Computer Lab1

Andreas C Charitos-andch552 Jiawei Wu-jiawu449

3 April 2019

Exersice 1-Bernoulli

We are given that $y_1, .., y_n/\theta \sim Bern(\theta)$ for $n = 20, s = 14$ $(n = sample, s = suceses)$ with prior $Beta(\alpha_o, \beta_o), \alpha_o = \beta_o = 2$

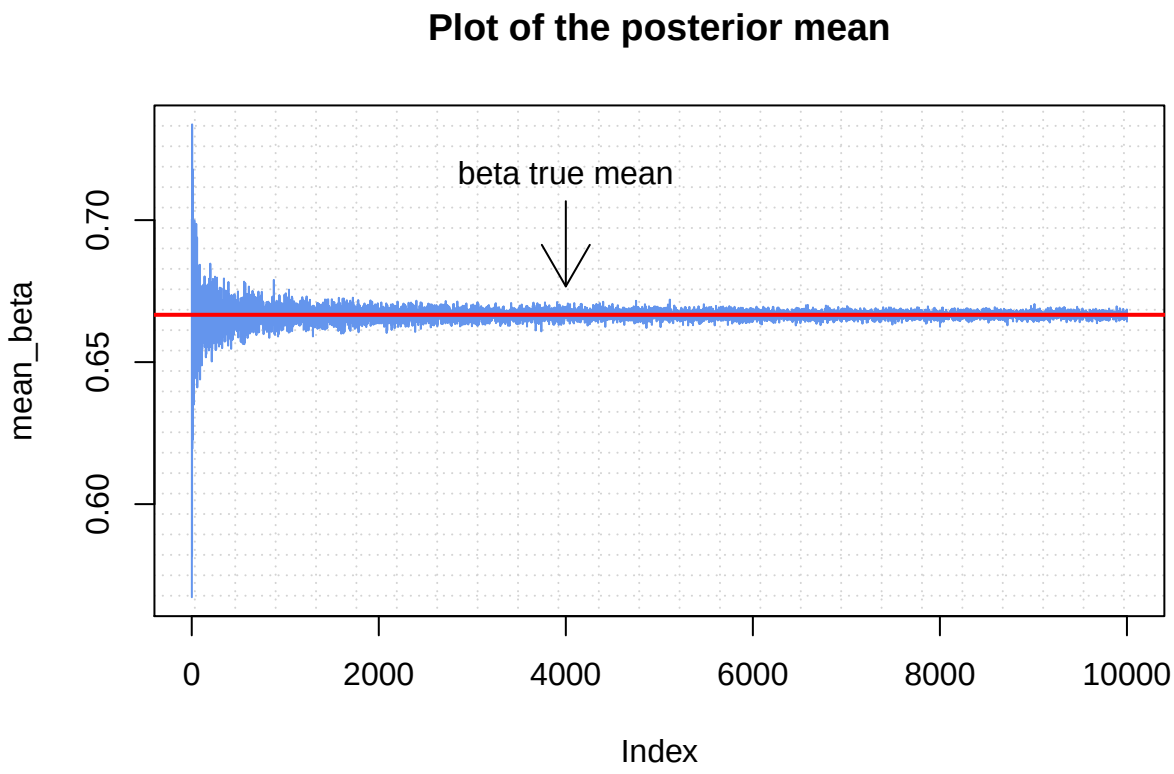
Question 1.a

Draw random numbers from the posterior $\theta \sim Beta(\alpha_o + s, \beta_o + f)$ and verify graphically that the posterior mean and standard deviation converges to the true values as the number of random draws grows. large

```
# set seed
set.seed(123456)
#initiate vectors for mean and sd
mean_beta=double(length(1:10000))
sd_beta=double(length(1:10000))
# Vector of beta parameters : a,b
ab <- c(16, 8)

# Simulate 1000 draws from the beta posterior: beta_sim
for(i in 1:10000){
  beta_sim <- rbeta(i, ab[1], ab[2])
  mean_beta[i]<-mean(beta_sim)
  sd_beta[i]<-sd(beta_sim)

}
# calculate the true mean analytically
beta_true_mean=ab[1]/(ab[1]+ab[2])
# calculate the true sd analytically
beta_true_sd=sqrt((ab[1]*ab[2])/((ab[1]+ab[2])^2*(ab[1]+ab[2]+1)))
```

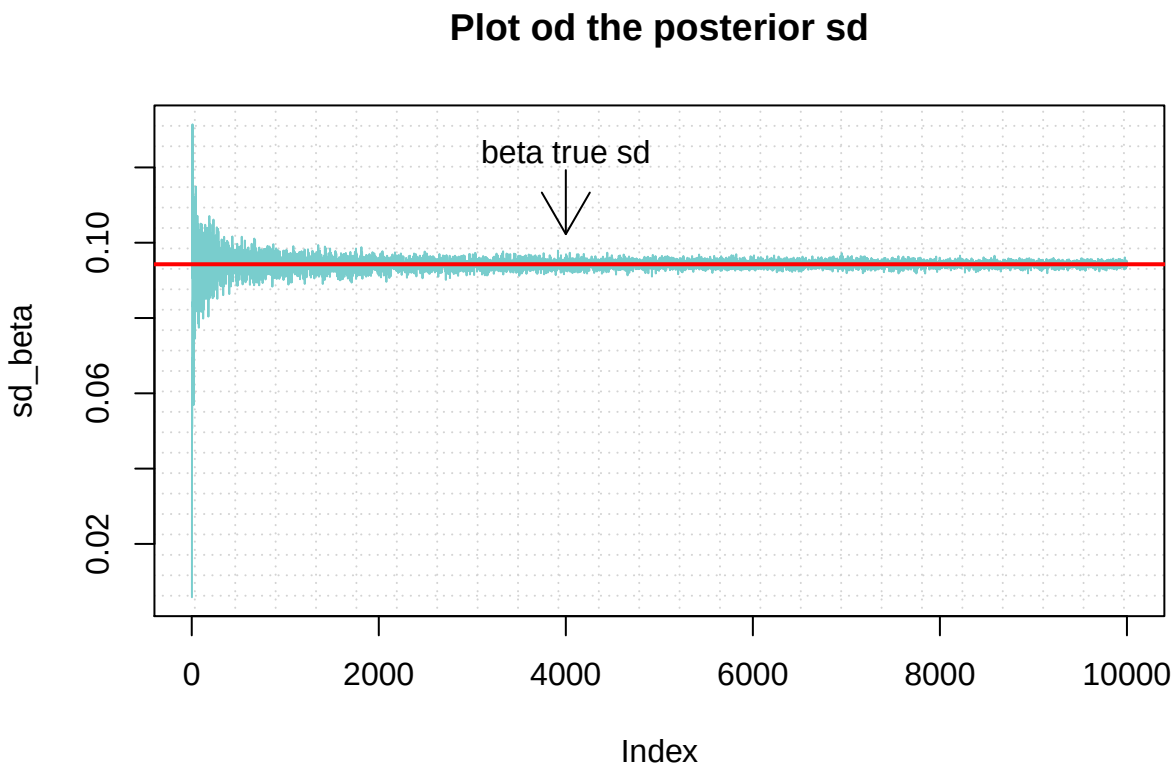
Plot of posterior mean

```
# plot of the posterior mean with abline the true mean
plot(mean_beta,type="l",col="cornflowerblue",main="Plot of the posterior mean",
      panel.first=grid(25,25))
abline(h=beta_true_mean, col="red",lwd=2)
text(x=4000, y=beta_true_mean+0.05, "beta true mean")
arrows(4000, beta_true_mean+0.04, x1 = 4000, y1 = beta_true_mean+0.01, length = 0.25,
       angle =30)
```



