Bayesian_Lab1

Olayemi Morrison(olamo208) Greeshma Jeev Koothuparambil (greko370)

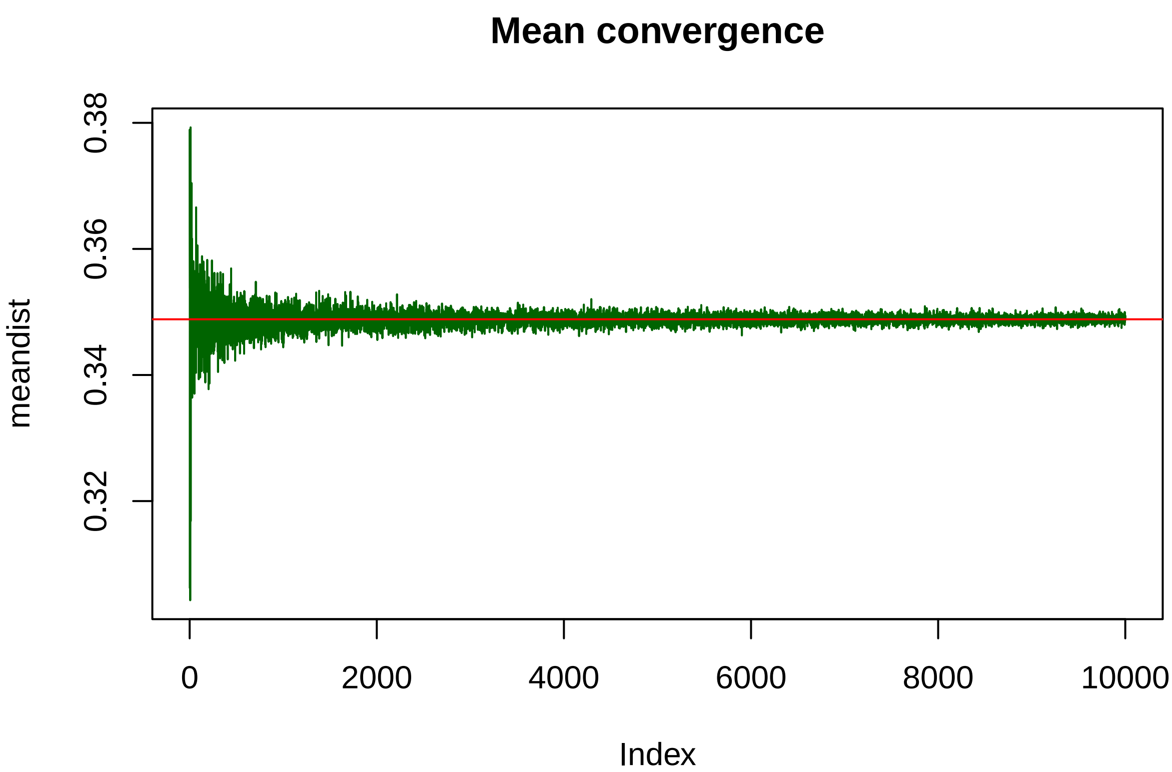
2024-04-15

1a) Draw 10000 random values (nDraws = 10000) from the posterior θ |y Beta(α 0 +s, β 0 + f), where y = (y1 , . . . , yn), and verify graphically 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.a

#Trail information
n= 70
s= 22
f= n-s

#Beta Prior
a= 8
b=8

newalpha <- a+s
newbeta <- b+f

truemean <- newalpha/(newalpha+newbeta)
truesd <- sqrt((newalpha*newbeta)/((newalpha+newbeta+1)*(newalpha+newbeta)^2))
meandist <- c()
sddist <- c()


nDraws = 10000

for(i in 1:nDraws ){

  posterior = rbeta(i, a+s, b+f)
  meandist[i] <- mean(posterior)
  sddist[i] <- sd(posterior)
}
```

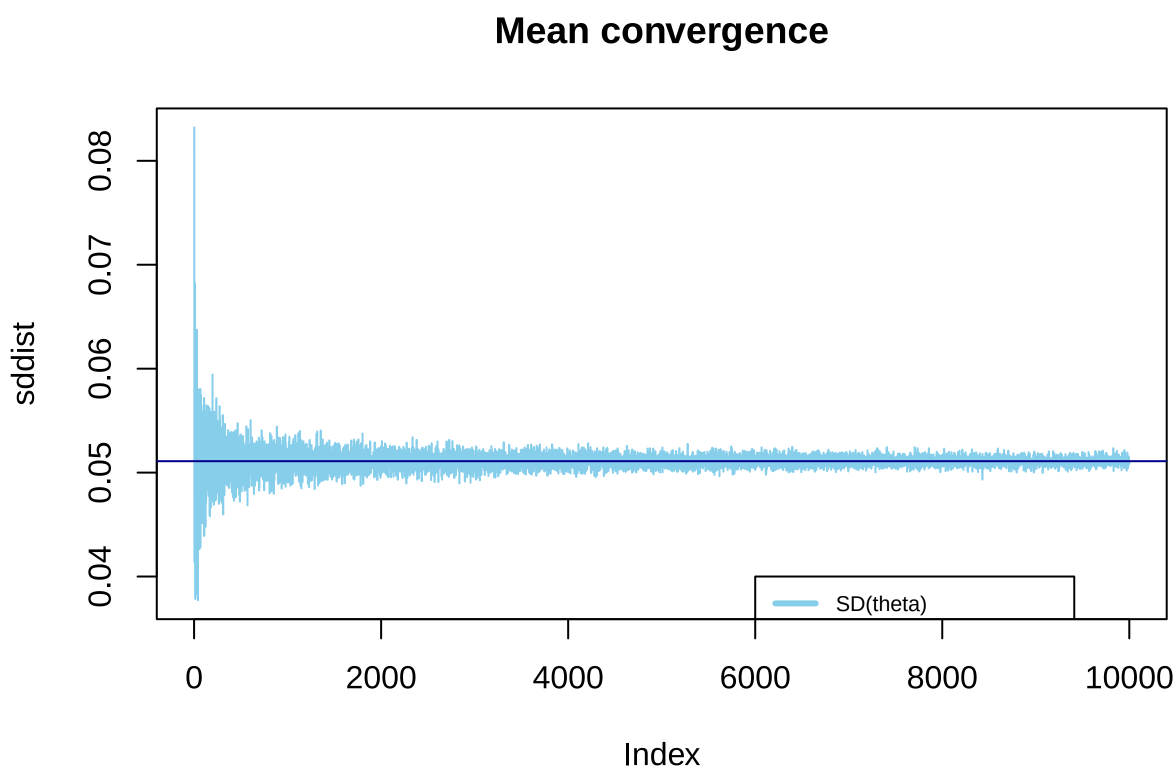
The plot on mean convergence trend is given below:

```
plot(meandist, type = 'l', col = "darkgreen", main ="Mean convergence")
abline(h= truemean, col="red")
legend(x = 6000 ,y = 0.3,
       legend = c("E(theta)","True Mean"),
       col = c("darkgreen","red"), lwd = c(3,3,3), cex = 0.7)
```



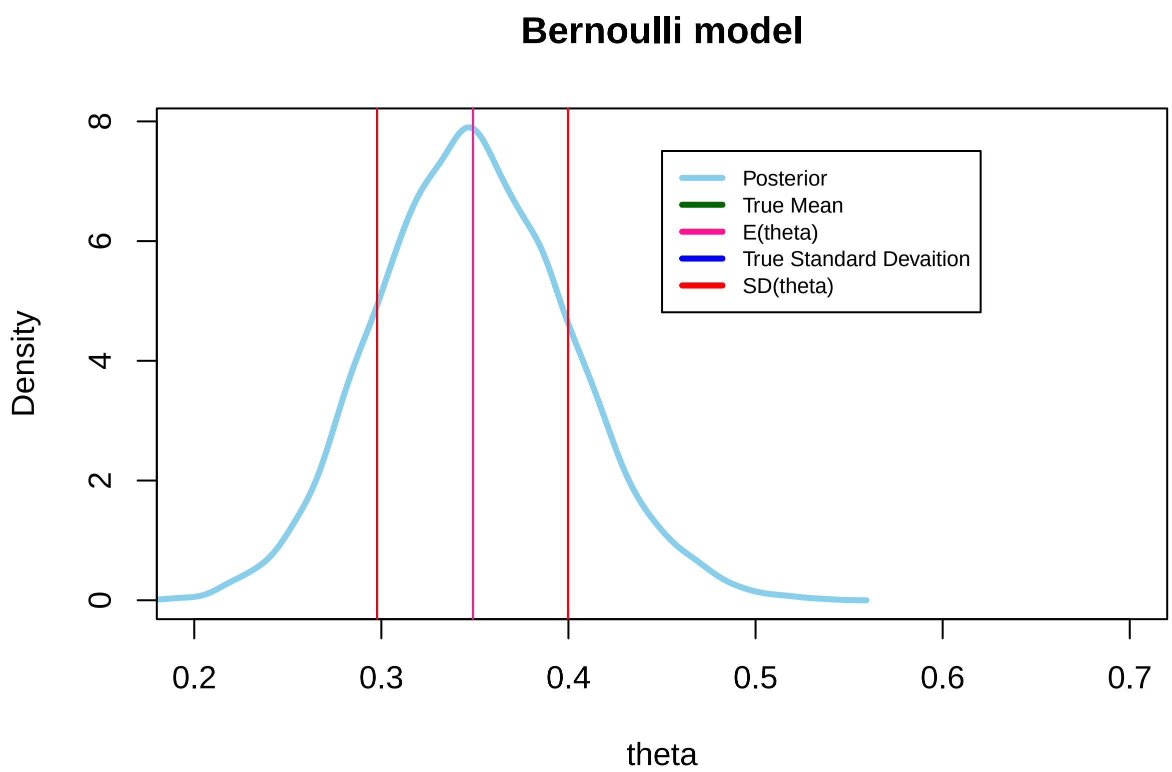
The plot on standard deviation convergence is as below:

