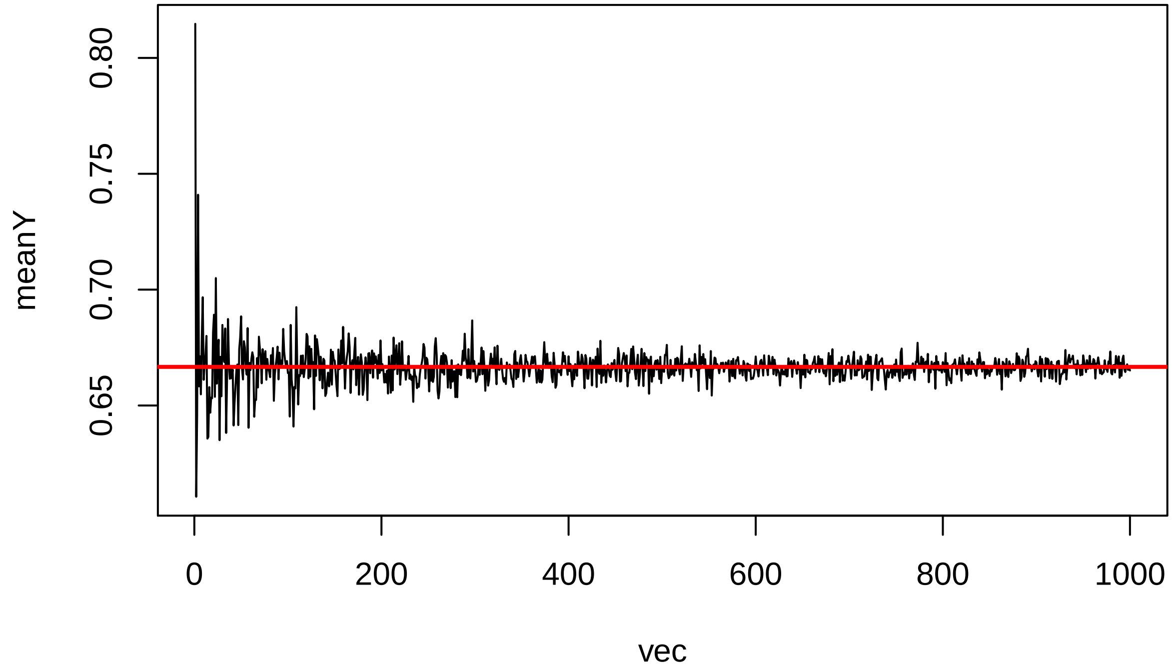
April 14, 2019

1

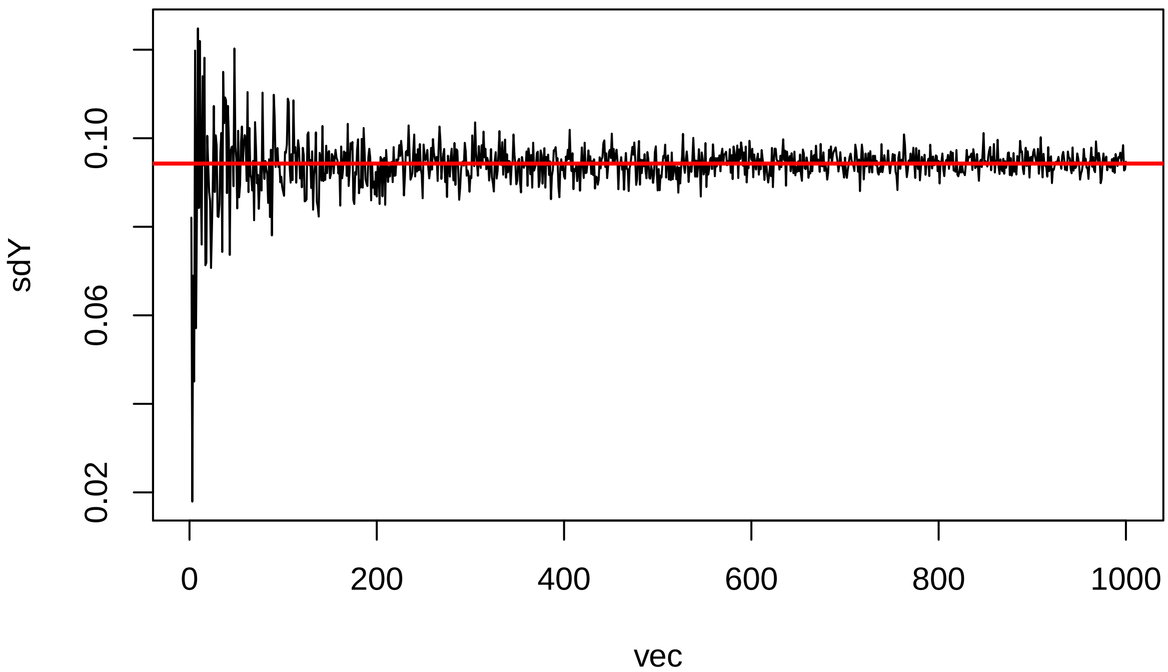
a Draw random numbers from the posterior and verify graphically that the posterior mean and standard deviation

