Computer Lab 1

Ravinder Reddy Atla

4/6/2021

Daniel Bernoulli

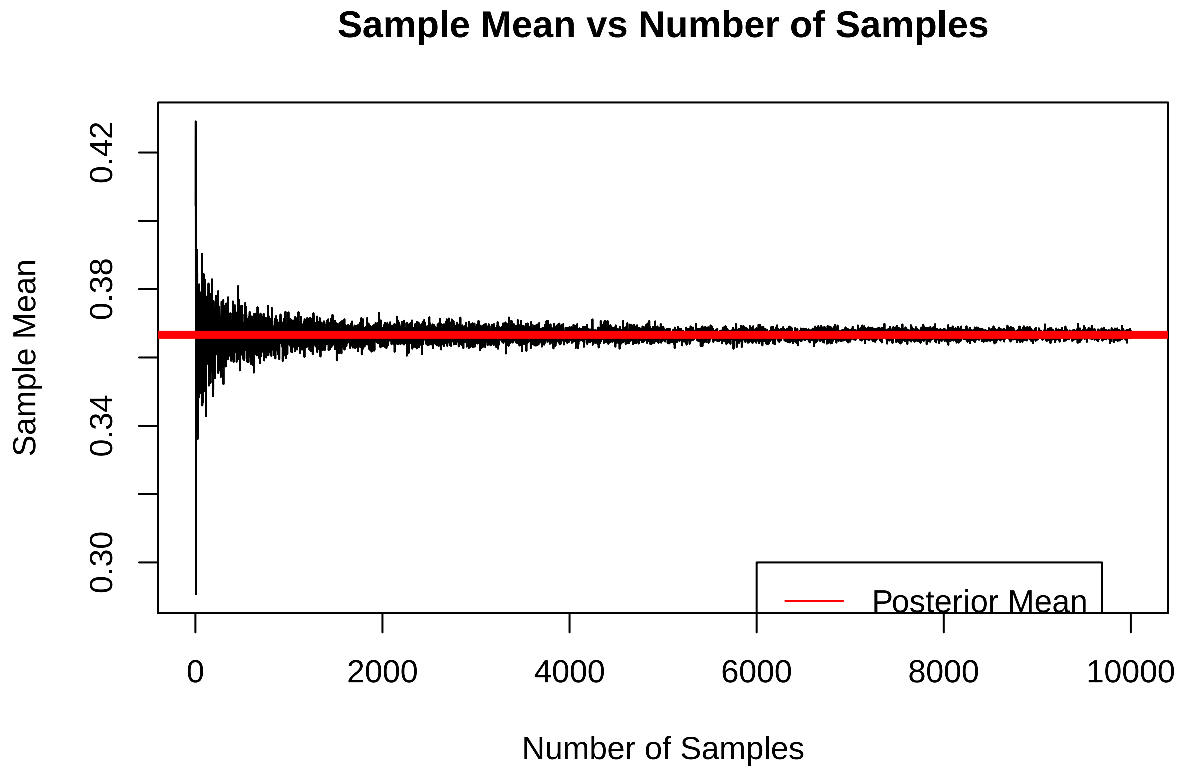
```
# Data distribution parameters
n <- 24
s <- 8
f <- n-s
# Prior distribution parameters
alpha_0 <- 3
beta_0 <- 3
```

a. Generating random numbers from posterior distribution and verifying graphically

