732A91-Lab1-Report

Fengjuan Chen(fench417) Zhixuan Duan(zhidu838)

4/9/2020

Question 1 Bernoulli

We have data $y_1, ..., y_n|\theta \sim Bern(\theta)$ with $s = 5$ successes in $n = 20$ trials.

The conjugate prior is $\theta \sim Beta(\alpha_0, \beta_0)$ where $\alpha_0 = \beta_0 = 2$.

Then from the Beyesian formula, the posterior is $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$ where $f = n - s$, $y = y_1, ..., y_n$.

(a) The mean and standard deviation converge to the true values

We first use rbeta() function draw random numbers from the posterior $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$.

Then calculate the mean and standard deviation of these draws.

Then calculate the true mean and standard deviation of the posterior by formulas

$mean = \frac{\alpha}{\alpha+\beta}$

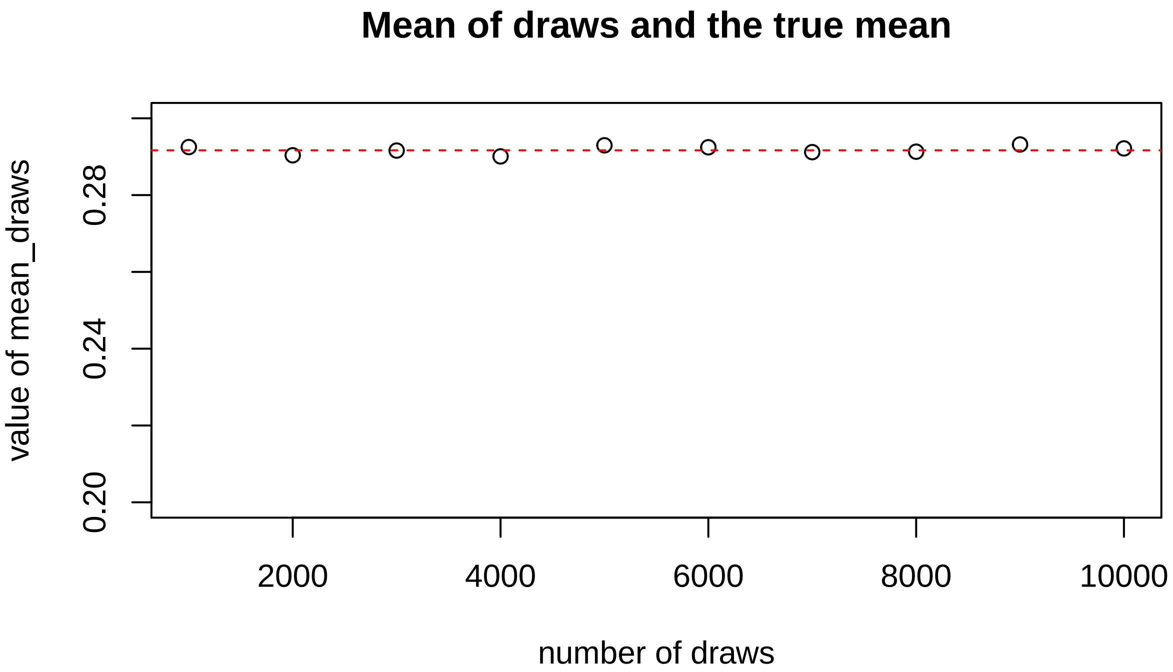
$sd = \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}$ where $\alpha = \alpha_0 + s$ and $\beta = \beta_0 + f$