```
plot(sddist, type = 'l', col = "skyblue", main ="Mean convergence")
abline(h= truesd, col="darkblue")
legend(x = 6000 ,y = 0.04,
       legend = c("SD(theta)","True Standard Deviation"),
       col = c("skyblue","darkblue"), lwd = c(3,3,3), cex = 0.7)
```



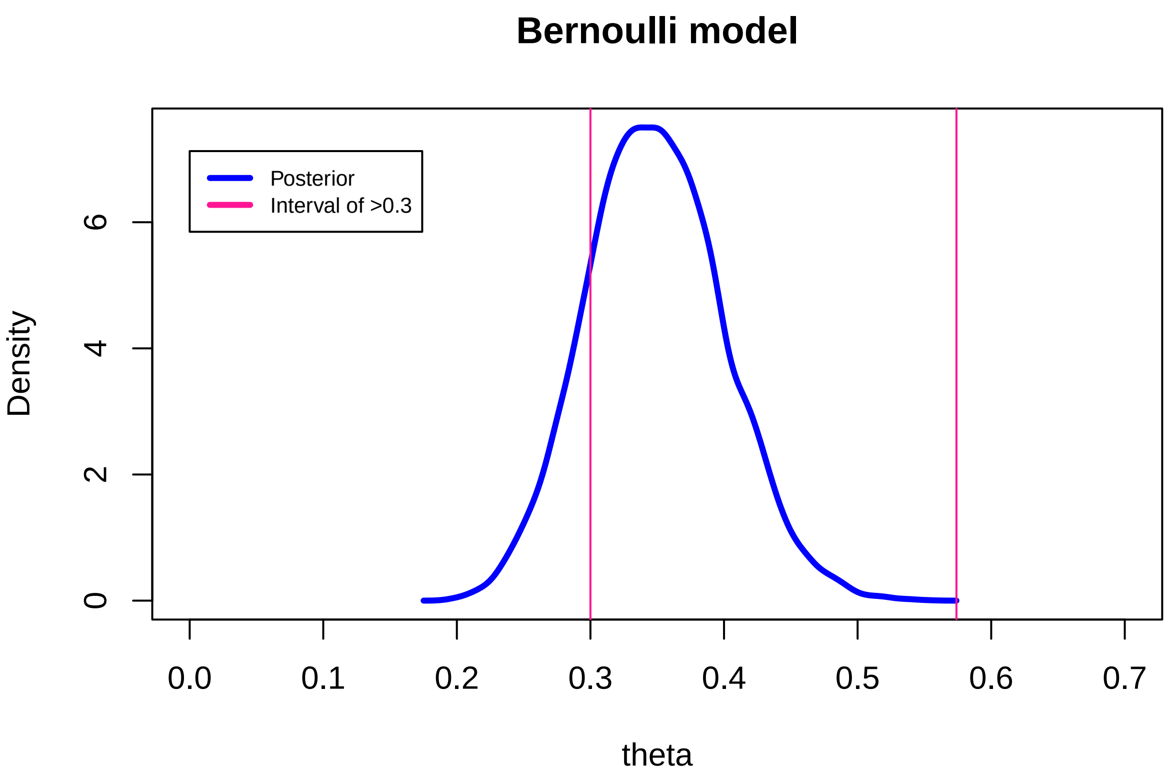
Generalised version of the density graph:

```
posterior = rbeta(nDraws, a+s, b+f)
 plot( density(posterior)$x, density(posterior)$y , type = 'l', lwd = 3, col = "skyblue",
       xlim = c(0.2,.7),xlab = "theta", ylab = 'Density', main = 'Bernoulli model')
 abline(v= truemean, lwd= 1, col = "darkgreen")
 abline(v= meandist[nDraws], lwd= 1, col = "deeppink")
 abline(v= c(truemean - truesd, truemean+truesd), lwd=1 , col= "blue")
 abline(v= c(meandist[nDraws]-sddist[nDraws], meandist[nDraws]+sddist[nDraws]),
        lwd= 1, col = "red")
 legend(x = 0.45, y = max(density(posterior)$y)*0.95,
        legend = c("Posterior", "True Mean", "E(theta)", "True Standard Devaition", "SD(theta)"),
        col = c("skyblue","darkgreen","deeppink","blue","red"), lwd = c(3,3,3), cex = 0.7)
```



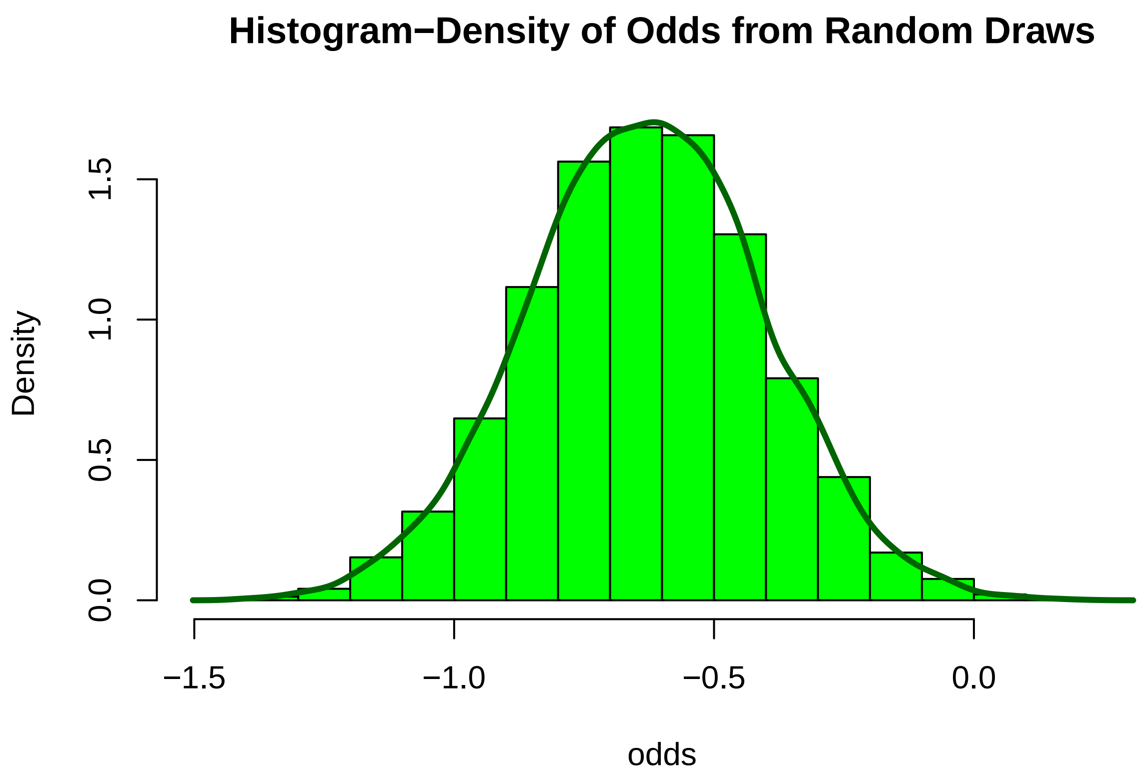
1b) Draw 10000 random values from the posterior to compute the posterior probability Pr($\theta > 0.3|$y) and compare with the exact value from the Beta posterior

