Bayesian Learning Computer Lab 1

Yi Hung Chen, Jonathan Dorairaj

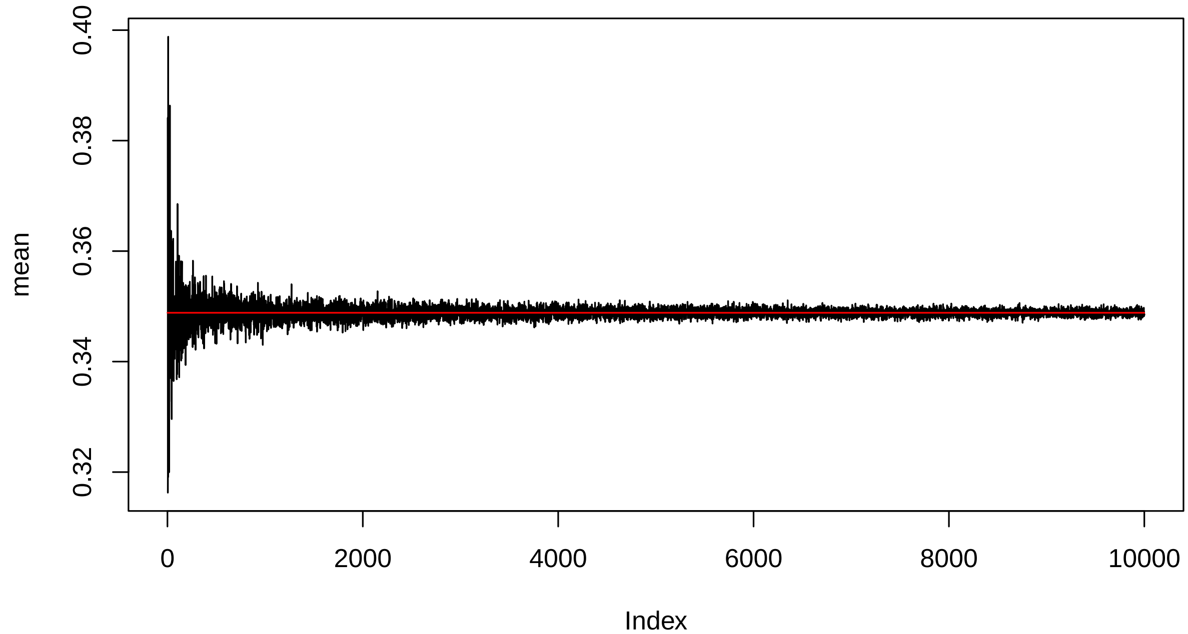
2023-04-14

Question 1

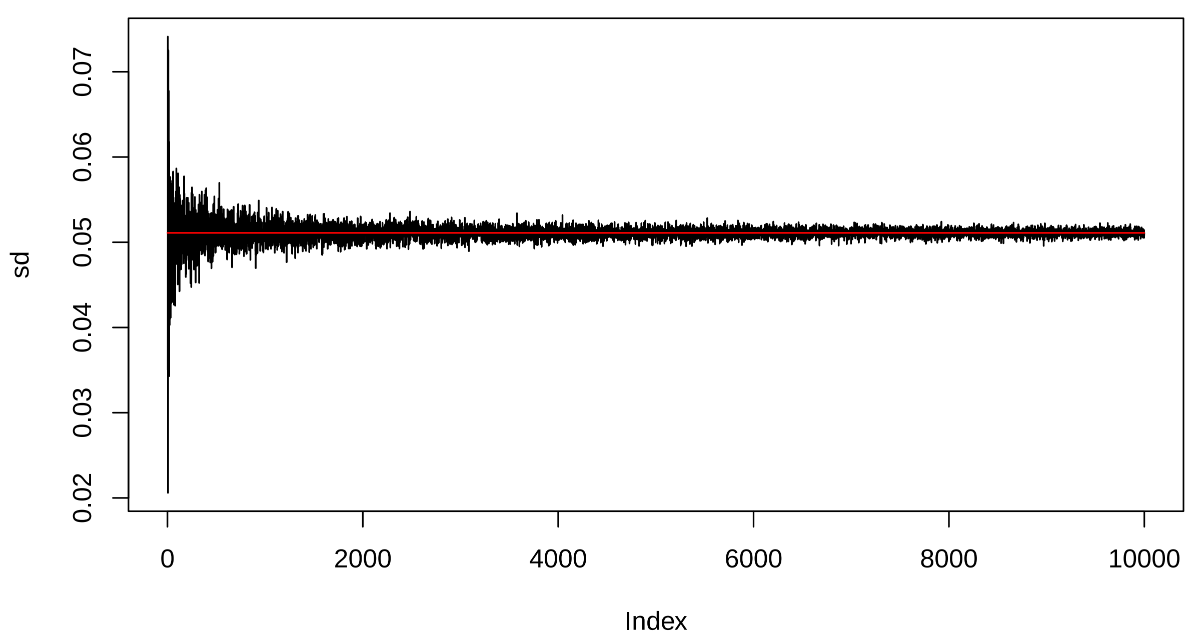
1a)