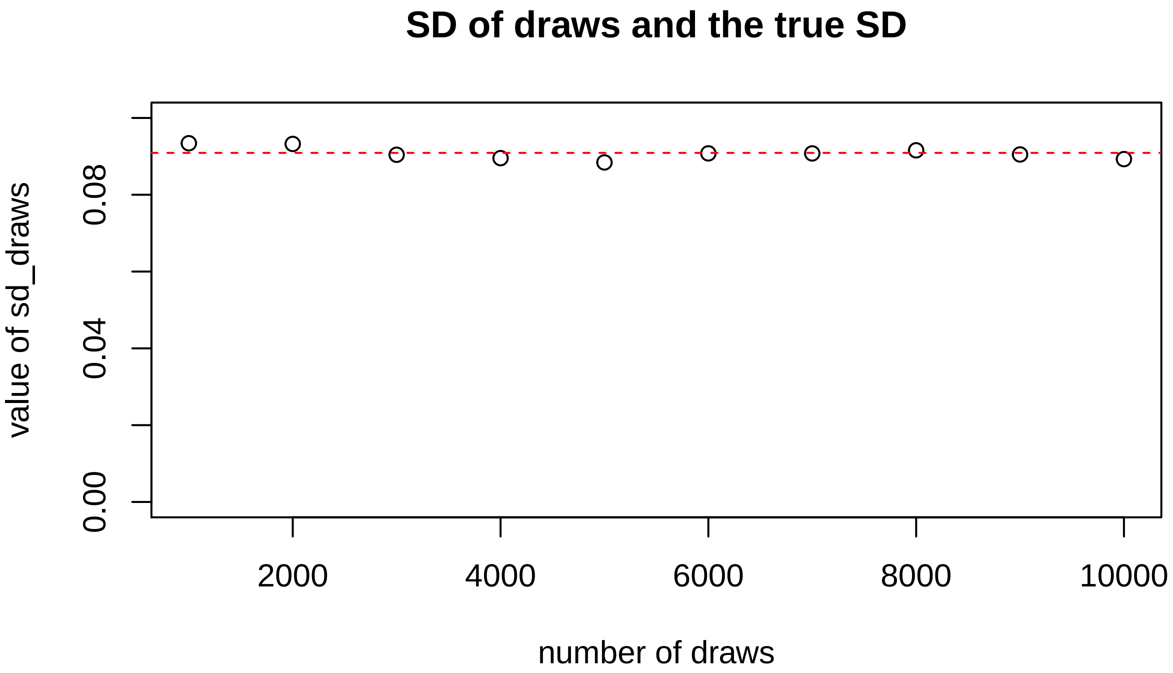
Finaly, plot the mean and standard deviation repectively.

```r
set.seed(12345)
nDraws=seq(1000,10000,1000)
mean_draws=rep(0,length(nDraws))
sd_draws=rep(0,length(nDraws))
j=1
alpha0=2
beta0=2
s=5
n=20
f=n-s
for (i in nDraws) {
  mean_draws[j]=mean(rbeta(i,alpha0+s,beta0+f))
  sd_draws[j]=sd(rbeta(i,alpha0+s,beta0+f))
  j=j+1
}
true_mean=(alpha0+s)/(alpha0+s+beta0+f)
true_var=((alpha0+s)*(beta0+f))/(alpha0+s+beta0+f)^2/(alpha0+s+beta0+f+1)
true_sd=sqrt(true_var)
plot(nDraws,mean_draws,ylim = c(0.2,0.3))
abline(h=true_mean)
plot(nDraws,sd_draws,ylim = c(0,0.1))
abline(h=true_sd)
```

The two plots are shown below.

In each plot, the staight dot red line represents the true value.


Mean of draws and the true mean
value of mean_draws
0.20
0.24
0.28
2000
4000
6000
8000
10000
number of draws


SD of draws and the true SD
value of sd_draws
0.00
0.04
0.08
2000
4000
6000
8000
10000
number of draws

(b) Compute the posterior probability

Use step1-3 to calculate the $Pr(\theta > 0.3|y)$ by 10000 simulations.

Then use step 4 to obtain the exact value of $Pr(\theta > 0.3|y)$.

Step 1. Simulate 10000 random numbers from the posterior distribution $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$.

Step 2. Count the number of draws which are more than 0.3.

Step 3. Divide the number with the total number 10000.