```
#1.b

posterior <- rbeta(nDraws, a+s, b+f)
sampleProbabilityofpoint3 <- sum(posterior>0.3)/nDraws
ActualProbabilityofpoint3 <- 1- pbeta(0.3,a+s, b+f)

plot( density(posterior)$x, density(posterior)$y , type = 'l', lwd = 3, col = "blue",
      xlim = c(0,.7), xlab = "theta", ylab = 'Density', main = 'Bernoulli model')
abline(v= c(0.3,max(density(posterior)$x)), lwd= 1, col = "deeppink")
legend(x = 0, y = max(density(posterior)$y)*0.95,
       legend = c("Posterior", "Interval of >0.3"),
       col = c("blue","deeppink"), lwd = c(3,3,3), cex = 0.7)
```



1c) Draw 10000 random values from the posterior of the odds $\phi = \theta/1 - \theta$ the previous random draws from the Beta posterior for θ and plot the posterior distribution of ϕ.

```
#1.c

odds = log(posterior/(1- posterior))
hist(odds, prob=T, col = "green",main= "Histogram-Density of Odds from Random Draws")
lines(density(odds), lwd=3, col="darkgreen")
```

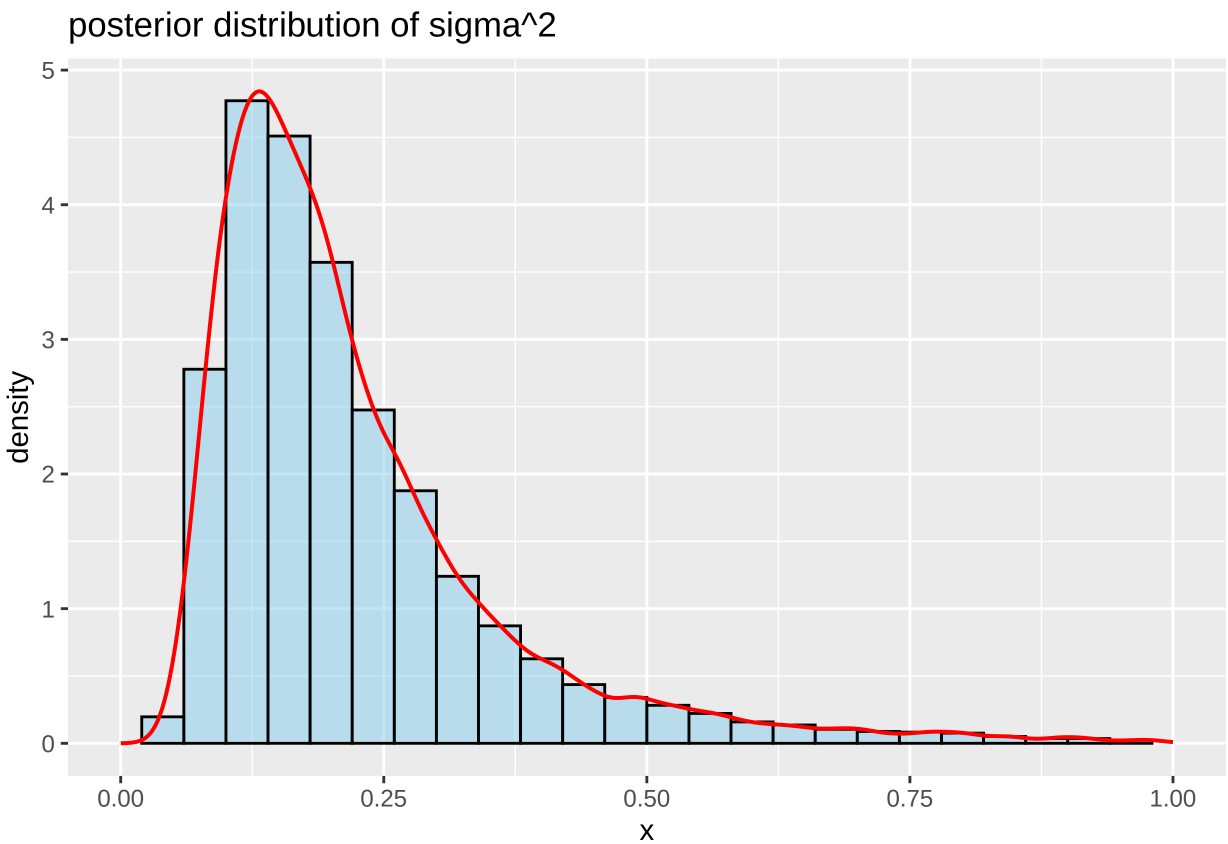


2a) Draw 10000 random values from the posterior of $\sigma\hat{}2$ by assuming μ=3.6 and plot the posterior distribution.

The code for the posterior distribution is as shown below:

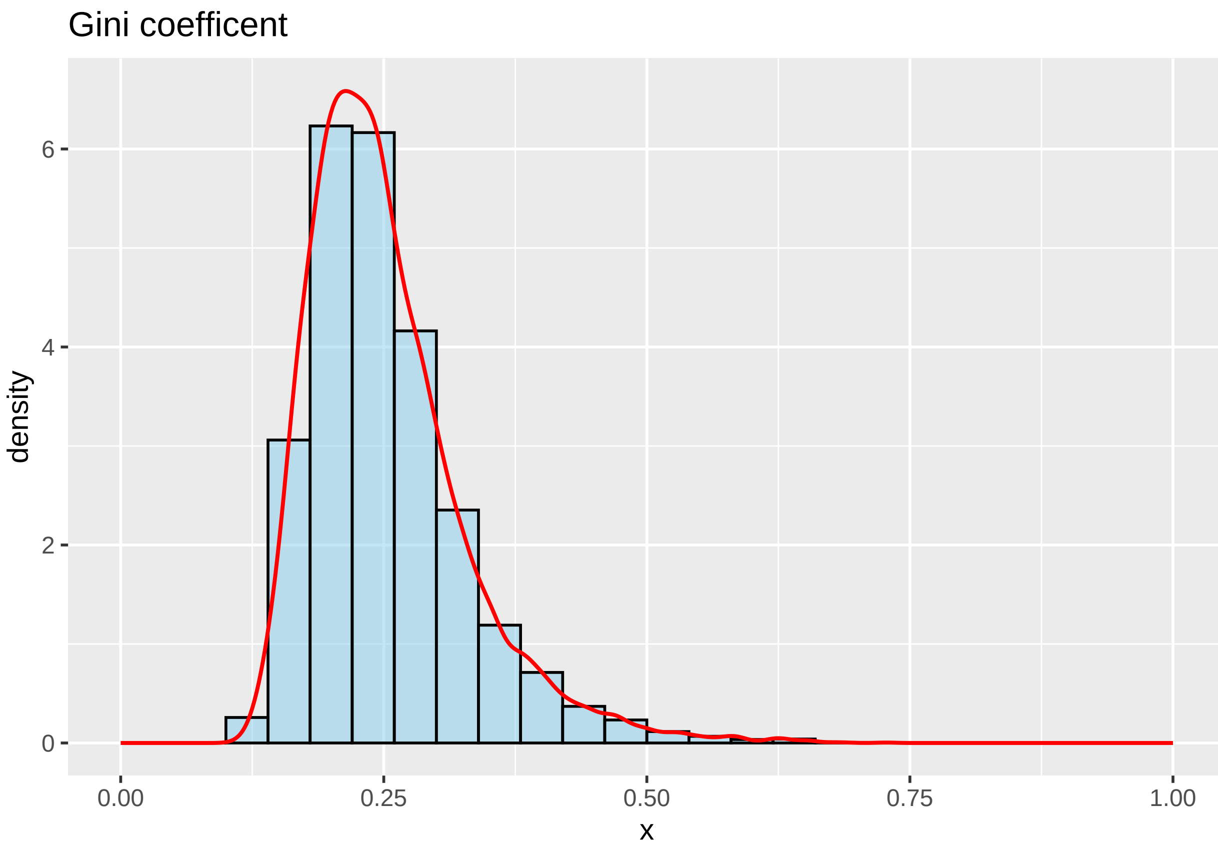
```
library(asbio)
library(ggplot2)
library(dplyr)
library(coda)