Plot of posterior variace

```
# plot of the posterior sd with abline the true sd
plot(sd_beta,type="l",col="darkslategray3",main="Plot od the posterior sd",
      panel.first=grid(25,25))
abline(h=beta_true_sd,col="red",lwd=2)
text(x=4000, y=beta_true_sd+0.03, "beta true sd")
arrows(4000, beta_true_sd+0.025, x1 = 4000, y1 = beta_true_sd+0.008, length = 0.25,
       angle =30)
```

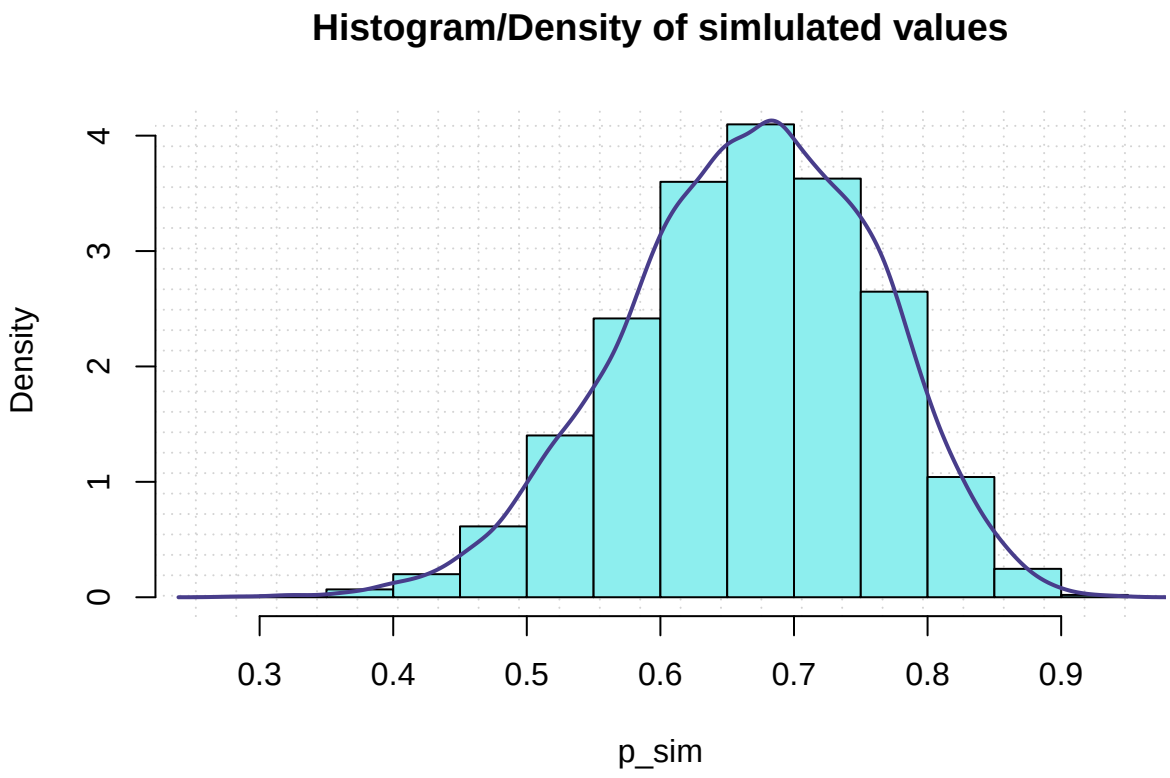


From the 2 plots we can see that as the sample increases the sample mean and sd are convering to the true mean and sd.

Question 1.b

Use simulation (nDraws=10000) to compute the posterior $Pr(\theta < 0.4)$ and compare with the exact value

```
set.seed(123456)
# sample size
n=10000
# Vector of beta parameters : a,b
ab <- c(16, 8)
# Simulate 1000 draws from the beta posterior: p_sim
p_sim <- rbeta(n, ab[1], ab[2])
# Construct a histogram of the simulated values
hist(p_sim,prob=T,main="Histogram/Density of simlulated values",
     col="darkslategray2", panel.first=grid(25,25))
lines(density(p_sim),lwd=2,col="darkslateblue")
```



The above plot shows the histogram and density for a sample of size $n = 10000$ for a $Beta(\alpha = 16, \beta = 8)$