Step 4. Use pbeta() with parameter lower.tail=FALSE to calculate the probability $Pr(\theta > 0.3|y)$.

```
set.seed(12345)
draws=rbeta(10000,alpha0+s,beta0+f)
prob=mean(draws>0.3)
pbeta(0.3,alpha0+s,beta0+f,lower.tail = FALSE)
```

```
## The probability computed by simulation is: 0.4392
```

```
## The exact value of it is:  0.4399472
```

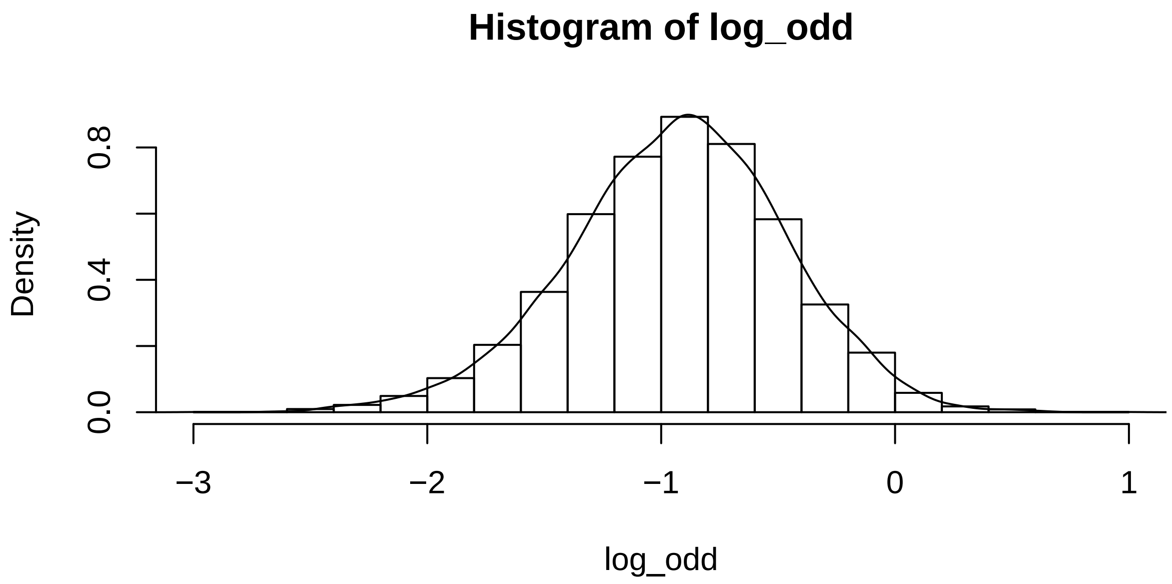
(c) The posterior distribution of the log-odds

We use the 10000 simulations obtained from (b) to solve this question.

First, translate the 10000 draws from posterior distribution of θ to the log-odds $\phi = log\frac{\theta}{1-\theta}$.

Then, plot the histogram and density of these log-odds in one picture. The figure is shown below.

```
log_odd=log(draws/(1-draws))
hist(log_odd,prob=TRUE)
lines(density(log_odd))
```



Question 2 Log-normal distribution and the Gini coefficient

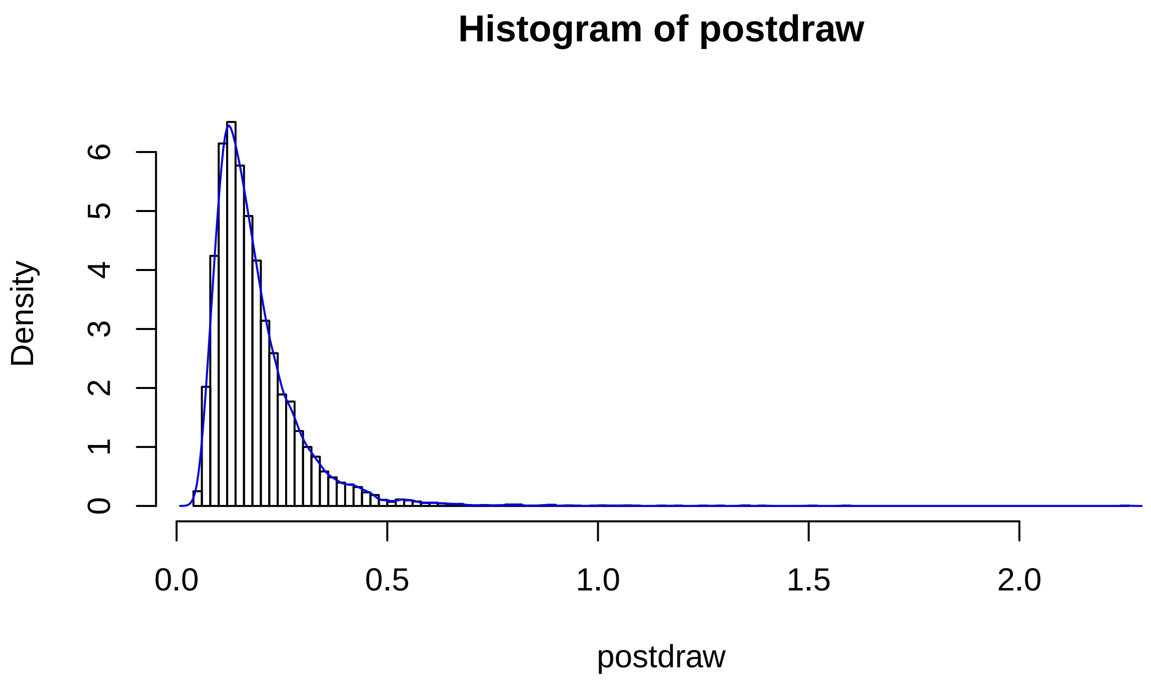
(a) Simulate 10000 draws from the posterior of sigma