y= c(33, 24, 48, 32, 55, 74, 23)
val = 10000
mu = 3.6
n = length(y)

tau_sq <- function(y, mu){

  t2 = sum((log(y)-mu)^2)/n
  return(t2)
}

tau2 <- tau_sq(y, mu)

posterior1 <- rinvchisq(val, n, tau2)
dfpost1 <- data.frame(x=posterior1)

ggplot(dfpost1,aes(x))+
  geom_histogram(aes(y=after_stat(density)), binwidth = 0.04,
```

```
                 fill="skyblue", color="black", alpha=0.5)+
  geom_density(color="red", linewidth=0.7)+
  scale_x_continuous(limits = c(0,1))+
  labs(title = "posterior distribution of sigma^2")
```



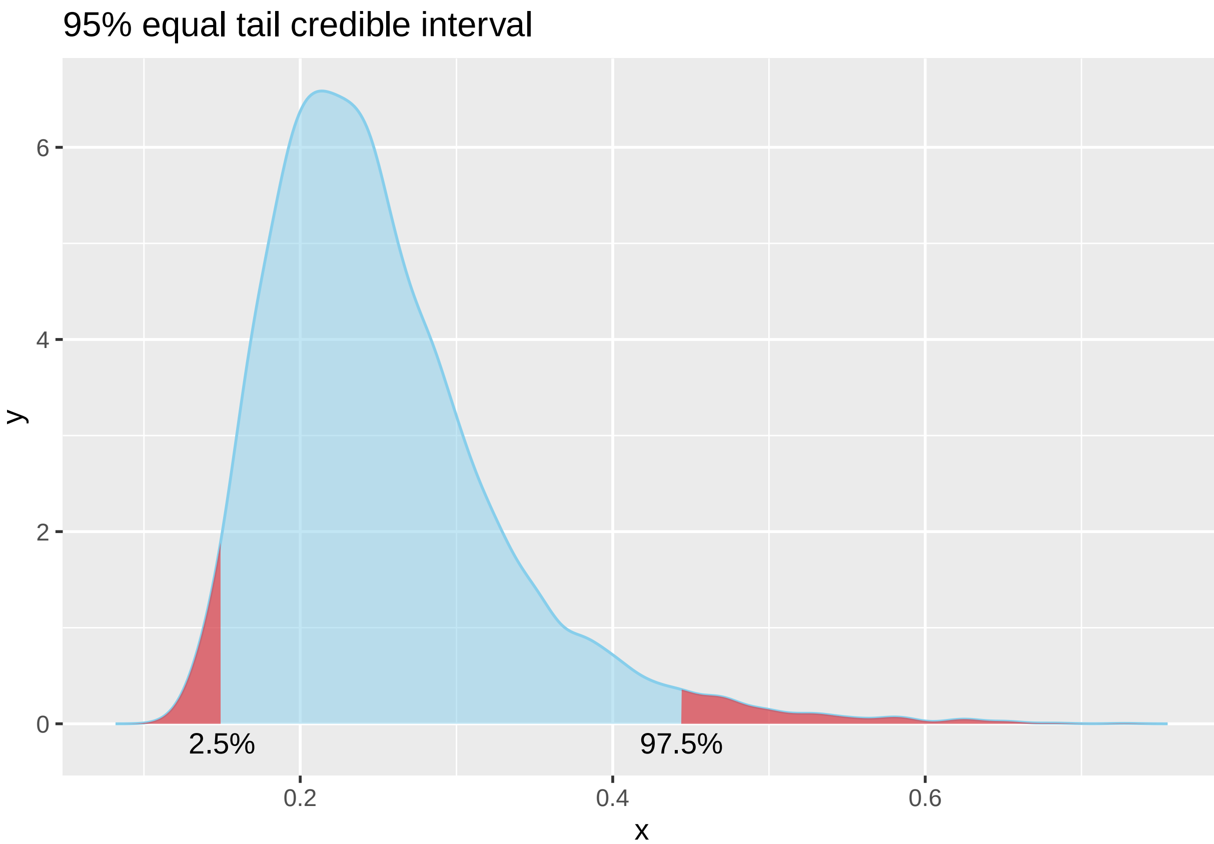
2b) Compute the posterior distribution of the Gini coefficient