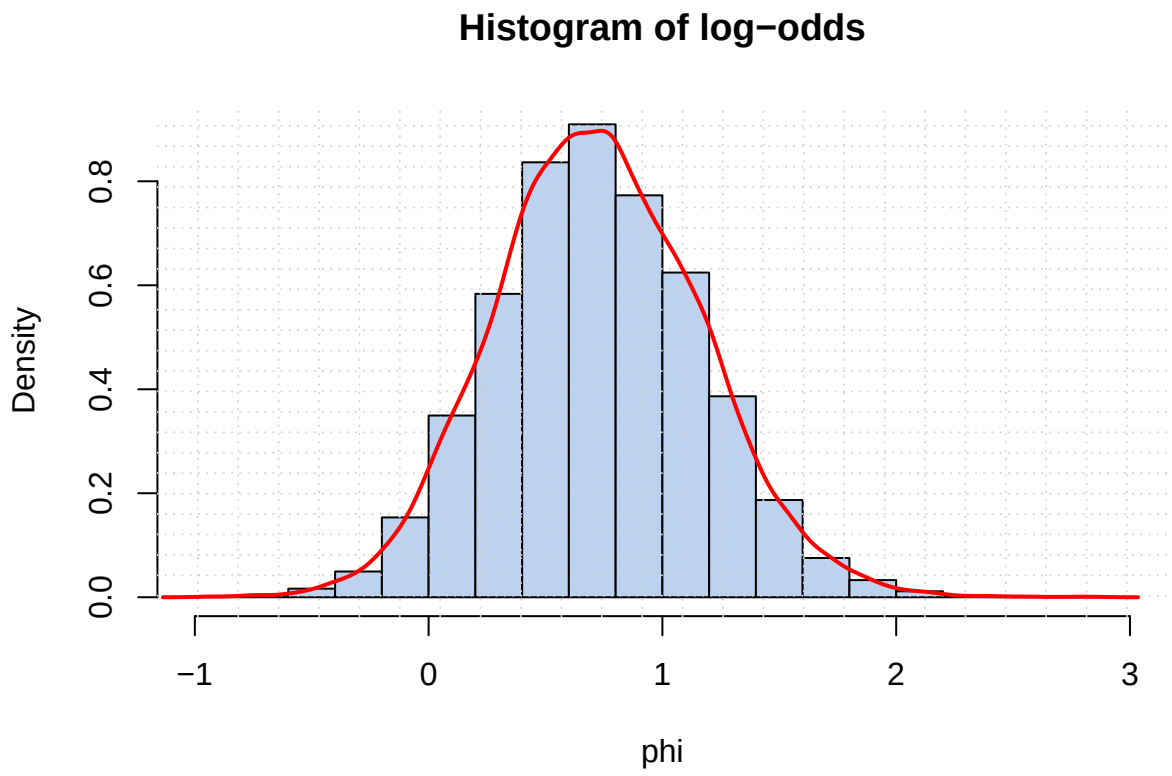
```
# Compute the probability that P is smaller than 0.4
prob_sim<-sum(p_sim < 0.4) / n
# the exact value is
prob_exact<-pbeta(0.40, ab[1], ab[2])
# quantile
quant<-quantile(p_sim, c(0.05, 0.95))

cat("The probability <0.4 from simulated data is : ",prob_sim,
    "\nThe exact probability <0.4 is : ",prob_exact)
```

```
## The probability <0.4 from simulated data is :  0.0043
## The exact probability <0.4 is :  0.003972681
```

Question 1.c

Compute the posterior distribution of log-odds $\phi = \frac{\theta}{1-\theta}$ by simulation (nDraws=10000)

```
# calculate log odds
phi=log(p_sim/(1-p_sim))
# histogram
hist(phi,prob=T,col="lightsteelblue2",main="Histogram of log-odds")
lines(density(phi),lwd=2,col="red", panel.first=grid(25,25))
```



The plot above shows the distribution for the log-odds from simulation with $nDraws = 10000$

Exersice 2-Log-normal distribution and the Gini coefficient

We are given a sample of incomes $(14, 25, 45, 25, 30, 33, 19, 50, 34, 67)$ $\sim logN(\mu, \sigma^2)$ where the density of log-Normal is :

$$p(y/\mu, \sigma^2) = \frac{1}{y\sqrt{2}\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(logy-\mu)^2} \quad , y > 0, \mu > 0, \sigma^2$$

Also if y $\sim logN(\mu, \sigma^2)$ then $logy$ $\sim N(\mu, \sigma^2)$ Let $y_1, ..., y_n$ $\sim logN(\mu, \sigma^2)$ where $\mu = 3.5$ and σ^2 unkown with prior $p(\sigma^2) \propto \frac{1}{\sigma^2}$.The posterior for σ^2 is $Inv - \chi^2(n, \tau^2)$ where

$$\tau = \frac{\sum_{i=1}^{n}(logy - \mu)^2}{n}$$

Question 2.a

Simulate 10.000 draws from $\sigma^2, (\mu = 3.5)$ and compare with the thoeretical $Inv - \chi^2(n, \tau^2)$ posterior distibution

```
data<-c(14,25, 45, 25, 30, 33, 19, 50, 34, 67)
cat("The mean of the log data is: ",mean(log(data)))
```

```
## The mean of the log data is:  3.436869
```

```
cat("\n")
```