Draw 10000 random values (nDraws = 10000) from the posterior $\theta|y \sim \text{Beta}(\alpha 0 + s, \beta 0 + f)$, where y = (y1, . . . , yn), and verify graphically that the posterior mean E $[\theta|y]$ and standard deviation SD $[\theta|y]$ converges to the true values as the number of random draws grows large.

```
s <- 22
n <- 70
alpha_0 <- 8
beta_0 <- 8
f <- n-s

alpha <- alpha_0+s
beta <- beta_0+f
real_mean <- alpha/(alpha+beta)
real_sd <- sqrt(alpha*beta/((alpha+beta)**2*(alpha+beta+1)))

mean <- c()
sd <- c()

for(i in seq(0,10000)){
  nDraws <- rbeta(i,alpha,beta)
  mean <- append(mean,mean(nDraws))
  sd <- append(sd,sd(nDraws))
}

plot(mean,type = 'l')
lines(x=seq(0,10000),y=rep(real_mean,10001),col="red")
```




```
plot(sd,type = 'l')
lines(x=seq(0,10000),y=rep(real_sd,10001),col="red")
```



By observing the plots above, it can be seen that the posterior mean and standard deviation converge to the true value(red line).

1b)

Draw 10000 random values from the posterior to compute the posterior probability $\Pr(\theta > 0.3|\mathrm{y})$ and compare with the exact value from the Beta posterior.

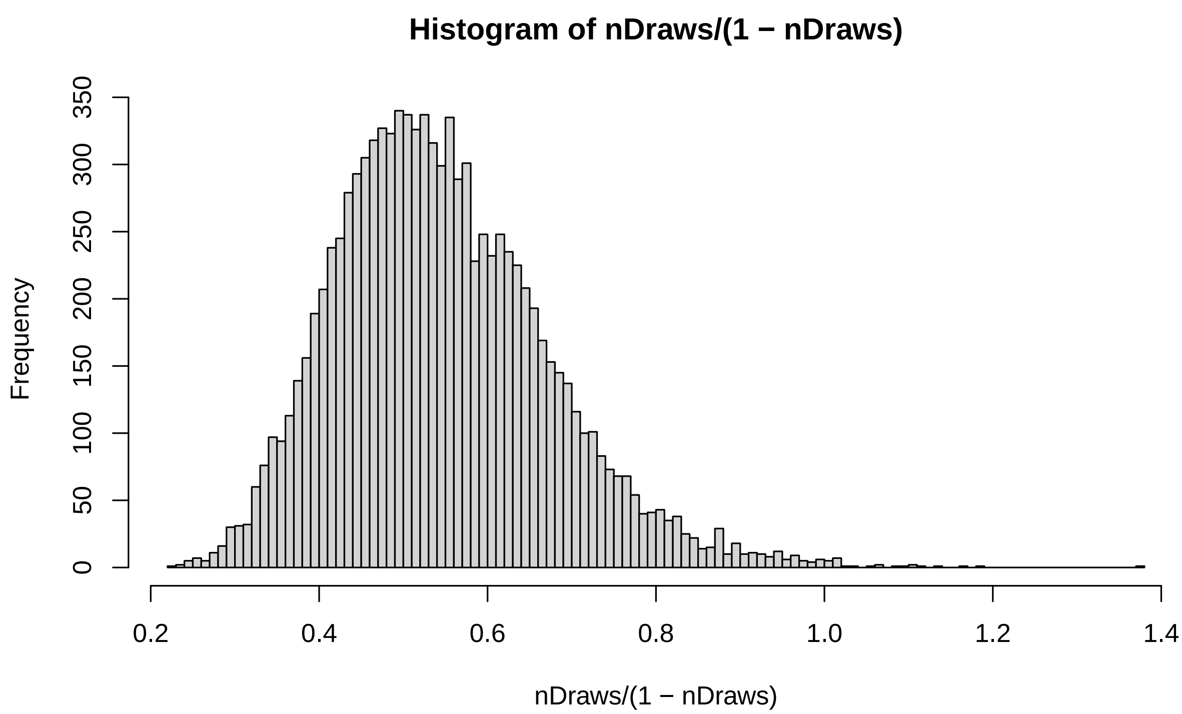
```
real_prob <- pbeta(0.3,alpha,beta,lower.tail = FALSE)
nDraws <- rbeta(10000,alpha,beta)
sample_prob <- sum(nDraws>0.3)/length(nDraws)

cat("The difference bettween real and simulation probability are",sample_prob-real_prob,"which is very
```

```
## The difference bettween real and simulation probability are -0.0008935873 which is very close
```