```
gini <- 2*pnorm(sqrt(posterior1)/sqrt(2))-1
dfgini <- data.frame(x=gini)
ggplot(dfgini, aes(x))+
  geom_histogram(aes(y=after_stat(density)), binwidth = 0.04, fill="skyblue", color="black",
                 alpha=0.5)+
  geom_density(color="red", linewidth=0.7)+
  scale_x_continuous(limits = c(0,1))+
  labs(title = "Gini coefficent")
```



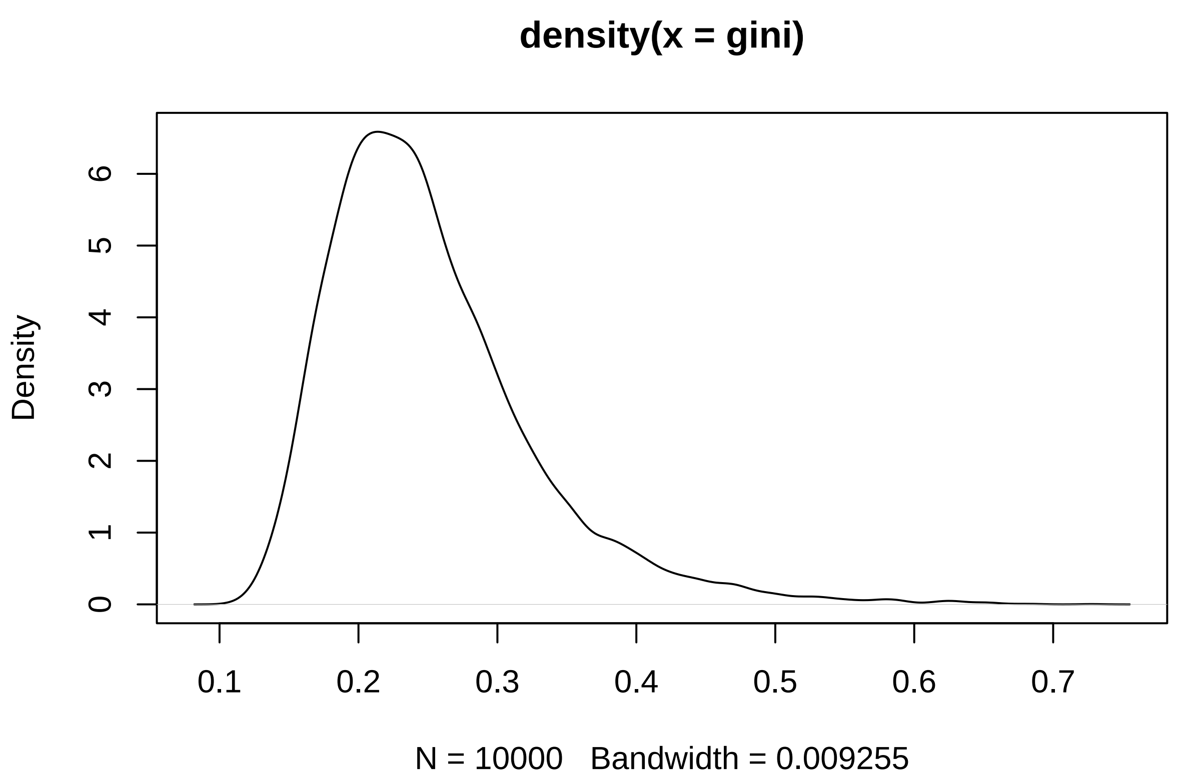
2c) Use the posterior draws from b) to compute 95% equal tail credible interval for G.

```
equaltint <- quantile(gini,c(0.025,0.975))
df <- with(density(dfgini$x), data.frame(x,y))
df1<-df%>%filter(x<equaltint[1])
df2<-df%>%filter(x>equaltint[2])
ggplot()+
  geom_density(data=dfgini,aes(x=x),color='skyblue',fill="skyblue",alpha=0.5)+
  geom_area(data=df1,aes(x=x,y=y),fill="red",alpha=0.5)+
  geom_area(data=df2,aes(x=x,y=y),fill="red",alpha=0.5)+
  annotate(geom='text',x=equaltint[1],y=-0.2,label='2.5%')+
  annotate(geom='text',x=equaltint[2],y=-0.2,label='97.5%')+
  labs(title="95% equal tail credible interval")
```



2d) Use the posterior draws from b) to compute 95% Highest Posterior Density interval for G.