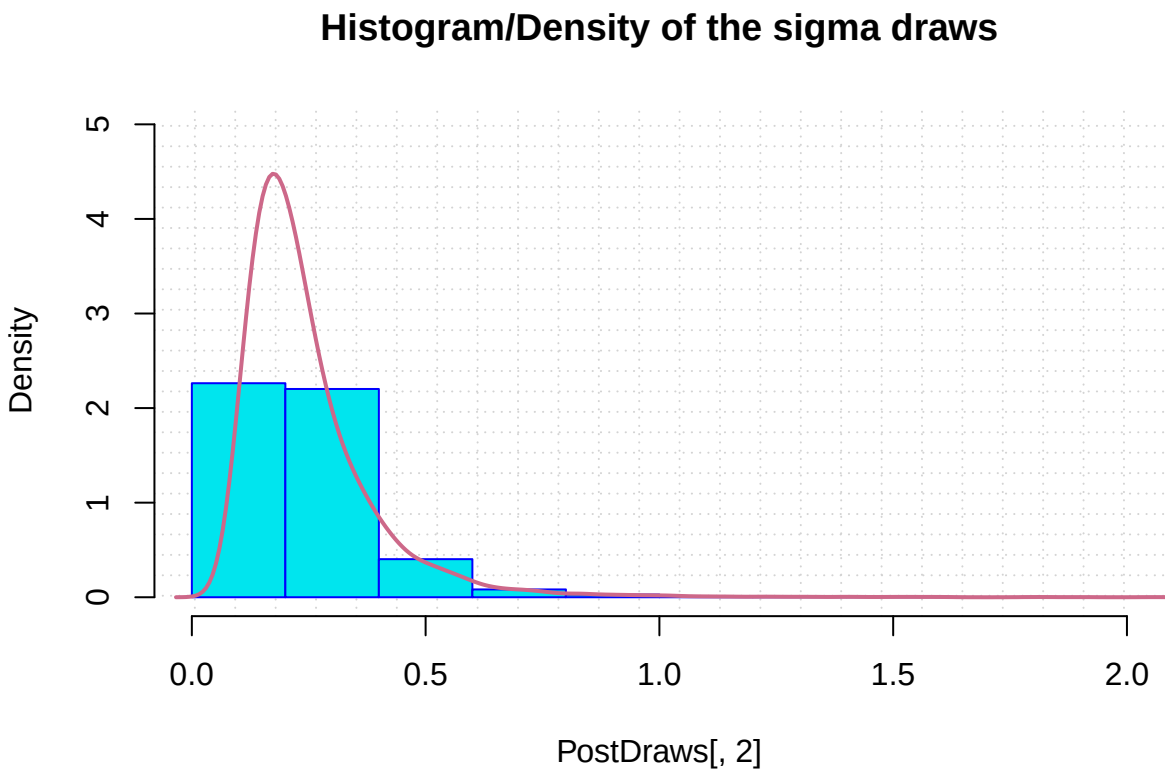
```r
cat("The variance of log data is: ",var(log(data)))
```

```
## The variance of log data is:  0.2154381
```

```r
NormalNonInfoPrior<-function(NDraws,Data){

  ##############################################################################################
  # PURPOSE:    Generates samples from the joint posterior distribution of the parameters in the
  #
  #     MODEL: x1,....xn iid Normal(mu,sigma^2) model.
  #     PRIOR: p(mu,sigma^2) propto 1/sigma^2
  #
  # INPUT:  NDraw:      (Scalar)    The number of posterior draws
  #     Data:       (n-by-1)    Data vector of n observations
  #
  # OUTPUT: PostDraws:  (NDraws-by-2)   Matrix of posterior draws.
  #                     First column holds draws of mu
  #                     Second column holds draws of sigma^2
  #
  # AUTHOR: Mattias Villani, Sveriges Riksbank and Stockholm University.
  ##############################################################################################

  n<-length(Data)
  Datamean<-mean(Data)
  tau2<-(sum((Data-3.5)^2))/n
  PostDraws=matrix(0,NDraws,2)
  PostDraws[,2]<-(n*tau2)/rchisq(NDraws,n)
  PostDraws[,1]<-Datamean+rnorm(NDraws,0,1)*sqrt(PostDraws[,2]/n) # we don't need that in our case

  return(PostDraws)
}
```

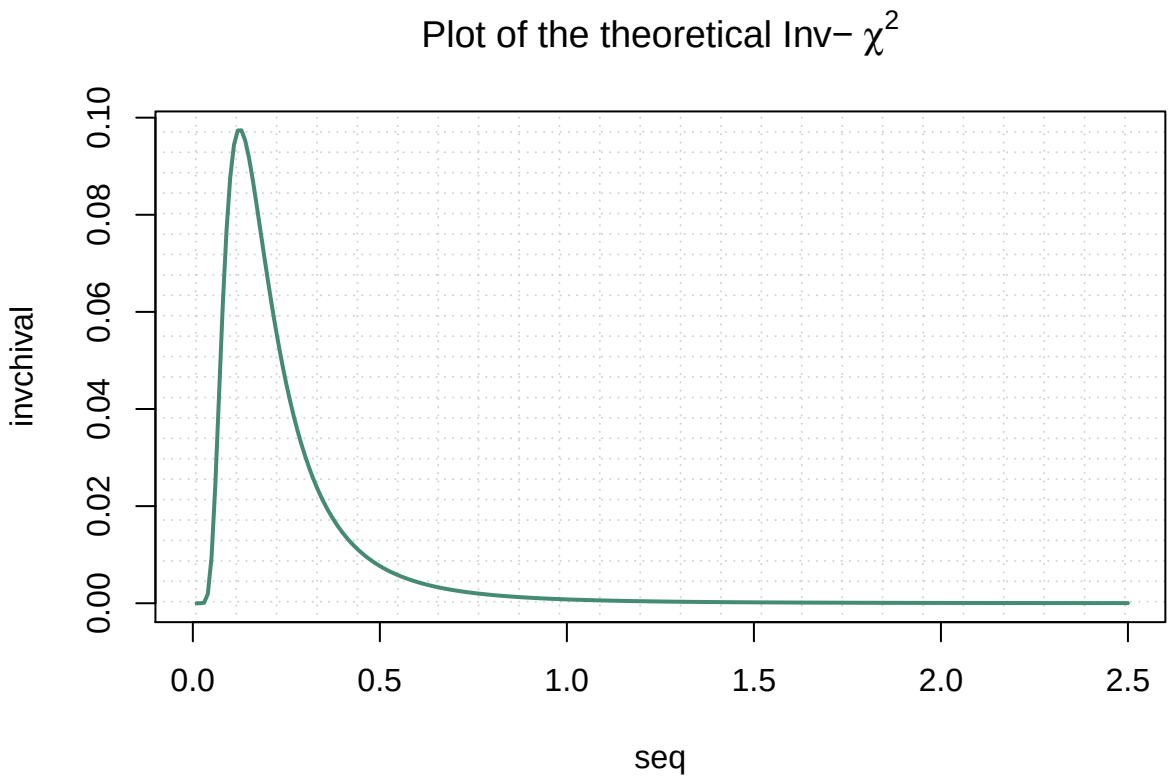
Plot of the sigma-draws

```r
# Generating 1000 draws from the joint posterior density of mu and sigma^2
PostDraws<-NormalNonInfoPrior(10000,log(data))
# Plotting the histogram of sigma-draws
hist(PostDraws[,2],prob=T,border="blue",
     main="Histogram/Density of the sigma draws",
     ylim=c(0,5),xlim=c(0,2),col="turquoise2",
      panel.first=grid(25,25))
lines(density(PostDraws[,2], adjust=2), col="palevioletred3", lwd=2)
```