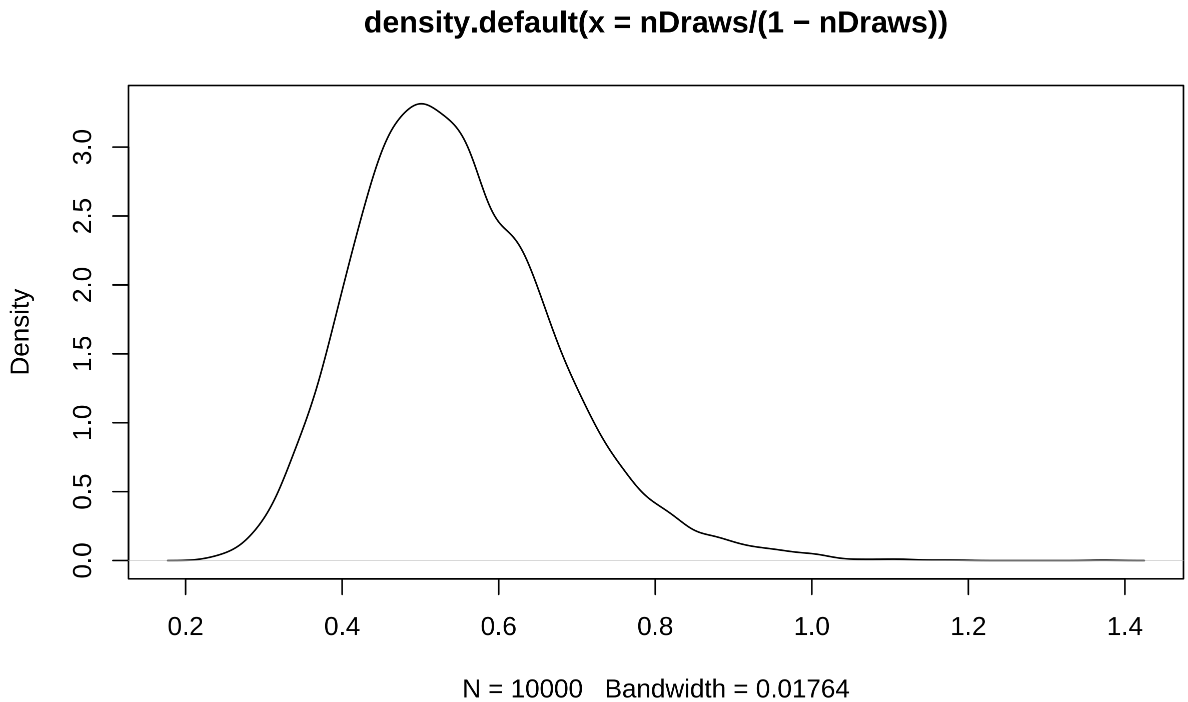
1c)

Draw 10000 random values from the posterior of the odds $\phi = \frac{\theta}{1-\theta}$ by using the previous random draws from the Beta posterior for θ and plot the posterior distribution of ϕ.

```
hist(nDraws/(1-nDraws),breaks = 100)
```




```
plot(density(nDraws/(1-nDraws)))
```


density.default(x = nDraws/(1 − nDraws))
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
N = 10000   Bandwidth = 0.01764