Step 1. Simulate 10000 draws X from the chi-square distribution with degrees of freedom n.

Step 2. Obtain the posterior draws of σ^2 by $\sigma^2 = \frac{n*\tau^2}{X}$,

where n=10,
$\tau^2 = \frac{\sum_{i=1}^{n}(logy_i-\mu)^2}{n}$, $\mu = 3.7$ and y={44,25,45,52,30,63,19,50,34,67}.

```
n=10
y=c(44,25,45,52,30,63,19,50,34,67)
mu=3.7
tao2=sum((log(y)-mu)^2)/n
nDraws=10000
set.seed(12345)
draw_chisq=rchisq(nDraws,df=n)
postdraw=n*tao2/draw_chisq
hist(postdraw,breaks = 100,freq = FALSE)
lines(density(postdraw),col="blue")
mean(postdraw)
var(postdraw)
```



```
## The mean and variance of 10000  posterior draws are:  0.1869477 0.01189701
```

Compare with the theoretical Inv-chisqaure distribution

The theoretical scaled Inverse-chisqured probability distribution function is

$f(x) = \frac{(\tau^2 v/2)^(v/2)}{\Gamma(v/2)} \frac{exp(-\frac{v\tau^2}{2x})}{x^{1+v/2}}$ where v is the degrees of freedom. So, v is equal to n=10 here.

The mean and variance are

$mean = \frac{v\tau^2}{v-2}$

$variance = \frac{2v^2\tau^4}{(v-2)^2(v-4)}$

```
x=seq(0.01,2,0.001)
scal_inv_chisq=(tao2*n/2)^(n/2)*exp(-n*tao2/(2*x))/(gamma(n/2)*x^(1+n/2))
mean_scaled=(n*tao2)/(n-2)
var_scaled=(2*n^2*tao2^2)/((n-2)^2*(n-4))
plot(density(postdraw),col="blue",main = "Comparison of two distributions")
lines(x,scal_inv_chisq, type = "l",col="red")
```

```
## The mean and variance of theoretical scaled inv-chisquare distribution are: 0.1874264 0.01170956
```

To compare simulation results with the theoretical results, we illustrate the values of mean, variance as well as standard deviatioan in a talbe and plot densities on one figure.

In the figure, the blue curve is the posterior density obtained by simulation while the red curve is the theoretical scaled Inverse chi-squared posterior distribution.