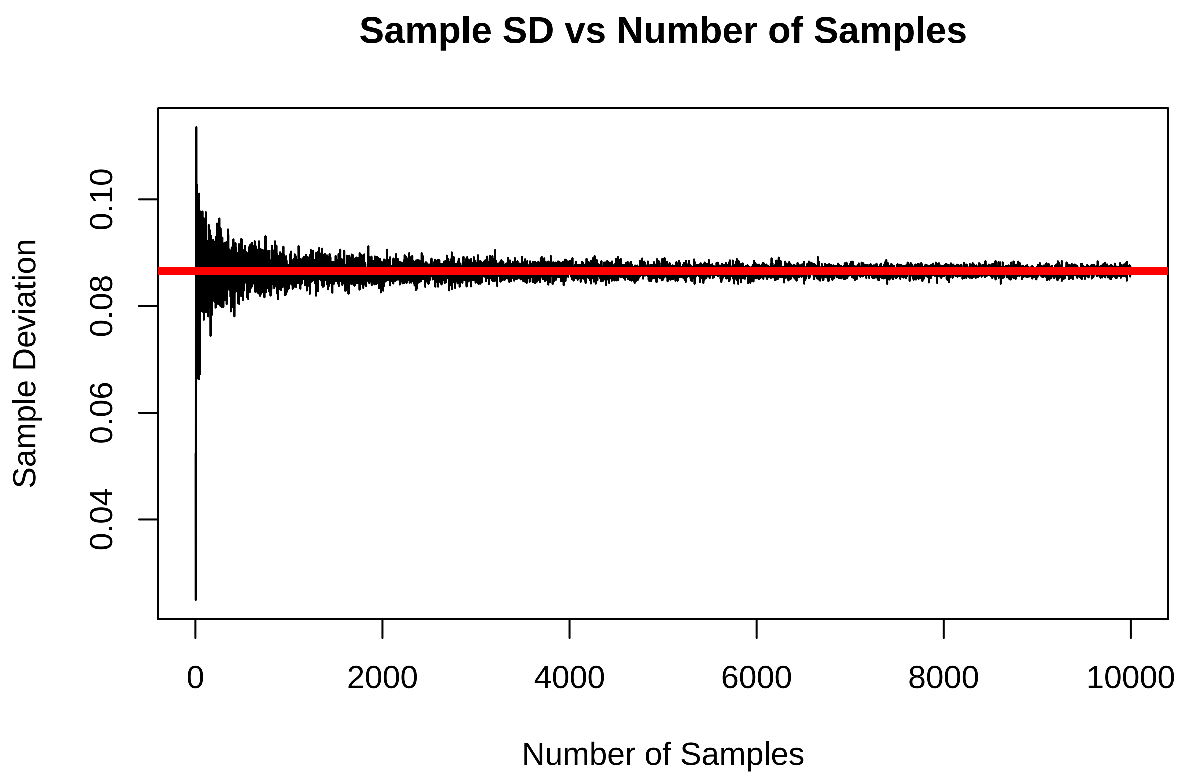
```
alpha_n <- alpha_0 + s
beta_n <- beta_0 + f

posterior_mean <- alpha_n/(alpha_n + beta_n)
posterior_variance <- (alpha_n * beta_n)/((alpha_n + beta_n)^2 *
                                            (alpha_n + beta_n + 1))
posterior_sd <- sqrt(posterior_variance)

posteriorDrawParams <- function(num_draws,alpha_n,beta_n){
  #set.seed(1234)
  theta_n <- rbeta(num_draws, shape1 = alpha_n, shape2 = beta_n)
  sample_mean <- mean(theta_n)
  sample_sd <- sd(theta_n)
  return(c(sample_mean,sample_sd))
}
```

```
num_samples <- 1
mu_vector <- c()
sample_mu <- 0.3
sample_dev <- 1
#plot(num_samples,sample_mu)

#while(abs(posterior_mean - sample_mu) > 1e-7){
while(num_samples < 10000){
  num_samples <- num_samples + 1
  sample_mu <- posteriorDrawParams(num_samples,alpha_n,beta_n)[1]
  mu_vector <- c(mu_vector,sample_mu)
}
plot(c(2:num_samples),mu_vector,'l',
     xlab = 'Number of Samples',ylab = 'Sample Mean',
```

```
     main = 'Sample Mean vs Number of Samples')
legend(6000,0.30,legend = c('Posterior Mean'),col = c('red'),lty=1)
abline(h = posterior_mean,lwd=4,col = 'red')
```




```
num_samples <- 1
sd_vector <- c()
sample_dev <- 1

while(num_samples < 10000){
  num_samples <- num_samples + 1
  sample_dev <- posteriorDrawParams(num_samples,alpha_n,beta_n)[2]
  sd_vector <- c(sd_vector,sample_dev)
}
plot(c(2:num_samples),sd_vector,'l',
     xlab = 'Number of Samples',ylab = 'Sample Deviation',
     main = 'Sample SD vs Number of Samples')
legend(500,0.14,legend = c('Posterior SD'),col = c('red'),lty=1)
abline(h = posterior_sd,lwd=4,col = 'red')
```



b.

```
posteriorDraw <- function(num_draws,alpha_n,beta_n){
  set.seed(1234)
  theta_n <- rbeta(num_draws, shape1 = alpha_n, shape2 = beta_n)
  sample_mean <- mean(theta_n)
  sample_sd <- sd(theta_n)
  return(theta_n)
}
```

```
# P(theta > 0.4|y) from sampled data
simulated_data <- posteriorDraw(num_draws = 10000,alpha_n,beta_n)
posterior_prob <- length(simulated_data[simulated_data > 0.4])/length(simulated_data)
print(posterior_prob)
```

```
## [1] 0.3473
```