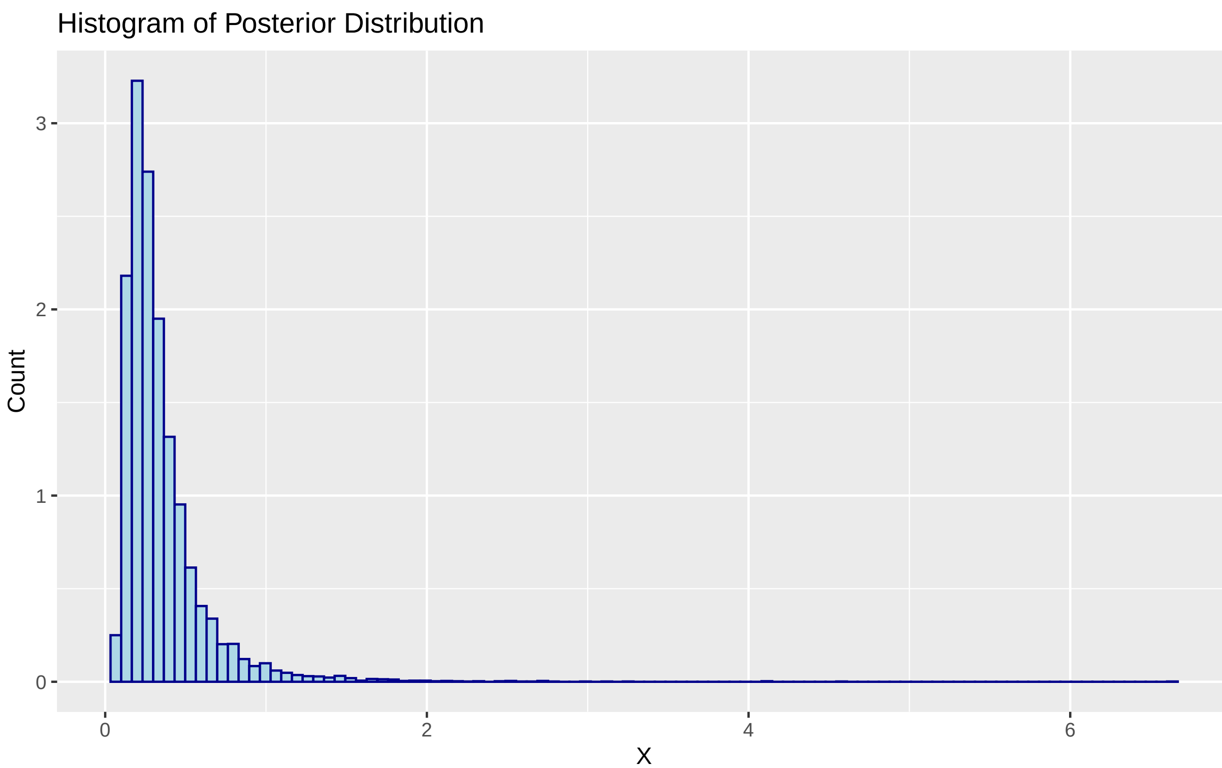
Question 2

Log-normal distribution and the Gini coefficient

2a) Draw 10000 random values from the posterior of σ^2 by assuming $\mu = 3.6$ and plot the posterior distribution.