Plot of the theoretical inverse chi-square

```
# make a grid
seq=seq(0,2.5,0.01)
#tao2=sum((log(salary)-3.5)^2)/10
# inverse chi-square function
invchi<-function(n,x){
  y=2^(-n/2)/gamma(n/2)*x^(-n/2+1)*exp(-1/(2*x))
  return(y)
}
# inverse scaled chi-squared function
invchiscaled<-function(n,x,s){
  y=(n/2)^(-n/2)/gamma(n/2)*s^n*x^(-n/2+1)*exp(-n*s^2/(2*x))
  return(y)
}
#values=invchis(10,seq,tao2)
# make a sample
invchival=invchi(10,seq)
#ress=as.data.frame(cbind(seq,values))
plot(seq,invchival,type="l", panel.first=grid(25,25),
     col="aquamarine4",lwd=2,
     main=expression(paste("Plot of the theoretical Inv- ",chi^2)))
```

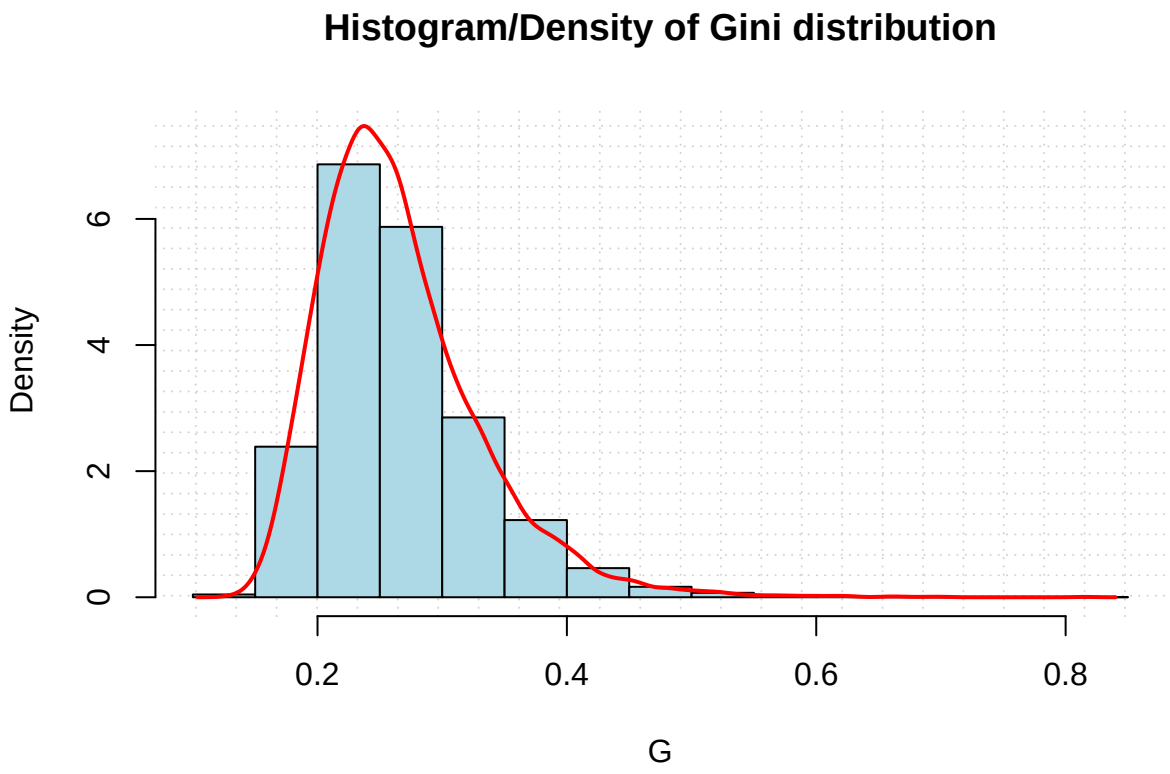


Comparing the plots obtained we can say that our simlulation method approximates the theoretical distribution very well.

Question 2.b

Use the posterior draws from a) to compute the posterior distribution of Gini coefficient $G = 2\Phi(\frac{\sigma}{\sqrt{2}}) - 1$ where $\Phi(\zeta)$ is the CDF of standard normal distibution with $\mu = 0$ $\sigma^2 = 1$

```
# compute gini coefficient
G=2*pnorm(sqrt(PostDraws[,2])/sqrt(2),mean=0,sd=1)-1
# hostogram of the Gini distribution
hist(G,col="lightblue",freq = F,
     main="Histogram/Density of Gini distribution",
     ylim = c(0,7.5), panel.first=grid(25,25))
lines(density(G),col="red",lwd=2)
```



Question 2.c

Use the posterior draws from b) to compute 95% equal tail credible interval for G where 95% equal tail interval (a; b) cuts off 2:5% percent of the posterior probability mass to the left of a, and 97:5% to the right of b.Also compute the kernel density (density() function) and use to compute compute a 95% Highest Posterior Density interval for G

```
#tail credible intervalfor G
interval=quantile(G, c(0.025, 0.975))
cat("The equal tail credit 95% interval is :\n")
```

```
## The equal tail credit 95% interval is :
```

```
interval
```

```
##      2.5%     97.5%
## 0.1731751 0.4187257
```