converges to the true values as the number of random draws grows large.

```
#1a
# Given
alpha_0 <- 2
beta_0 <- 2
s <- 14
f <- 6
alpha <- alpha_0+s
beta <- beta_0+f

# Mean calculated using the formula
trueMean <- alpha/(alpha+beta)
trueSD <- sqrt(alpha*beta/((alpha+beta)^2*(alpha+beta+1)))

# Mean calculated by taking the average of random numbers generated
# using posterior Beta distribution
meanY <- c()
sdY <- c()
Y <- c()
vec <- seq(1,1000,1)
for(i in 1:length(vec)){
  n <- vec[i]
  Y <- rbeta(n,alpha_0+s,beta_0+f)
  meanY[i] <- mean(Y)
  sdY[i] <- sd(Y)
}
plot(x = vec,y = meanY,type = "l")
abline(h=trueMean,col="red",lwd=2)
```




```
plot(x = vec,y = sdY,type = "l")
abline(h=trueSD,col="red",lwd=2)
```



It is evident graphically that the posterior mean and standard deviation converges to the true values as the number of random draws grows large.

b Use simulation (nDraws = 10000) to compute the posterior probability and compare with the exact value

```r
#1b
exactPostprob <- pbeta(0.4,alpha_0+s,beta_0+f)
generateValue <- rbeta(10000,alpha_0+s,beta_0+f)
noOfValues <- generateValue[which(generateValue<0.4)]
calPostprob <- length(noOfValues)/10000
cat("\n Exact value of posterior probability:",exactPostprob)
```

```
##
##  Exact value of posterior probability: 0.003972681
```