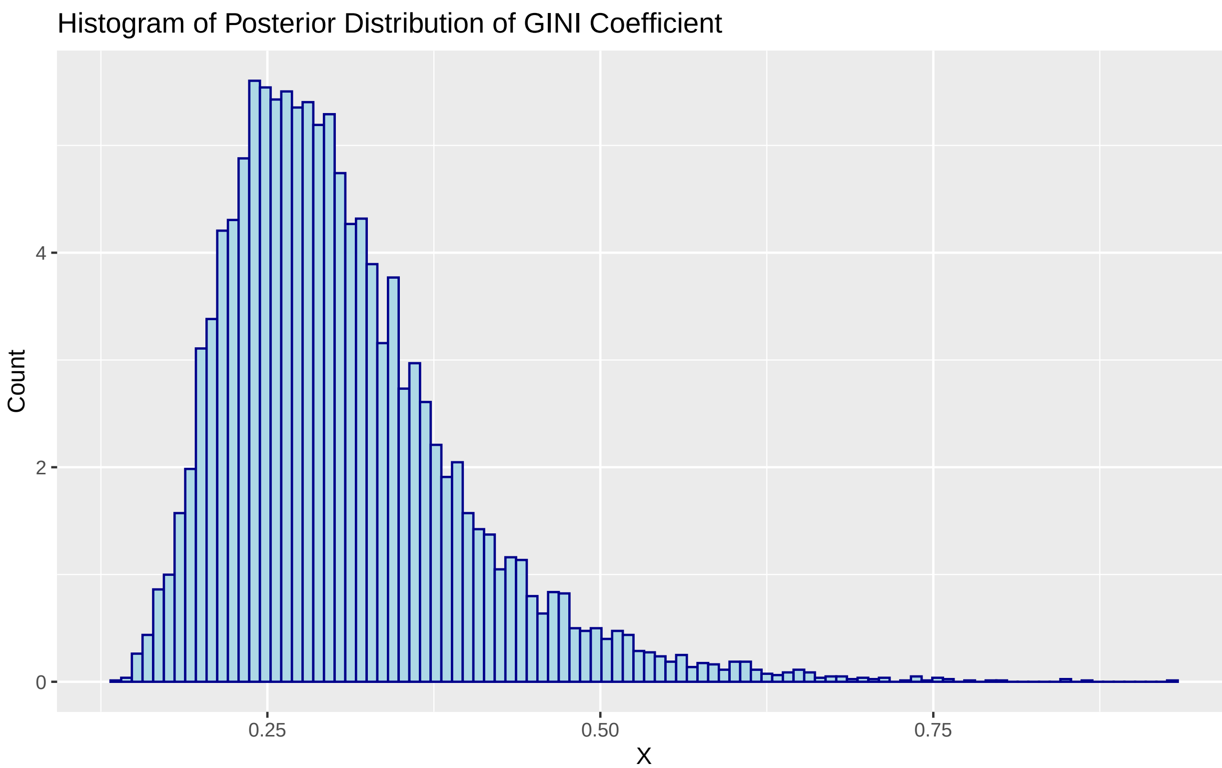
```
obs <- c(33,24,48,32,55,74,23,17)
n <- length(obs)-1

calculate_tau <- function(mu)
{
  res <- (sum((log(obs) - mu)^2))/n
  return(res)
}

tau_2 <- calculate_tau(mu = 3.6)

#draws from chi-sq distribution
X <- rchisq(10000,df = n)

# convert to inverse chi-sq distribution
xs <- (n*tau_2)/X
```

```
xs_df <- as.data.frame(xs)

# histogram
ggplot(data = xs_df, aes(x = xs)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "lightblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution", x = "X", y = "Count")
```



2b Use the posterior draws in 2a) to compute the posterior distribution of the Gini coefficient G for the current data set.

```
phi_z <- sqrt(xs)/sqrt(2)
# Gini coeff
G <- (2 * pnorm(phi_z,mean = 0,sd = 1)) -1

G_df <- as.data.frame(G)

#plotting
ggplot(data = G_df, aes(x = G)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "lightblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution of GINI Coefficient", x = "X", y = "Count")
```



2c Use the posterior draws in 2b) to compute a 95% equal tail credible interval for G.

```
# 2.5 % each side because it is 2 tailed

lower_b <- quantile(G,0.025)
upper_b <- quantile(G,0.975)

CI <- c(lower_b,upper_b)
CI
```

```
##      2.5%     97.5%
## 0.1832921 0.5298789
```

The equal tail interval for 95% is 0.1832921and 0.5298789

2d Use the posterior draws in 2b) to compute a 95% Highest Posterior Density interval for G. Compare the two intervals in (c) and (d).