Plot of G Histogram with 95% equal tail cradit interval

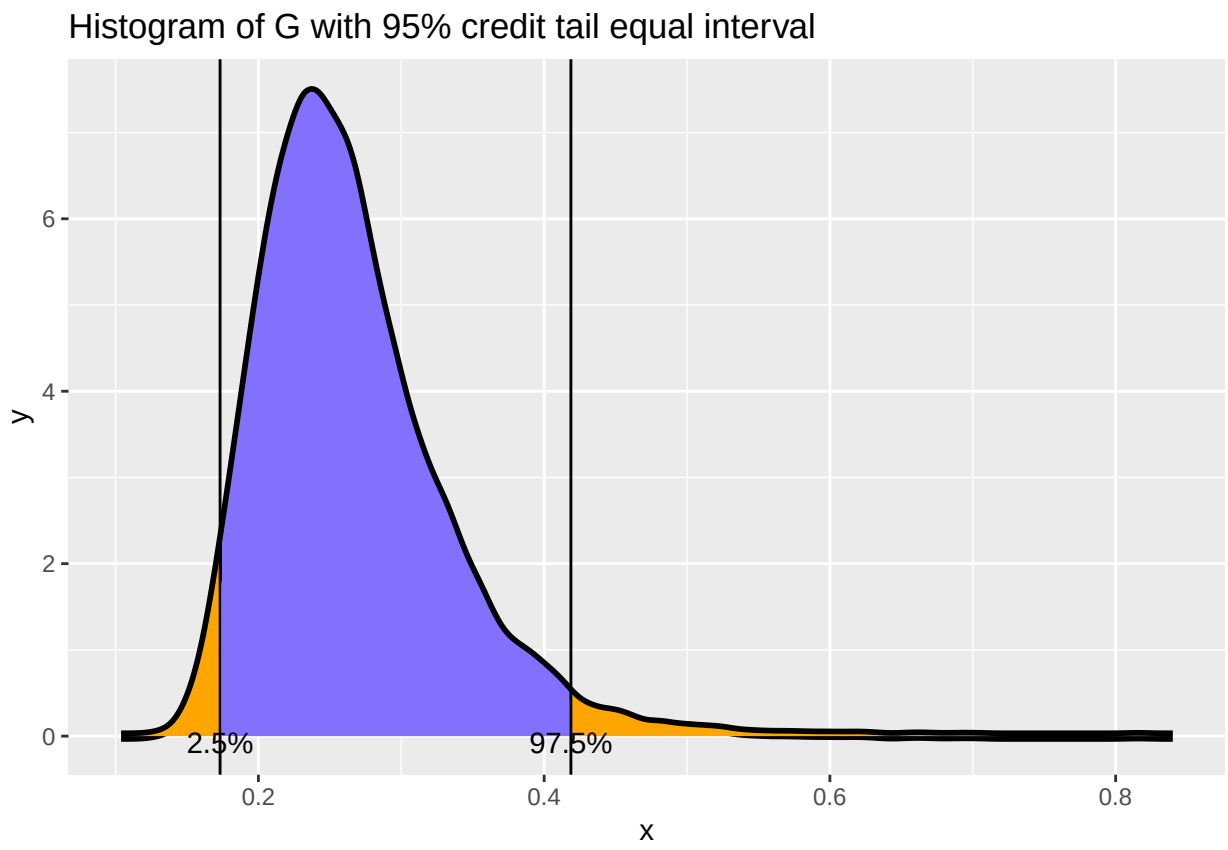
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
# calculate density
dens <- density(G)
# make data tible with case_when
data <- tibble(x = dens$x, y = dens$y) %>%
  mutate(variable = case_when(
    (x >= interval[1] & x <= interval[2]) ~ "On",
    ( x<=interval[1]) ~ "Off",(x >=interval[2])~"onOff",
    TRUE ~ NA_character_))
#> Warning: package 'bindrcpp' was built under R version 3.4.4
ggplot(data, aes(x, y)) + geom_line(size=2) +
  geom_area(data = filter(data, variable == 'On'), fill = 'lightslateblue') +
  geom_area(data = filter(data, variable == 'Off'), fill = 'orange')+
  geom_area(data = filter(data, variable == 'onOff'), fill = 'orange')+
  ggtitle("Histogram of G with 95% credit tail equal interval")+
  annotate(geom='text', x=interval[1],y=-0.07, label="2.5%")+
  annotate(geom='text', x=interval[2],y=-0.07, label="97.5%")+
  geom_vline(xintercept = interval[1])+geom_vline(xintercept = interval[2])
```




```
library(coda)
```

```
## Warning: package 'coda' was built under R version 3.5.2
```

```
# find the y point that covers 95% probability
# calculate density
dx <- density(G)
# make matrix from x,y in density-dx
mat<-cbind(dx$x,dx$y)
# oreder the matrix by col y decrending
mat<-mat[order(mat[,2], decreasing = TRUE),]
# calculate the probability
dn <- cumsum(mat[,2])/sum(mat[,2])
# fnd the indx of y where 0.95%
bi <- which(dn>=0.95)[1]
# find the y from indx
yy<-mat[bi,2]
# Highest Posterior Density interval

HPI<-HPDinterval(as.mcmc(G), prob=0.95)
cat("The higher posterior interval is :\n")
```

```
## The higher posterior interval is :
```

```
HPI
```

```
##          lower    upper
## var1 0.1617232 0.394816
## attr(,"Probability")
## [1] 0.95
```

Plot of G Histogram with 95% HPI interval

```
# make tible data
data1 <- tibble(x = dx$x, y = dx$y) %>%
  mutate(variable = case_when(
    (y >= yy) ~ "On",
    ( y<yy) ~ "Off",
    TRUE ~ NA_character_))
#> Warning: package 'bindrcpp' was built under R version 3.4.4