```
dens_gini=density(gini)
plot(dens_gini)
```




```
sortdens_gini=sort(dens_gini$y,decreasing = TRUE)
sumdens_gini=sum(dens_gini$y)
cum_sum=0
i=1
while(cum_sum<0.95){
  cum_sum=cum_sum+(sortdens_gini[i]/sumdens_gini)
  i=i+1
}
cat('The highest density in the HPDI:',sortdens_gini[i])
```

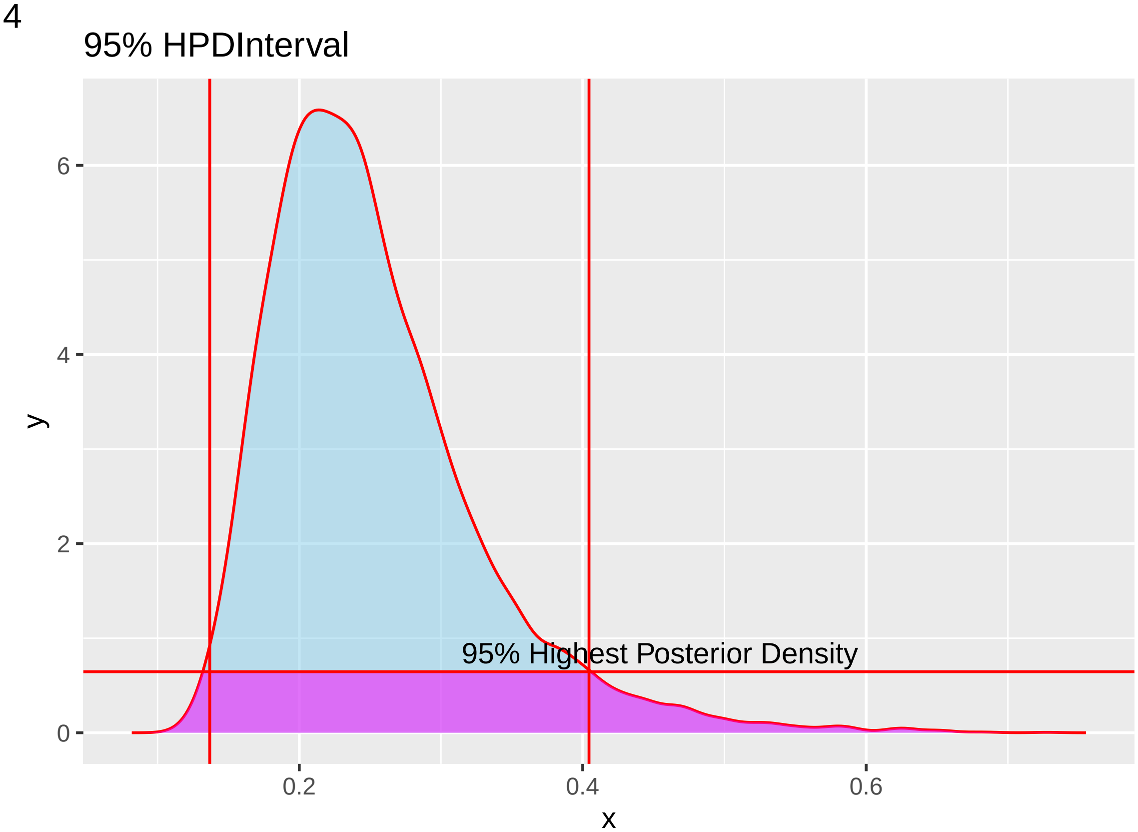
```
## The highest density in the HPDI: 0.645399
```

```
hpdi=HPDinterval(as.mcmc(gini),prob=0.95)
hpdi
```

```
##          lower     upper
## var1 0.1368328 0.4044398
## attr(,"Probability")
## [1] 0.95
```

```
dat_int=dens_gini$x[dens_gini$y<=sortdens_gini[i]]
dat=with(density(dfgini$x),data.frame(x,y))
dat1=dat%>%filter(x%in%dat_int)
ggplot()+
  geom_density(data=dfgini,aes(x=x),color='red',fill="skyblue",alpha=0.5)+
  geom_area(data=dat1,aes(x=x,y=y),fill="magenta",alpha=0.5)+
  labs(title="95% HPDInterval",tag='Fig 2.4')+
  geom_vline(xintercept = hpdi[1],color='red')+
  geom_vline(xintercept = hpdi[2],color='red')+
  geom_hline(yintercept = sortdens_gini[i],color='red')+
  annotate(geom='text',x=hpdi[2]+0.05,y=sortdens_gini[i]+0.2,label='95% Highest Posterior Density')
```

Fig 2.4




```
Interval=data.frame("lower"=c(equaltint[1],hpdi[1]),"upper"=c(equaltint[2],hpdi[2]))
rownames(Interval)=c("equal tail credible interval","HPDI")
Interval
```