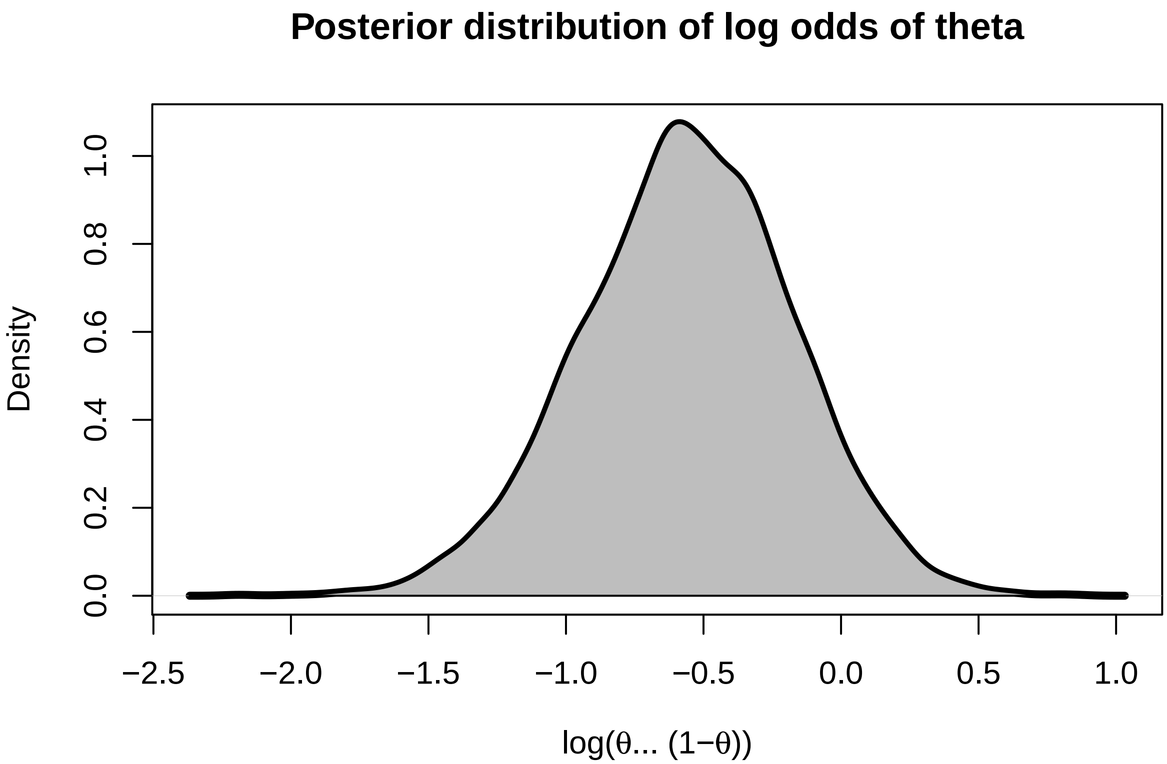
```
# Actual value of P(theta > 0.4|y)
x <- 0.4
actual_prob <- pbeta(x,alpha_n,beta_n,lower.tail = FALSE)
print(actual_prob)
```

```
## [1] 0.3426654
```

c.

```
theta_post <- posteriorDraw(num_draws = 10000,alpha_n,beta_n)
phi <- log(theta_post) - log(1 - theta_post)
plot(density(phi),lwd=4,
     main = 'Posterior distribution of log odds of theta',
```

```
     xlab = expression(paste('log(',theta, '/ (1-',theta,'))')))
polygon(density(phi),col = 'grey')
```