ggplot(data1, aes(x, y)) + geom_line(size=2) +
  geom_area(data = filter(data1, variable == 'On'), fill = 'skyblue1') +
  geom_area(data = filter(data1, variable == 'Off'), fill = 'slateblue')+
  annotate(geom='text', y=yy+0.2,x=0.44, label="95% HPI")+
  geom_vline(xintercept = HPI[,1])+geom_vline(xintercept = HPI[,2])+
  geom_hline(yintercept = yy)
```

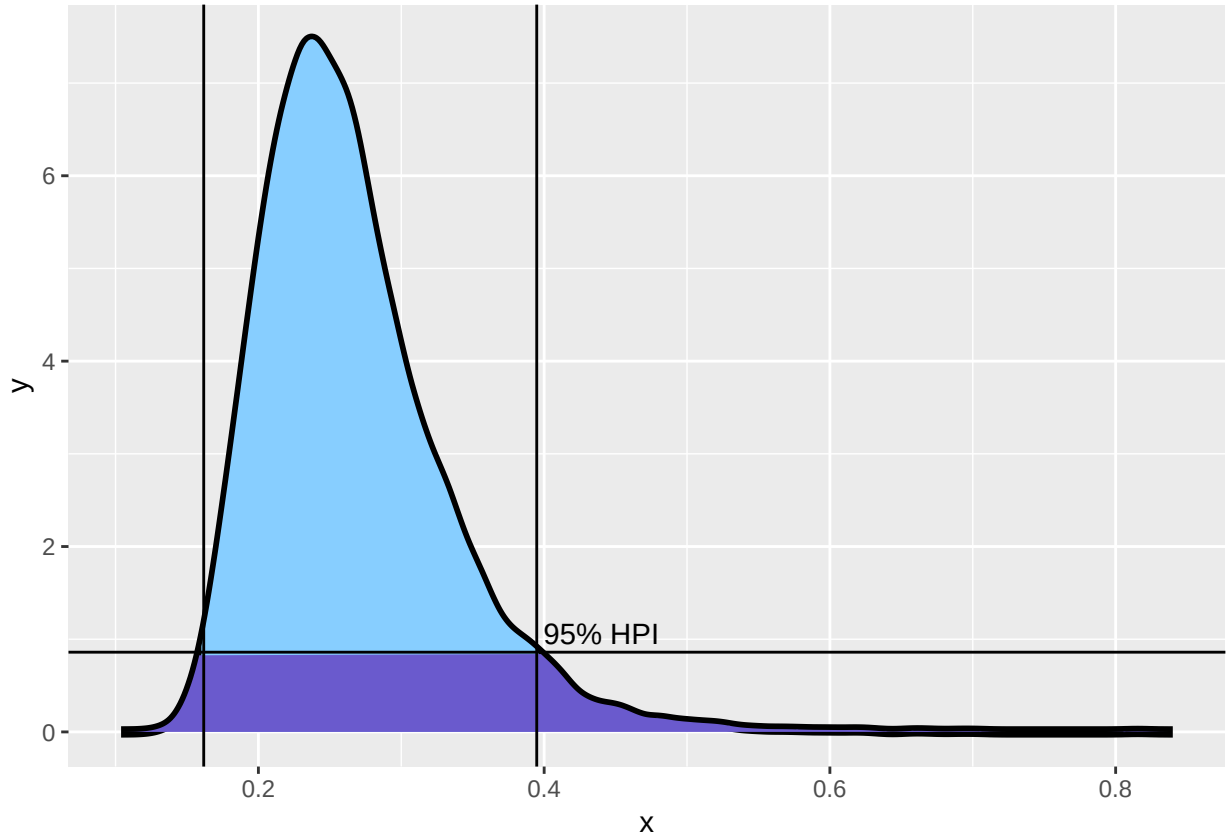


Table of the intervals

```
library(knitr)
table<-rbind(interval,HPI)
colnames(table)<-c("lower_bound","higher_bound")
rownames(table)<-c("equal tail credit interval","higher posterior interval")
kable(data.frame(table))
```

	lower_bound	higher_bound
equal tail credit interval	0.1731751	0.4187257
higher posterior interval	0.1617232	0.3948160

Generally, quantiles will give interval containing probability mass concentrated around the median (the middle $100\alpha\%$ of your distribution), while highest density region is a region around the modes of the distribution.(source https://stats.stackexchange.com/questions/240749/how-to-find-95-credible-interval) As we can see from the table the 2 intervals are very close together.From the plots we can see that the higher posterior interval cuts the plot in 2 parts horizontal with the upper part contaning the 95% HPI.On the other hand, the equal tail interval cuts the plot vertically with the are between the interval having the 95%.

Exersice 3-Bayesian inference for the concentration parameter in the von Mises distribution

We are given (-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02) wind directions at a givenlocation on ten different days in radians.Assume that these data points are independent observations following the von Mises distribution

$$p(y/\mu,\kappa) = \frac{e^{-\kappa\cos(y-\mu)}}{2I_o(\kappa)}, -\pi \leq y \leq \pi$$

and $I_o(\kappa)$ is the modified Bessel function of the first kind of order zero. The parameter $-\pi \leq y \leq \pi$ is the mean direction and $\kappa > 0$ is called the concentration parameter. Large κ gives a small variance around μ, and vice versa. Assume that μ is known to be 2:39. Let $\kappa \sim Exponential(\lambda = 1)$ apriori, where λ is the rate parameter of the exponential distribution

Question 3.a

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

We need to calculate the posterior distribution

$$p(\kappa/\mu,y) \propto\ p(\kappa) * p(y/\mu,\kappa)$$

The posterior is given that $k \sim Expon(\theta), \lambda = 1$ so the posterior is

$$p(\kappa) = \kappa e^{-\kappa}$$

The likelihood is

$$p(y_1,..,y_n/\mu,\kappa) = \prod_{i=1}^{n} \frac{e^{k\ cos(y-\mu)}}{2\pi I_o(\kappa)} =$$

$$\frac{e^{k\ \sum_{i=1}^{n} cos(y-\mu)}}{(2\pi I_o(\kappa))^n}$$

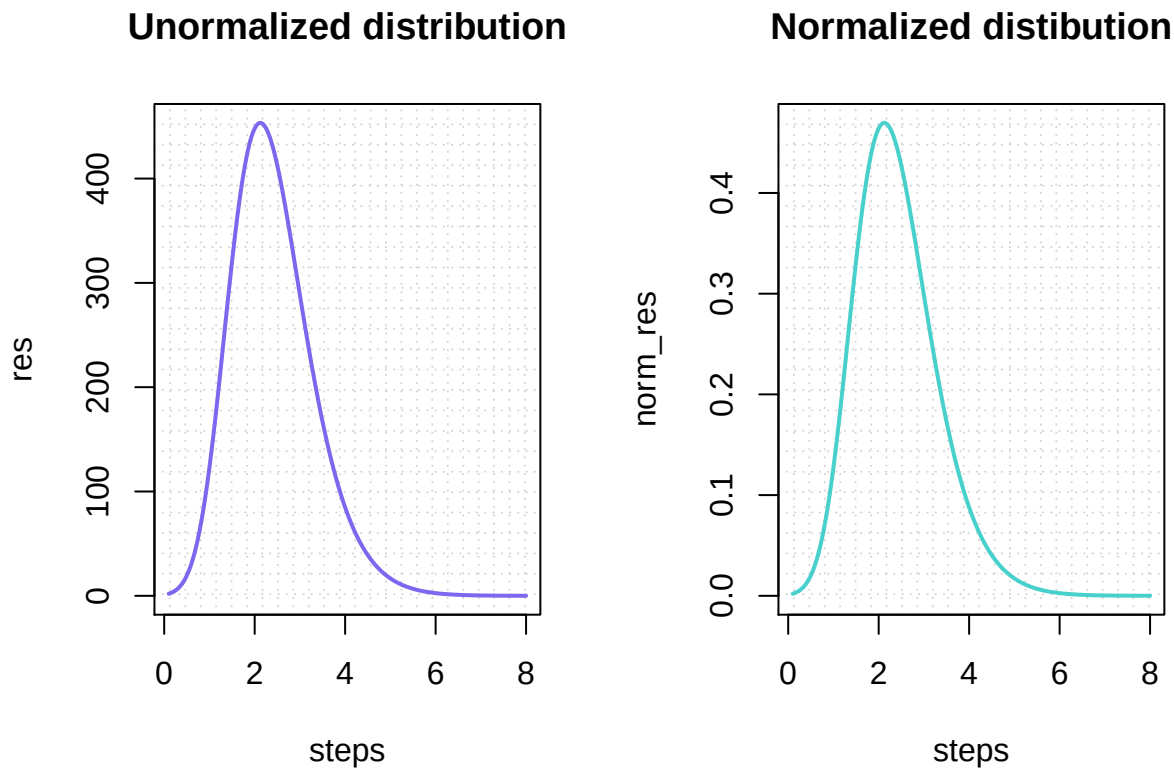
so the posterior is then

$$p(\kappa/\mu,y) \propto \frac{e^{k(\sum_{i=1}^{n} cos(y-\mu)-1)}}{I_o(\kappa)^n}$$

Plot of the unormalized and normalized posterior distribution