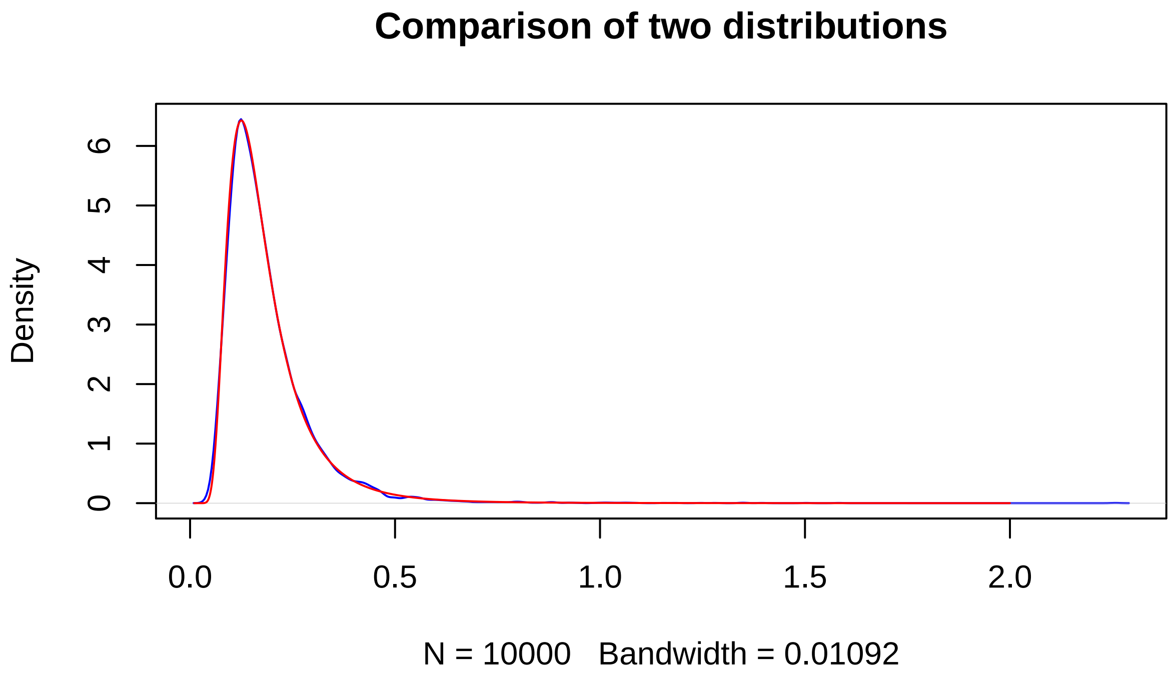


Table 1: Comparison of simulation and theoretical scaled inverse chisquare

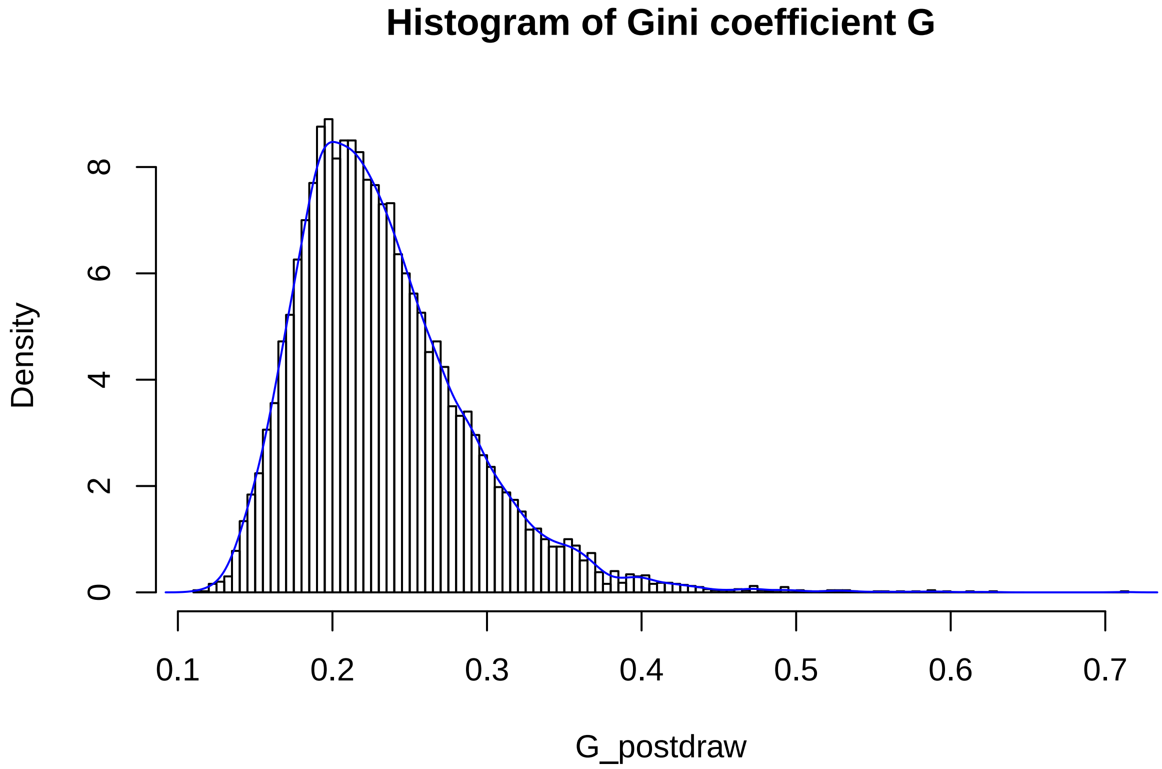
	mean	variance	standard_deviation
simulation	0.1869477	0.0118970	0.1090734
theoretical	0.1874264	0.0117096	0.1082107