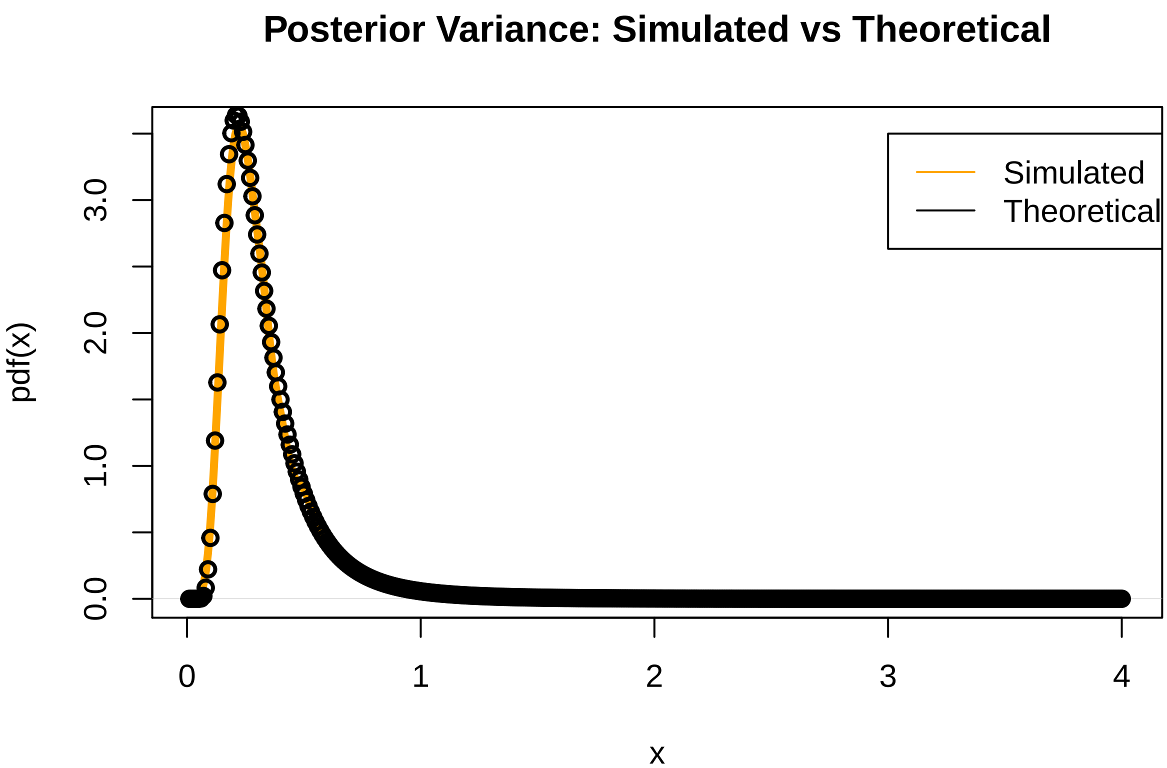
2. Log-normal distribution and Gini coefficient

a.

```
y <- c(38, 20, 49, 58, 31, 70, 18, 56, 25, 78)
mu <- 3.8
n <- 10
tau_sq <- sum((log(y)-mu)^2)/n

rinvchisquare <- function(num_draws, n, tau_sq){
  set.seed(1234)
  x <- rchisq(num_draws,df = n-1)
  x_inv <- ((n-1)*tau_sq)/x
  return(x_inv)
}

# Scaled Inverse chi square distribution
pdfinvchisquare <- function(x,n,tau_sq){
  res <- (((tau_sq*(n-1))/2)^(n-1)/2 * exp(-(tau_sq*(n-1))/(2*x)))/((x^(1+(n-1)/2)) * gamma((n-1)/2))
  return(res)
}
```