```
# data
dat<-c(-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02)
# posterior fuction
posterior<-function(k){
  return (exp(k*(sum(cos(dat-2.39))-1))/besselI(x = k, nu=0)^length(dat))
}
# grid
steps=seq(0.1,8,0.01)
# initiate vector
res<-double(length(seq(0.1,8,0.01)))
# loop for every value in grid
for(k in 1:length(steps)){
  res[k]<-posterior(steps[k])
}
#normalize the density so integral is 1
norm_constant=sum(res)*0.01
#normalized sample
```

```r
norm_res=res/norm_constant
#
par(mfrow=c(1,2))
plot(steps,res,type="l",main="Unormalized distribution",
     col="mediumslateblue",lwd=2, panel.first=grid(25,25))
plot(steps,norm_res,type="l",main="Normalized distibution",
     col="mediumturquoise",lwd=2, panel.first=grid(25,25))
```

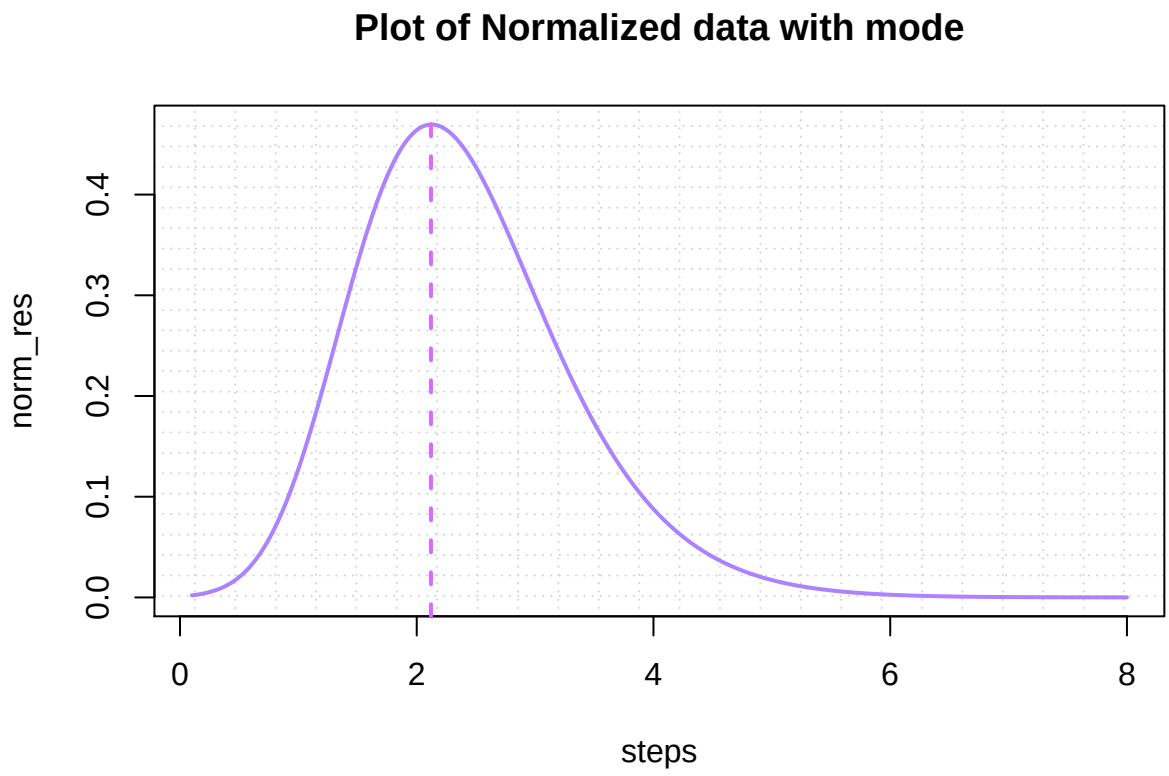


Question 3.b

Find the (approximate) posterior mode of κ from the information in a)

```r
# get the mode function
# getmode <- function(v) {
#   uniqv <- unique(v)
#   uniqv[which.max(uniqv)]
# }
#
# getmode(norm_res)

# without function
# find the index of the max
indx_mode=which.max(norm_res)
# tale the value
mode=norm_res[indx_mode]
# plot
```

```r
plot(steps,norm_res,type="l",
     main="Plot of Normalized data with mode",
     col="mediumpurple1",lwd=2, panel.first=grid(25,25))
abline(v=steps[indx_mode],lwd=2,lty="dashed",col="mediumorchid1")
```



We know thatThe mode of a set of data values is the value that appears most often.If X is a discrete random variable, the mode is the value x (i.e, X = x) at which the probability mass function takes its maximum value. In other words, it is the value that is most likely to be sampled (source wikepedia) The plot shows the normalized posterior distribution and the dashed line shows the location of the mode according to definition.