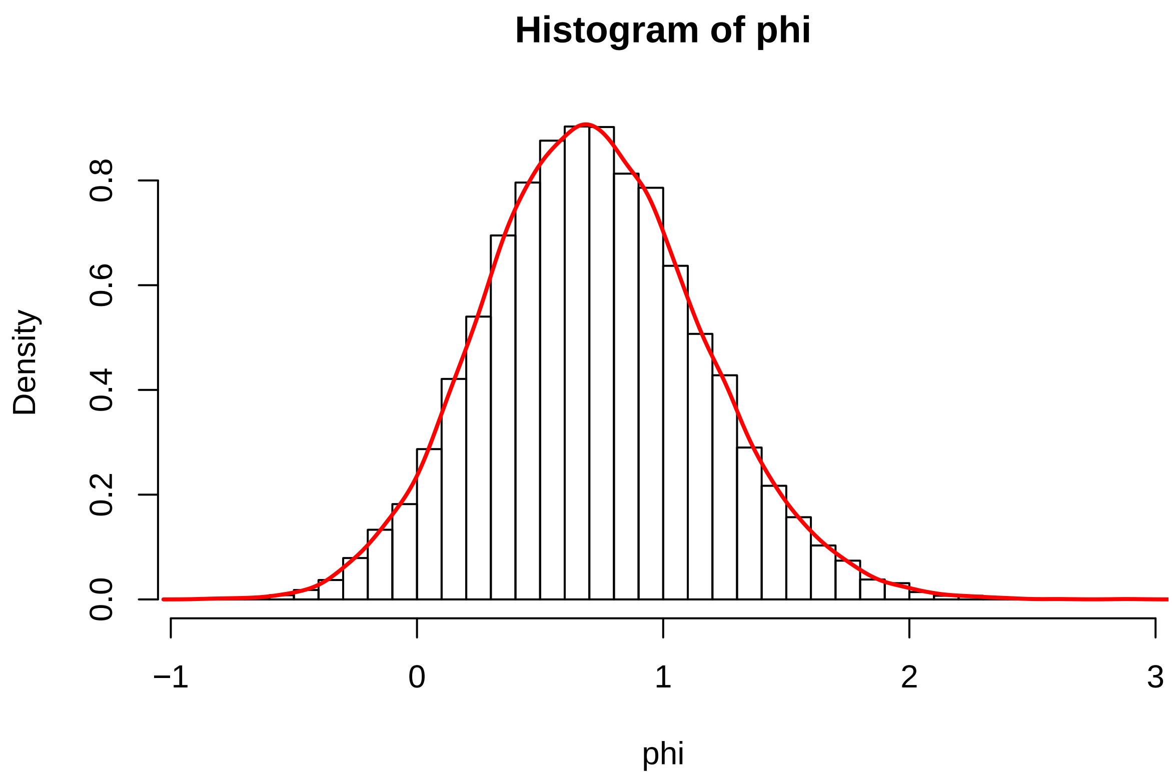
```r
cat("\n Posterior probability using simulation:",calPostprob)
```

```
##
##  Posterior probability using simulation: 0.0031
```

c Compute the posterior distribution of the log-odds by simulation

```r
#1c
theta <- generateValue