(b) Compute the posterior distribution of G

The Gini coefficient G is calculated by the formula $G = 2\Phi(\sigma/\sqrt{2}) - 1$ where σ is random number simulated from the posterior distribution of σ^2.

After the calculation, we plot the histogram and density of the posterior distribution of the Ginin coefficient G for the current data set.

```r
G_postdraw=2*pnorm(sqrt(postdraw/2))-1
hist(G_postdraw,freq = FALSE,breaks = 100,main ="Histogram of Gini coefficient G" )
lines(density(G_postdraw),col="blue")
```



(c) Credible intervals for G

90% equal tail credible interval

The 90% equal tail credible interval for G is obtained by finding the 5% quantile and 95% quantile of the simulations of G. We use the quantile() function to find these quantiles.

```r
lower_equ=quantile(G_postdraw,0.05)
upper_equ=quantile(G_postdraw,0.95)
```

```
## The 90% equal tail credible interval is [ 0.1600229 0.3354106 ]
```

90% Highest Posterior Density interval

The 90% Highest Posterior Density interval for G satisfies two conditions. (https://www.sciencedirect.com/topics/mathematics/highest-density-interval)

(i) The region under the curve within the 90% HPD limits has total area of 0.9.

(ii) Any variable value within those limits has higher density than any other value outside those limits.

According these conditions, we first order the density of draws of G (density is obtained by function density()) decreasingly.

Then compute the cumulative sum of them and divide these cumulative sum by the sum of all densities to calculate the proportion.

Then, find the minimum position where the proportion is large than or equal to 0.9.

Finally, from 1 to this position is the 90% Highest Posterior Density interval.

The code is shown as follows.

```
den_x=density(G_postdraw)$x
den_y=density(G_postdraw)$y
data=data.frame(x=den_x,y=den_y,cum=rep(0,length(den_x)))
order_data=data[order(data$y,decreasing = TRUE),]
order_data$cum=cumsum(order_data$y)/sum(order_data$y)
pos=min(which(order_data$cum>=0.9))
hpd_int=sort(order_data$x[1:pos])
lower_hpd=hpd_int[1]
upper_hpd=hpd_int[pos]
```

```
## The 90% HPD interval is [ 0.1473145 0.3143029 ]
```

We call the function hdi() from package “HDInterval” to confirm our results of 90% HPD interval.

```
library(HDInterval)
HPD_region=hdi(density(G_postdraw),credMass = 0.90,allowSplit = TRUE)
height=attr(HPD_region,"height")
hpd_lower=HPD_region[1,1]
hpd_upper=HPD_region[1,2]
```

```
## The 90% HPD interval from hdi() is [ 0.1473145 0.3143029 ]
```