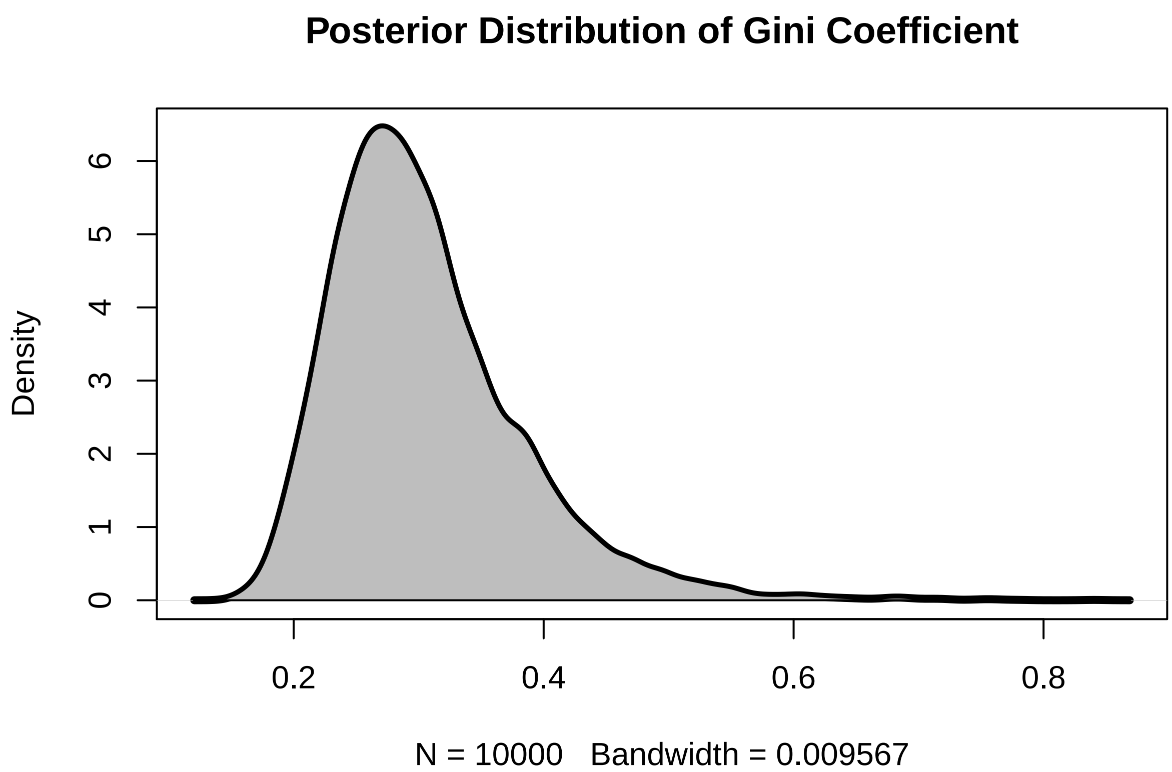
```
post_sample <- rinvchisquare(10000, n, tau_sq)
plot(density(post_sample),
     col = 'orange',lwd=4,
     main = 'Posterior Variance: Simulated vs Theoretical',
```

```
     xlab = 'x', ylab = 'pdf(x)')

x <- seq(0,4,by=0.01)
org_val <- mapply(pdfinvchisquare,x,rep(n,length(x)),rep(tau_sq,length(x)))
points(x,org_val,col = 'black',lwd = 2)
legend(3, 3.5,
       legend = c('Simulated','Theoretical'),
       col = c('orange','black'),
       lty=c(1,1))
```