phi <- c()
for(i in 1:10000){
```

```
  phi[i] <- log(theta[i]/(1-theta[i]))
}
hist(phi,breaks = 50,freq = F)
lines(density(phi),col="red",lwd=2)
```



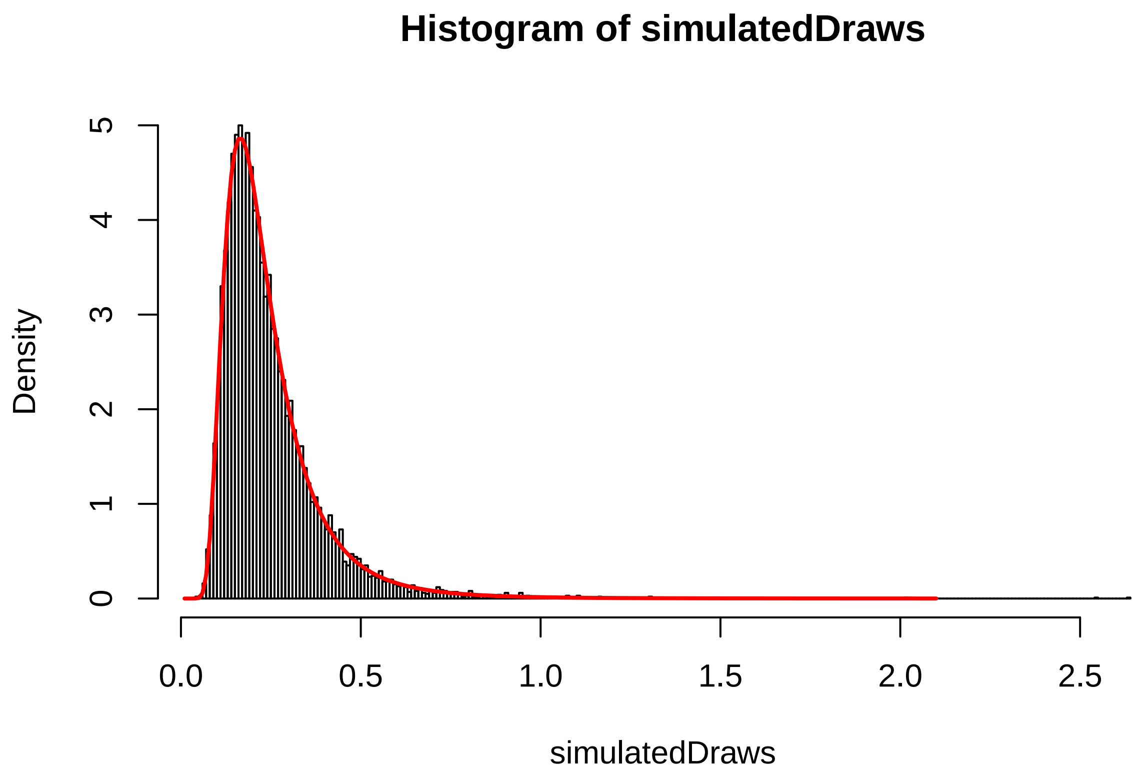
The posterior distribution of the log-odds by simulation is shown in the plot.