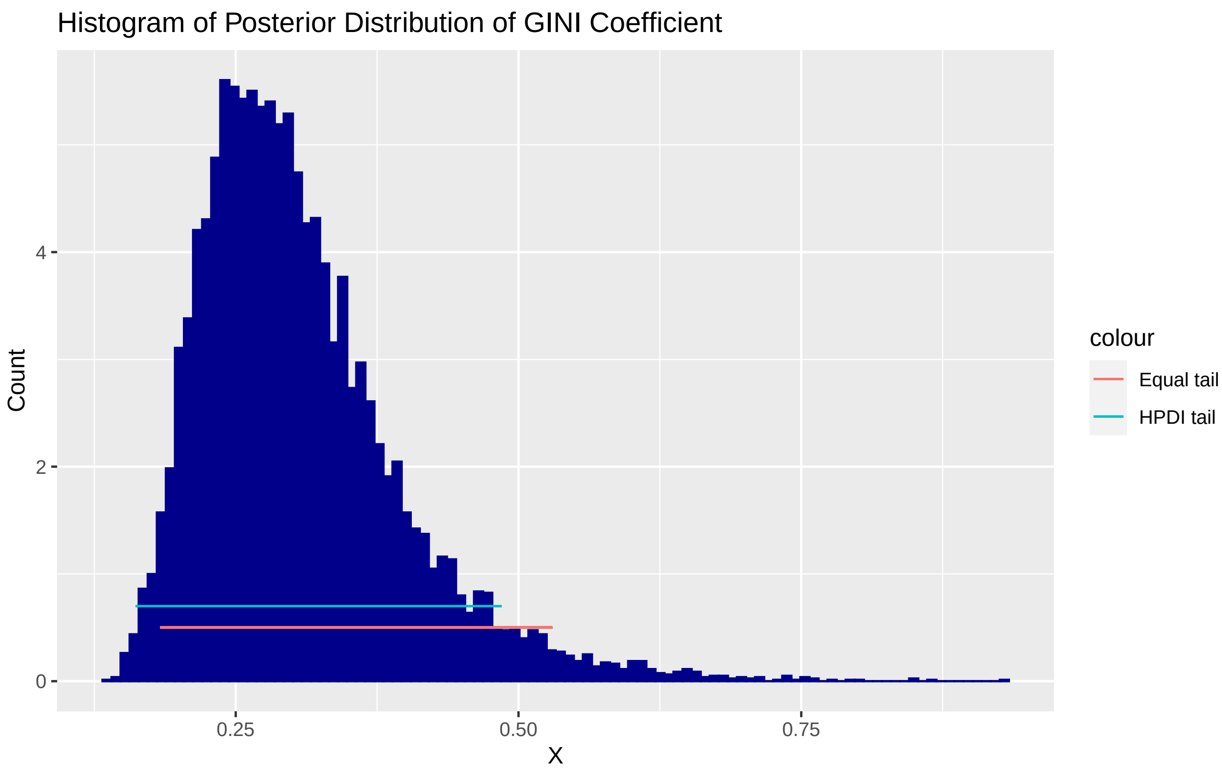
```
kdens_estimate <- density(G)

dens_df <- data.frame(x = kdens_estimate$x,y  = kdens_estimate$y)

# sort in descending order
ordered_indices <- order(dens_df$y,decreasing = TRUE)
```

```
ordered_dens_df <- dens_df[ordered_indices,]

# adding a row for cumulative sum of y's
ordered_dens_df$csum <- cumsum(ordered_dens_df$y)

#cut-off is 95% of the last value in the csum column.
cutoff <- 0.95* ordered_dens_df$csum[dim(ordered_dens_df)[1]]

#filtering for all values that are less than eq to cuttoff
HPdensity <- ordered_dens_df[ordered_dens_df$csum <= cutoff,]

# min and max to show the end points of the CI
HPDIntervals <- c(min(HPdensity$x),max(HPdensity$x))
```

The HPDI is 0.1616693 and 0.4850015

Comparing the two intervals

```
ggplot(data = G_df, aes(x = G)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "darkblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution of GINI Coefficient", x = "X", y = "Count") +
  geom_segment(aes(x = CI[1],y = 0.5,yend =  0.5,xend = CI[2],colour = 'Equal tail'))  +
  geom_segment(aes(x = HPDIntervals[1],y = 0.7,yend =  0.7,xend = HPDIntervals[2],colour = 'HPDI tail'))
```



We see that the HPDI intervals calculated are in line with the skew of the posterior distribution.

Question 3

3a) Derive the expression for what the posterior is proportional to

Since the likelihood $L(p(y|\mu,\kappa))$ has the below expression

$$Likelihood = \prod_{i=1}^{n} \frac{exp(\kappa * cos(y_i - \mu))}{2\pi I_0(\kappa)}$$