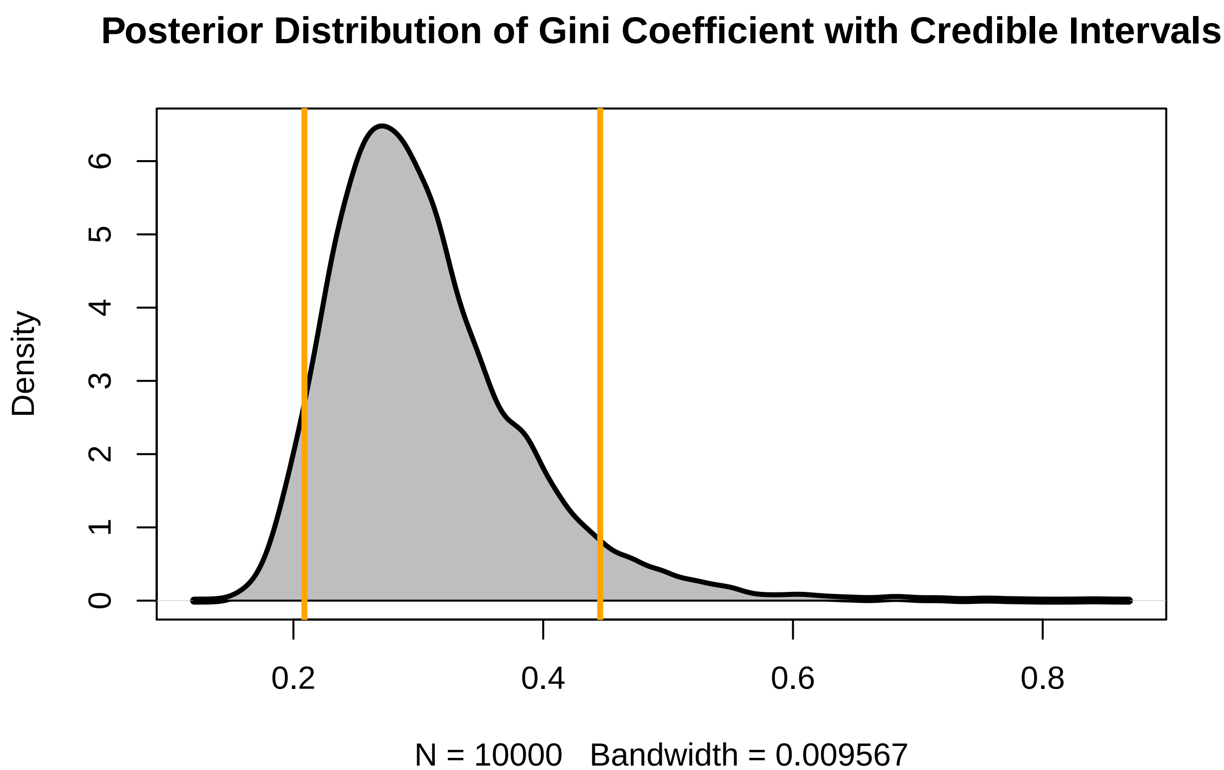
b.

```
phi <- pnorm(sqrt(post_sample/2), mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
gini_coeff <- 2*phi -  1
plot(density(gini_coeff),lwd = 4,
     main = 'Posterior Distribution of Gini Coefficient')
polygon(density(gini_coeff),col = 'grey')
```