2

a Simulate 10,000 draws from the posterior and compare with the theoretical Inv posterior distribution.

```
#2a
library(geoR)

y_i <- c(14, 25, 45, 25, 30, 33, 19, 50, 34 ,67)
len <- length(y_i)
mu <- 3.5
n_draws <- 10000
tou_sq <- sum((log(y_i)-mu)^2)/len

#Simulate 10,000 draws from the posterior of sigma^2 (assuming mu = 3.5)
simulatedDraws <- rinvchisq(n = n_draws,df = len,scale = tou_sq)

x <- seq(0.01,2.1,0.01)
```

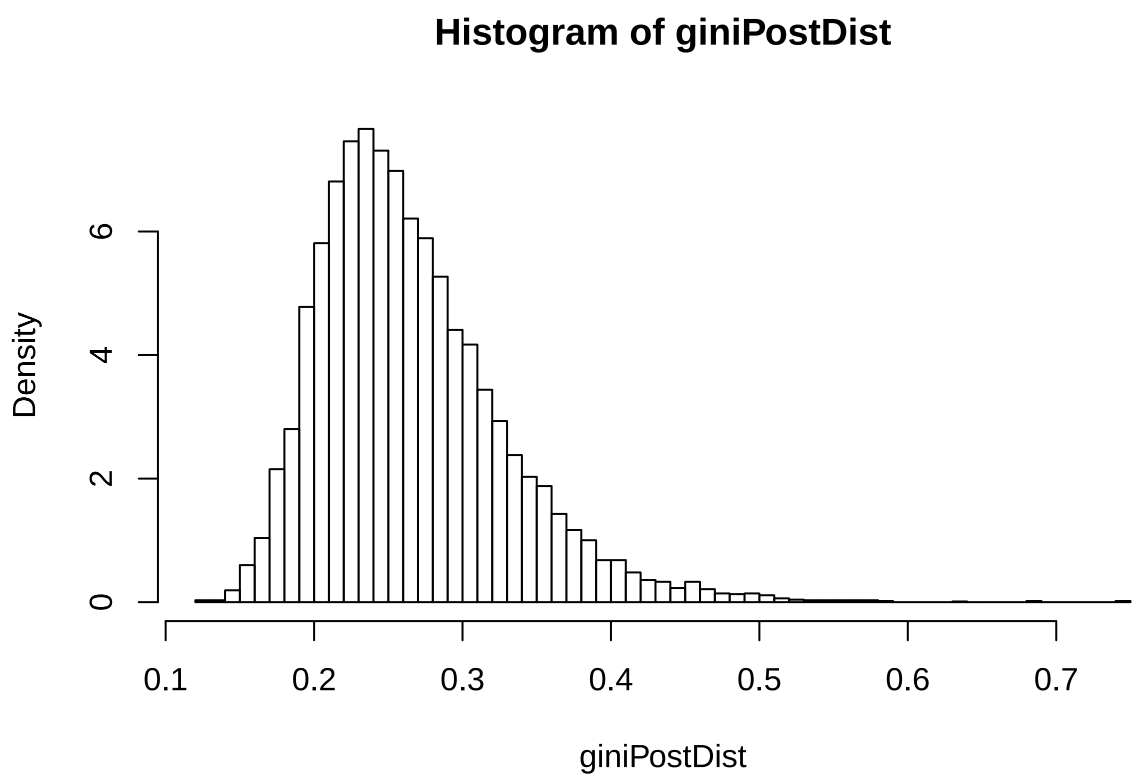
```
#Theoretical Inverse Chi square distribution
theoreticalDraws <- ((((tou_sq*len)/2)^(len/2))/gamma(len/2))*(exp(-(len*tou_sq)/(2*x))/x^(1+(len/2)))

#Comparision
hist(simulatedDraws, breaks = 200,freq = F)
lines(x = x,y = theoreticalDraws,col = "red",lwd=2)
```