Also,κ~exponational($\lambda = 0.5$), the prior has the expression

$$p(\kappa) = \lambda * exp(-\lambda * \kappa)$$

The posterior is proportional to prior*liklihood, we obtain

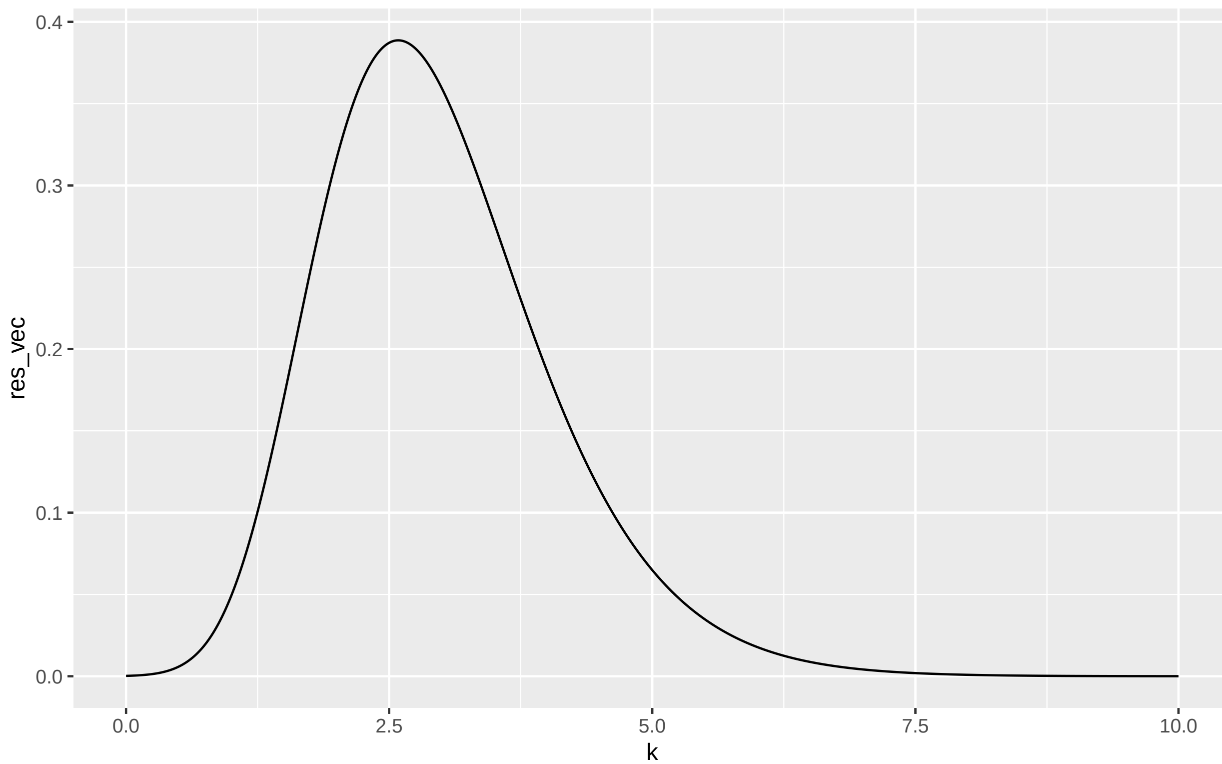
$$posterior \propto \frac{1}{2\pi I_0(\kappa)}^{n} * \lambda * exp[\kappa(\sum_{i=1}^{n} cos(y_i - \mu) - \lambda)]$$

To normalize the posterior distribution, we first integrate the existing posterior function(The upper and lower bound is set as we test kappa from 0~10). After that, we divide the value to existing posterior function and test if it will integrate to 1.

```
k <- seq(0,10,0.001)

posterior_func_before_normal <- function(k,data,lambda,mu){
  data <- c( -2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
  lambda <- 0.5
  mu <- 2.4
  n <- length(data)
  elem1 <- (1/(2*pi*besselI(k,nu=0)))**n
  elem2 <- sum(cos(data-mu))-lambda
  result <- lambda*elem1*exp(k*elem2)


  return (result)
}

integration_factor=integrate(posterior_func_before_normal, lower =0 , upper = 10)[[1]]

posterior_func_norm <- function(k){
  data <- c( -2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
  lambda <- 0.5
  mu <- 2.4
  n <- length(data)
  elem1 <- (1/(2*pi*besselI(k,nu=0)))**n
  elem2 <- sum(cos(data-mu))-lambda
  result <- lambda*elem1*exp(k*elem2)


  return (result/integration_factor)
}
testintegrate= integrate(posterior_func_norm, lower =0 , upper = 10)[[1]]
cat("The integration of normalized posterior distribution is ",testintegrate)
```

```
## The integration of normalized posterior distribution is  1
```

```
res_vec <- posterior_func_norm(k)
plotdf <- data.frame(k,res_vec)

ggplot(plotdf)+geom_line(aes(x=k,y=res_vec))
```



3b) Find the (approximate) posterior mode of k from the information in a)

To obtain the posterior mode, we can took the κ that produce maximum value of the distribution curve, which we obtain from the dataframe produce from Question 3a.

```
max_index <- which.max(plotdf$res_vec)
post_mode <- plotdf[max_index,1]

cat("The posterior mode of k is ",post_mode )
```

```
## The posterior mode of k is  2.586
```