```
##                                  lower     upper
## equal tail credible interval 0.1499021 0.4439632
## HPDI                         0.1368328 0.4044398
```

3a) Derive the expression for what the posterior p(k|y,μ) is proportional to.

The prior distribution is as follows:

$$P(\kappa) \sim Exp(\lambda), \lambda = 0.5$$
$$P(\kappa) = \lambda e^{-\lambda\kappa}$$
$$P(\kappa) \sim e^{-\lambda\kappa}$$

The Likelihood distribution is as follows:

$$P(y|\mu,\kappa) = \frac{e^{\kappa}.\sum_{i=1}^{n} cos(y_{i-\mu})}{(2\pi I_0(\kappa))^n}$$

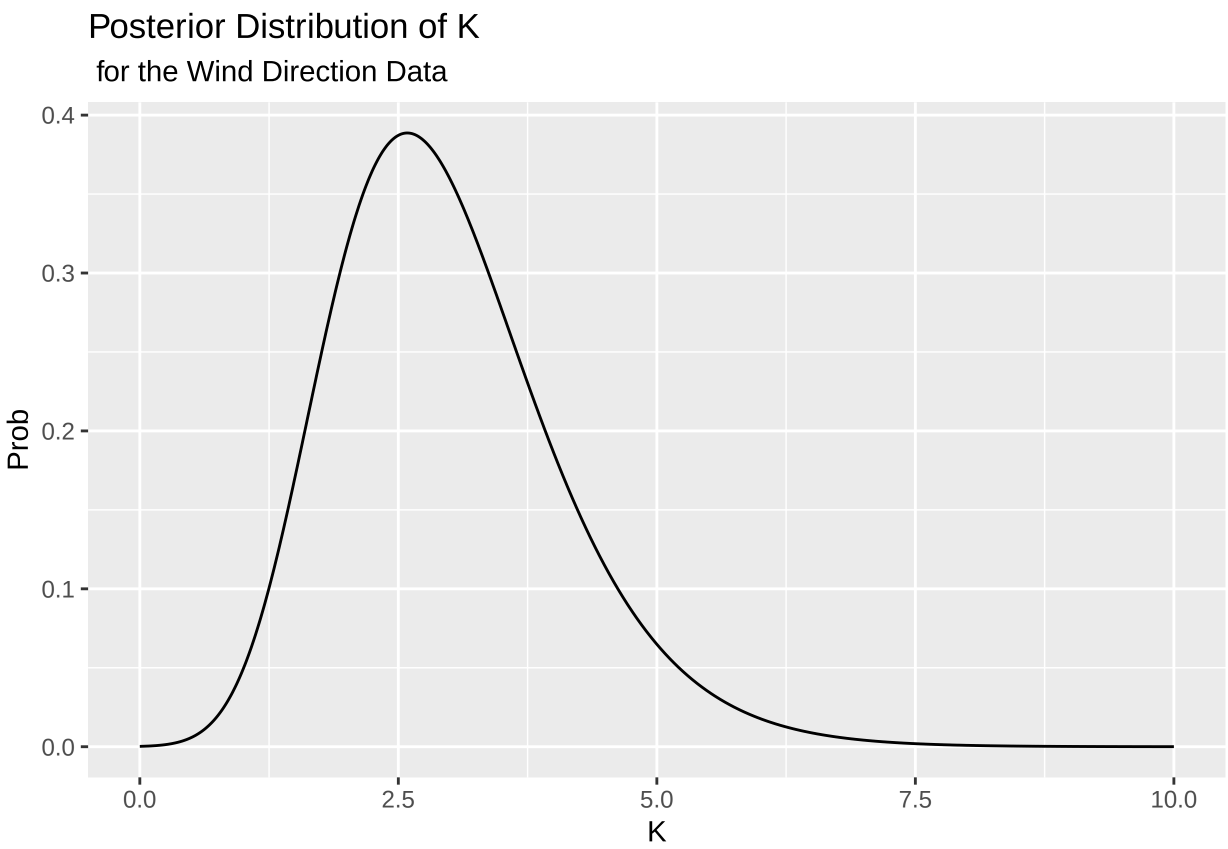
$$P(y|\mu,\kappa) \sim \frac{e^{\kappa}.\sum_{i=1}^{n} cos(y_{i-\mu})}{(I_0(\kappa))^n}$$

The posterior distribution thus becomes:

$$P(\kappa|y,\mu) \sim P(y|\mu,\kappa).P(\kappa)$$

$$P(\kappa|y,\mu) \sim \frac{e^{\kappa(\sum_{i=1}^{n} cos(y_{i-\mu})-\lambda)}}{(I_0(\kappa))^n}$$

```r
posterior=function(k,y,mu,l){
  prod=1
  for(i in 1:length(y)){
    prod=prod*(exp(k*cos(y[i]-mu))/(2*pi*besselI(k,0)))
  }
  p=prod*l*exp(-l*k)
  return(p)
}

y=c(-2.79,2.33,1.83,-2.44,2.23,2.33,2.07,2.02,2.14,2.54)
mu=2.4
l=0.5
prob=sapply(seq(0,10,0.01), posterior,y=y,mu=mu,l=l)
prob=prob/sum(prob)*100
dfplot=data.frame(x=seq(0,10,0.01),y=prob)
ggplot(data=dfplot)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=' for the Wind Direction Data',
       x='K',y='Prob')
```



3b) Find the approximate posterior mode pf k from the information in a).

```
approxPost=function(k){
  y=c(-2.79,2.33,1.83,-2.44,2.23,2.33,2.07,2.02,2.14,2.54)
  mu=2.4
  l=0.5
  prod=1
  for(i in 1:length(y)){
    prod=prod*(exp(k*cos(y[i]-mu))/(2*pi*besselI(k,0)))
  }
  p=prod*l*exp(-l*k)
  return(-p)
}
mode=optim(par=2.5,fn=approxPost,method=c('Brent'),lower=0,upper=10)
ggplot(data=dfplot)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=sprintf('The mode is %f',mode$par),
       x='K',y='Prob')+
  geom_vline(xintercept = mode$par,color='red')+
  annotate(geom='text',x=mode$par+0.5,y=0.1,label='mode',color='red')
```