The comparision of posterior of σ^2 with theoretical Inv $\chi^2(n, \tau^2)$ posterior distribution is shown in the plot. The histogram corresponds to posterior of σ^2 and the curve in Red corresponds to theoretical Inv $\chi^2(n, \tau^2)$ posterior distribution.

b Use the posterior draws in a) to compute the posterior distribution of the Gini coefficient G for the current data set.

```
#2b
sigma <- sqrt(simulatedDraws)
#We take the squareroot of simulatedDraws since it contains the posterior of sigma square
#but we have phi(sigma/sqrt(2))
giniPostDist <- 2*pnorm(sigma/sqrt(2))-1

#to compute the posterior distribution of the Gini coefficient G for the current data set
hist(giniPostDist,breaks = 50,freq = F)
```



Posterior distribution of the Gini coefficient G is shown in the plot.

c Use the posterior draws from b) to compute a 95% equal tail credible interval for G. Compare the two intervals.

```r
#2c
#95% equal tail credible interval for G
giniPostDrawSort <- sort(giniPostDist) #Posterior draws from b) sorted
#Compute a 95% equal tail credible interval for G
giniCredInt <- quantile(giniPostDrawSort,probs = c(0.025,0.975))
giniCredInt
```

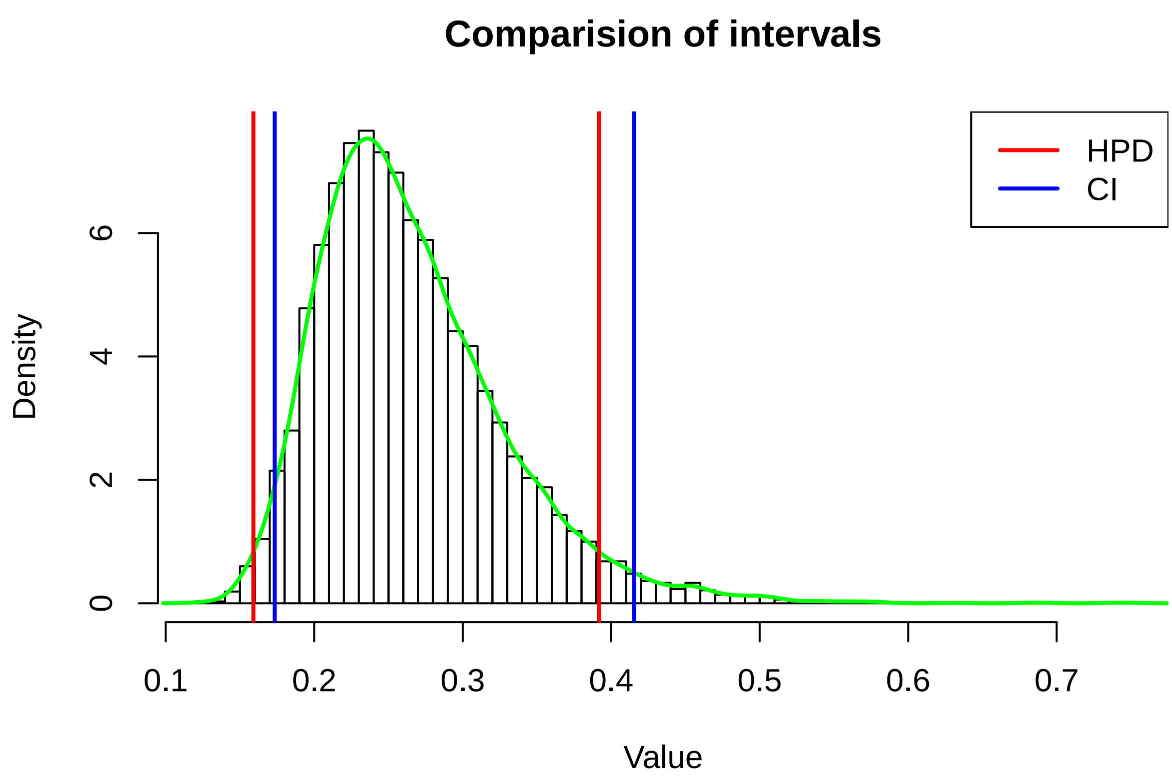
```
##      2.5%     97.5%
## 0.1733281 0.4153340
```

```r
#Kernel density estimate of the posterior of G using the density function in
# R with default settings
kerDenG <- density(giniPostDist)
kerDenG_dF <- data.frame(x=kerDenG$x,y=kerDenG$y)
kerDenG_dF_sort <- kerDenG_dF[order(kerDenG_dF$y),]
denPerG <- cumsum(kerDenG_dF_sort$y)/sum(kerDenG_dF_sort$y)

hpdG <- kerDenG_dF_sort[which(denPerG >= 0.05),] #extracting 95% density starting from the top

ciLow <- min(hpdG$x)
ciUp <- max(hpdG$x)
hpdInt <- c(ciLow,ciUp)
```

```r
hpdInt #lower and upper HPD intervals
```

```
## [1] 0.1590099 0.3918389
```

```r
#Compare the two intervals
hist(giniPostDist,breaks = 50,freq = F,main = "Comparision of intervals",xlab = "Value")
lines(density(giniPostDist),col = "green",lwd = 2)
legend("topright",c("HPD","CI"),col = c("red","blue"),lwd = 2)
abline(v = hpdInt,col = "red",lwd = 2)
abline(v = giniCredInt,col = "blue",lwd = 2)
```



The comparision of Highest posterior density interval and confidence interval is shown in the plot.

3

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

Having given Von Mises Distribution:

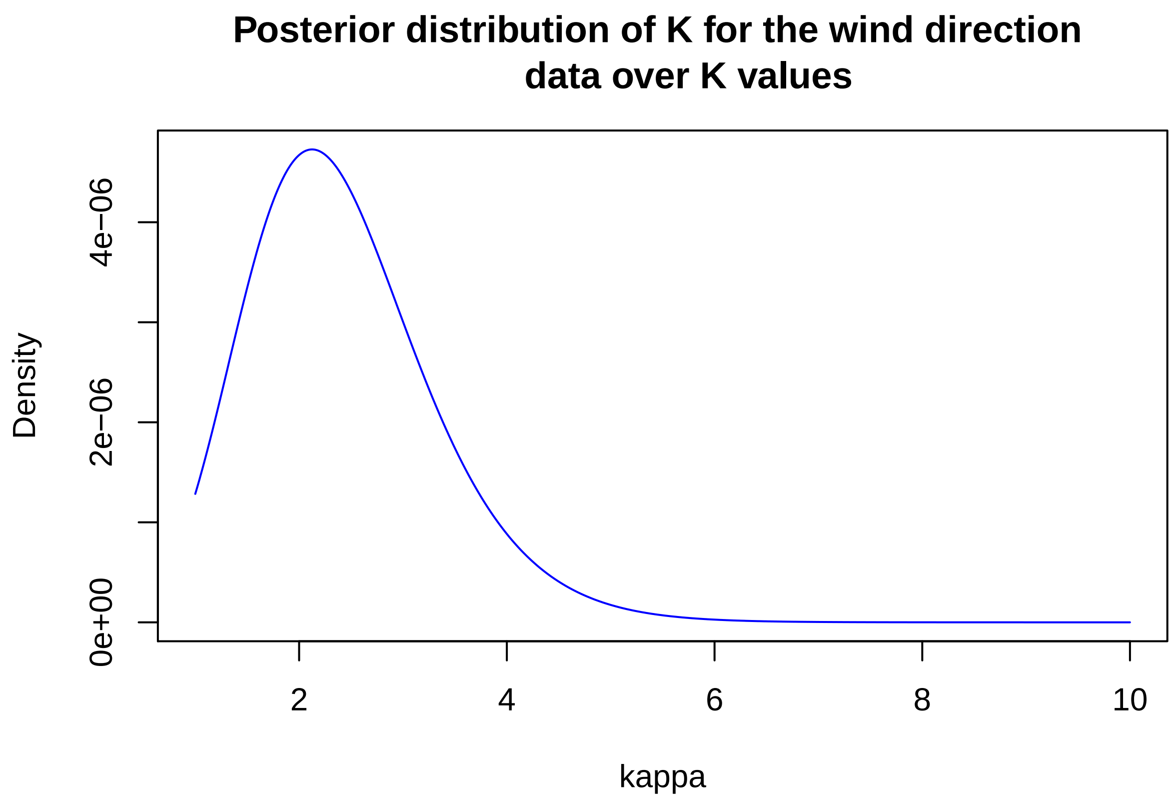
$$p(y|\mu, K) = \frac{\exp[K.cos(y-\mu)]}{2\pi * I_0(K)}, -\pi \leq y \leq \pi$$

we got to find the posterior distribution of K

$$p(y|\mu, K) \propto p(y|\mu, K)p(K)$$

```r
#3a
VonMises_dist <- function(K, y ,mu)
```

```r
{
exp(K * cos(y - mu)) / (2 * pi * besselI(K, nu = 0))
}
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
Values_of_k <- seq(1 ,10 ,by = 0.01)
Misval <- t(sapply(y , FUN = function(X) VonMises_dist(K = Values_of_k, y = X, mu = 2.39) ))
Prior <- dexp(Values_of_k, rate = 1)
Likelihood <- apply(Misval, MARGIN = 2, FUN = prod)
Posterior <- Likelihood * Prior

#Plot of posterior distribution of K for the wind direction data over a fine grid of K values
plot(Values_of_k, Posterior, type="l", main="Posterior distribution of K for the wind direction
     data over K values", ylab="Density", xlab="kappa", cex=0.5, col = "blue")
```



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

```r
#3b
cat("\n The approximate Posterior mode of K is:",Values_of_k[which.max(Posterior)])
```

```
##
##  The approximate Posterior mode of K is: 2.12
```