Appendix

```r
library(ggplot2)
set.seed(12345)

#Question 1a
s <- 22
n <- 70
alpha_0 <- 8
beta_0 <- 8
f <- n-s

alpha <- alpha_0+s
beta <- beta_0+f
real_mean <- alpha/(alpha+beta)
real_sd <- sqrt(alpha*beta/((alpha+beta)**2*(alpha+beta+1)))

mean <- c()
sd <- c()

for(i in seq(0,10000)){
  nDraws <- rbeta(i,alpha,beta)
  mean <- append(mean,mean(nDraws))
  sd <- append(sd,sd(nDraws))
}

plot(mean,type = 'l')
lines(x=seq(0,10000),y=rep(real_mean,10001),col="red")

plot(sd,type = 'l')
lines(x=seq(0,10000),y=rep(real_sd,10001),col="red")

#Question 1b
real_prob <- pbeta(0.3,alpha,beta,lower.tail = FALSE)
nDraws <- rbeta(10000,alpha,beta)
sample_prob <- sum(nDraws>0.3)/length(nDraws)

cat("The difference bettween real and simulation probability are",sample_prob-real_prob,"which is very 

#Question 1c
hist(nDraws/(1-nDraws),breaks = 100)
plot(density(nDraws/(1-nDraws)))

#-----------------------------------

#Question 2a
obs <- c(33,24,48,32,55,74,23,17)
n <- length(obs)-1

calculate_tau <- function(mu)
{
  res <- (sum((log(obs) - mu)^2))/n
  return(res)
```

```r
}

tau_2 <- calculate_tau(mu = 3.6)

X <- rchisq(10000,df = n)
xs <- (n*tau_2)/X

xs_df <- as.data.frame(xs)

ggplot(data = xs_df, aes(x = xs)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "lightblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution", x = "X", y = "Count")

#Question 2b
phi_z <- sqrt(xs)/sqrt(2)
G <- (2 * pnorm(phi_z,mean = 0,sd = 1)) -1

G_df <- as.data.frame(G)

ggplot(data = G_df, aes(x = G)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "lightblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution of GINI Coefficient", x = "X", y = "Count")

#Question 2c
lower_b <- quantile(G,0.025)
upper_b <- quantile(G,0.975)

CI <- c(lower_b,upper_b)
CI

#Question 2d
kdens_estimate <- density(G)

dens_df <- data.frame(x = kdens_estimate$x,y  = kdens_estimate$y)

ordered_indices <- order(dens_df$y,decreasing = TRUE)

ordered_dens_df <- dens_df[ordered_indices,]

ordered_dens_df$csum <- cumsum(ordered_dens_df$y)

cutoff <- 0.95* ordered_dens_df$csum[dim(ordered_dens_df)[1]]

HPdensity <- ordered_dens_df[ordered_dens_df$csum <= cutoff,]

HPDIntervals <- c(min(HPdensity$x),max(HPdensity$x))

ggplot(data = G_df, aes(x = G)) +
  geom_histogram(aes(y = ..density..), color = "darkblue", fill = "darkblue",bins = 100) +
  labs(title = "Histogram of Posterior Distribution of GINI Coefficient", x = "X", y = "Count") +
  geom_segment(aes(x = CI[1],y = 0.5,yend =  0.5,xend = CI[2],colour = 'Equal tail'))  +
  geom_segment(aes(x = HPDIntervals[1],y = 0.7,yend =  0.7,xend = HPDIntervals[2],colour = 'HPDI tail'))
```

```r
#--------------------------------
#Question 3a
posterior_func_before_normal <- function(k,data,lambda,mu){
  data <- c( -2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
  lambda <- 0.5
  mu <- 2.4
  n <- length(data)
  elem1 <- (1/(2*pi*besselI(k,nu=0)))**n
  elem2 <- sum(cos(data-mu))-lambda
  result <- lambda*elem1*exp(k*elem2)


  return (result)
}

k <- seq(0,10,0.001)
integration_factor=integrate(posterior_func_before_normal, lower =0 , upper = 10)[[1]]

posterior_func_norm <- function(k){
  data <- c( -2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
  lambda <- 0.5
  mu <- 2.4
  n <- length(data)
  elem1 <- (1/(2*pi*besselI(k,nu=0)))**n
  elem2 <- sum(cos(data-mu))-lambda
  result <- lambda*elem1*exp(k*elem2)


  return (result/integration_factor)
}
testintegrate= integrate(posterior_func_norm, lower =0 , upper = 10)[[1]]
cat("The integration of normalized posterior distribution is ",testintegrate)
res_vec <- posterior_func_norm(k)
plotdf <- data.frame(k,res_vec)

ggplot(plotdf)+geom_line(aes(x=k,y=res_vec))

#Question 3b
max_index <- which.max(plotdf$res_vec)
post_mode <- plotdf[max_index,1]

cat("The posterior mode of k is ",post_mode )
```