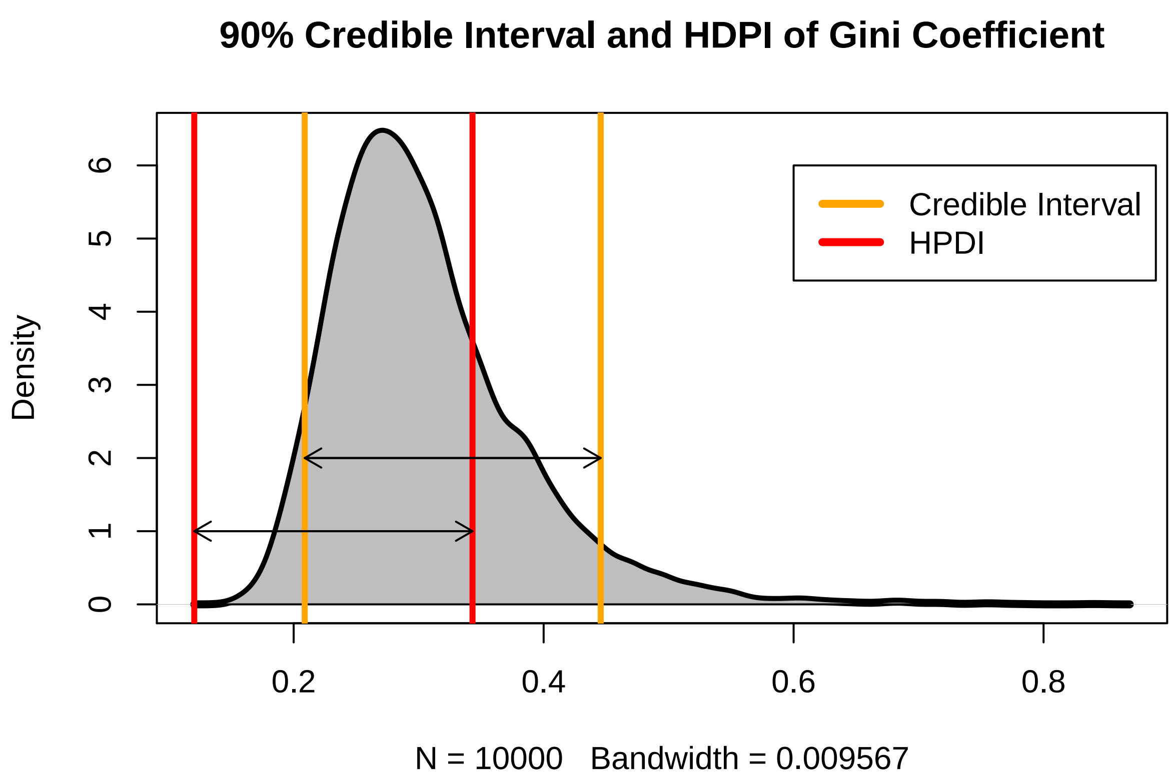


c.

```
# Credible Interval
CI <- c(0.05,0.95)
lower_interval <- quantile(gini_coeff,CI[1])
upper_interval <- quantile(gini_coeff, CI[2])

plot(density(gini_coeff),lwd = 4,
     main = 'Posterior Distribution of Gini Coefficient with Credible Intervals')
polygon(density(gini_coeff),col = 'grey')
abline(v = lower_interval,col = 'orange', lwd = 3)
abline(v = upper_interval,col = 'orange', lwd = 3)
legend(0.6, 30,
       legend = c('Credible Interval'),
       col = c('Orange'),
       lty=c(1,1))
```




```r
kernel_gini_coeff <- density(gini_coeff)

# Highest Posterior Density Interval (HDPI)

sorted_kgc <- sort(kernel_gini_coeff$y, decreasing = TRUE)
cumm_kgc <- cumsum(sorted_kgc)
hpd <- cumm_kgc[length(cumm_kgc)]*0.9
hpdi <- range(kernel_gini_coeff$x[which(cumm_kgc<hpd)])

plot(density(gini_coeff),lwd = 4,
     main = '90% Credible Interval and HDPI of Gini Coefficient')
polygon(density(gini_coeff),col = 'grey')
abline(v = lower_interval,col = 'orange', lwd = 3)
abline(v = upper_interval,col = 'orange', lwd = 3)
abline(v = hpdi[1],col = 'red',lwd = 3)
abline(v = hpdi[2],col = 'red',lwd = 3)
arrows(lower_interval,2,upper_interval,2,length = 0.1,col = 'black')
arrows(upper_interval,2,lower_interval,2,length = 0.1,col = 'black')
arrows(hpdi[1],1,hpdi[2],1,length = 0.1,col = 'black')
arrows(hpdi[2],1,hpdi[1],1,length = 0.1,col = 'black')

legend(0.6, 6,
       legend = c('Credible Interval','HPDI'),
       col = c('Orange','Red'),
       lty=c(1,1),lwd = 4)
```



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

```
likelihood <- function(y, mu, kappa){
  return_val <- exp(kappa * sum(cos(y - mu))) / ((2*pi*besselI(kappa,nu=0))^10)
  return(return_val)
}

prior_kap <- function(kapa,lambda = 1){
  return_val <- lambda*exp(-1*lambda*kapa)
  return(return_val)
}

posterior_kap <- function(y, mu, kappa){
  return_val <- likelihood(y, mu, kappa) * prior_kap(kappa)
  return(return_val)
}
```

```
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
mu <- 2.39
kap <- seq(from = 0, to = 10, by = 0.01)
#posterior_kappa <- rep(0,length(kap))
#for(k in 1:length(kap)){
#  posterior_kappa[k] <- (posterior_kap(y,mu,kap[k]))
#}
posterior_kappa <- (posterior_kap(y, mu, kappa = kap))
plot(kap,prior_kap(kap),'l',
     xlab = expression(paste(kappa)),
     ylab = expression(paste('p(',kappa,'|y)')),col = 'red',lwd = 2)
```

```r
lines(kap,posterior_kappa, col = 'blue','l',lwd = 2)
abline(v = max(posterior_kappa),lwd = 2)

legend(2.5, 1,
       legend = c('Prior', 'Posterior Mean', 'Approximate Mode of kappa'),
       col = c('Red','Blue','Black'),
       lty=c(1,1),lwd = c(2,2,2))
```