Appendix

```r
#1a
# Given
alpha_0 <- 2
beta_0 <- 2
s <- 14
f <- 6
alpha <- alpha_0+s
beta <- beta_0+f

# Mean calculated using the formula
trueMean <- alpha/(alpha+beta)
trueSD <- sqrt(alpha*beta/((alpha+beta)^2*(alpha+beta+1)))

# Mean calculated by taking the average of random numbers generated
# using posterior Beta distribution
meanY <- c()
sdY <- c()
Y <- c()
vec <- seq(1,1000,1)
for(i in 1:length(vec)){
  n <- vec[i]
  Y <- rbeta(n,alpha_0+s,beta_0+f)
  meanY[i] <- mean(Y)
  sdY[i] <- sd(Y)
}
plot(x = vec,y = meanY,type = "l")
abline(h=trueMean,col="red",lwd=2)

plot(x = vec,y = sdY,type = "l")
abline(h=trueSD,col="red",lwd=2)
#1b
exactPostprob <- pbeta(0.4,alpha_0+s,beta_0+f)
generateValue <- rbeta(10000,alpha_0+s,beta_0+f)
noOfValues <- generateValue[which(generateValue<0.4)]
calPostprob <- length(noOfValues)/10000
cat("\n Exact value of posterior probability:",exactPostprob)
cat("\n Posterior probability using simulation:",calPostprob)
#1c
theta <- generateValue
phi <- c()
for(i in 1:10000){
  phi[i] <- log(theta[i]/(1-theta[i]))
}
hist(phi,breaks = 50,freq = F)
lines(density(phi),col="red",lwd=2)
#2a
library(geoR)

y_i <- c(14, 25, 45, 25, 30, 33, 19, 50, 34 ,67)
len <- length(y_i)
mu <- 3.5
n_draws <- 10000
tou_sq <- sum((log(y_i)-mu)^2)/len
```

```r
#Simulate 10,000 draws from the posterior of sigma^2 (assuming mu = 3.5)
simulatedDraws <- rinvchisq(n = n_draws,df = len,scale = tou_sq)

x <- seq(0.01,2.1,0.01)
#Theoretical Inverse Chi square distribution
theoreticalDraws <- ((((tou_sq*len)/2)^(len/2))/gamma(len/2))*(exp(-(len*tou_sq)/(2*x))/x^(1+(len/2)))

#Comparision
hist(simulatedDraws, breaks = 200,freq = F)
lines(x = x,y = theoreticalDraws,col = "red",lwd=2)
#2b
sigma <- sqrt(simulatedDraws)
#We take the squareroot of simulatedDraws since it contains the posterior of sigma square
#but we have phi(sigma/sqrt(2))
giniPostDist <- 2*pnorm(sigma/sqrt(2))-1

#to compute the posterior distribution of the Gini coefficient G for the current data set
hist(giniPostDist,breaks = 50,freq = F)
#2c
#95% equal tail credible interval for G
giniPostDrawSort <- sort(giniPostDist) #Posterior draws from b) sorted
#Compute a 95% equal tail credible interval for G
giniCredInt <- quantile(giniPostDrawSort,probs = c(0.025,0.975))
giniCredInt

#Kernel density estimate of the posterior of G using the density function in
# R with default settings
kerDenG <- density(giniPostDist)
kerDenG_dF <- data.frame(x=kerDenG$x,y=kerDenG$y)
kerDenG_dF_sort <- kerDenG_dF[order(kerDenG_dF$y),]
denPerG <- cumsum(kerDenG_dF_sort$y)/sum(kerDenG_dF_sort$y)

hpdG <- kerDenG_dF_sort[which(denPerG >= 0.05),] #extracting 95% density starting from the top

ciLow <- min(hpdG$x)
ciUp <- max(hpdG$x)
hpdInt <- c(ciLow,ciUp)
hpdInt #lower and upper HPD intervals

#Compare the two intervals
hist(giniPostDist,breaks = 50,freq = F,main = "Comparision of intervals",xlab = "Value")
lines(density(giniPostDist),col = "green",lwd = 2)
legend("topright",c("HPD","CI"),col = c("red","blue"),lwd = 2)
abline(v = hpdInt,col = "red",lwd = 2)
abline(v = giniCredInt,col = "blue",lwd = 2)
#3a
VonMises_dist <- function(K, y ,mu)
{
exp(K * cos(y - mu)) / (2 * pi * besselI(K, nu = 0))
}
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
Values_of_k <- seq(1 ,10 ,by = 0.01)
Misval <- t(sapply(y , FUN = function(X) VonMises_dist(K = Values_of_k, y = X, mu = 2.39) ))
```

```r
Prior <- dexp(Values_of_k, rate = 1)
Likelihood <- apply(Misval, MARGIN = 2, FUN = prod)
Posterior <- Likelihood * Prior

#Plot of posterior distribution of K for the wind direction data over a fine grid of K values
plot(Values_of_k, Posterior, type="l", main="Posterior distribution of K for the wind direction
     data over K values", ylab="Density", xlab="kappa", cex=0.5, col = "blue")

#3b
cat("\n The approximate Posterior mode of K is:",Values_of_k[which.max(Posterior)])
```