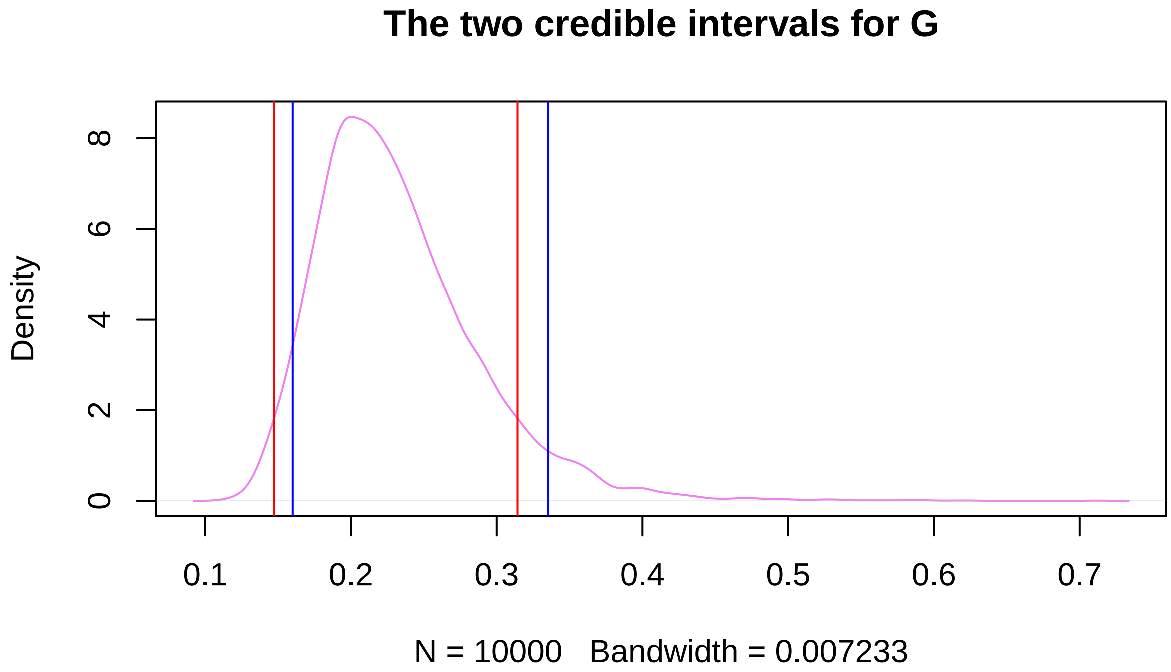
Compare the two credible intervals

To compare the two credible intervals, we first put them in a table, and then plot them in one figure with the density of the posterior of G.

Table 2: The two credible intervals for G

	lower	upper
equal_tail	0.1600229	0.3354106
HPD	0.1473145	0.3143029

In the figure below, the two blue straight lines are the lower bound and upper bound of 90% eqaul tail interval while the two red straight lines are the bounds of 90% HPD interval.



From these results, we can see that the two credible intervals are different. This is because they have different mechanism to choose the bounds. And as we all know the HPD is more useful when the density is multimodel.

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

(a) Plot the posterior distribution of kappa

In the von Mises distribution

$p(y|\mu,\kappa) = \frac{exp[\kappa \cdot cos(y-\mu)]}{2\pi I_0(\kappa)}$,

the parameter κ is the shape parameter and is reciprocal to dispersion. Here, κ is the parameter of interest.

The likelihood function is

$\prod_{i=1}^{n} p(y_i|\mu,\kappa)= \prod_{i=1}^{n} \frac{exp[\kappa \cdot cos(y_i-\mu)]}{2\pi I_0(\kappa)} = \frac{1}{[2\pi I_0(\kappa)]^n} exp[\kappa \cdot \sum_{i=1}^{n} cos(y_i-\mu)]$

The prior of κ is

$\kappa \sim Exponential(\lambda = 1)$ where λ is the rate parameter of the exponential distribution.

That is,

$p(\kappa) = \lambda \cdot e^{-\lambda\kappa} = e^{-\kappa} = exp(-\kappa)$.

Posterior of κ is $Posterior = Likelihood \cdot Prior$

$p(\kappa|y,\mu) = \frac{1}{[2\pi I_0(\kappa)]^n} exp[\kappa \cdot \sum_{i=1}^{n} cos(y_i-\mu)] \cdot exp(-\kappa)$ where $\mu = 2.39$.

Since we want to use grid method to plot the posterior distribution of κ over a fine grid of κ values, we do not make mathematical manipulation about the posterior.

We adopt the idea of grid method.

(i) Discretize the parameter space if it is not already discrete.

(ii) Compute prior and likelihood at each "grid point" in the discretized parameter space.

(iii) Compute posterior as $likelihood \cdot prior$ at each "grid point".

Find the fine grid of parameter kappa.

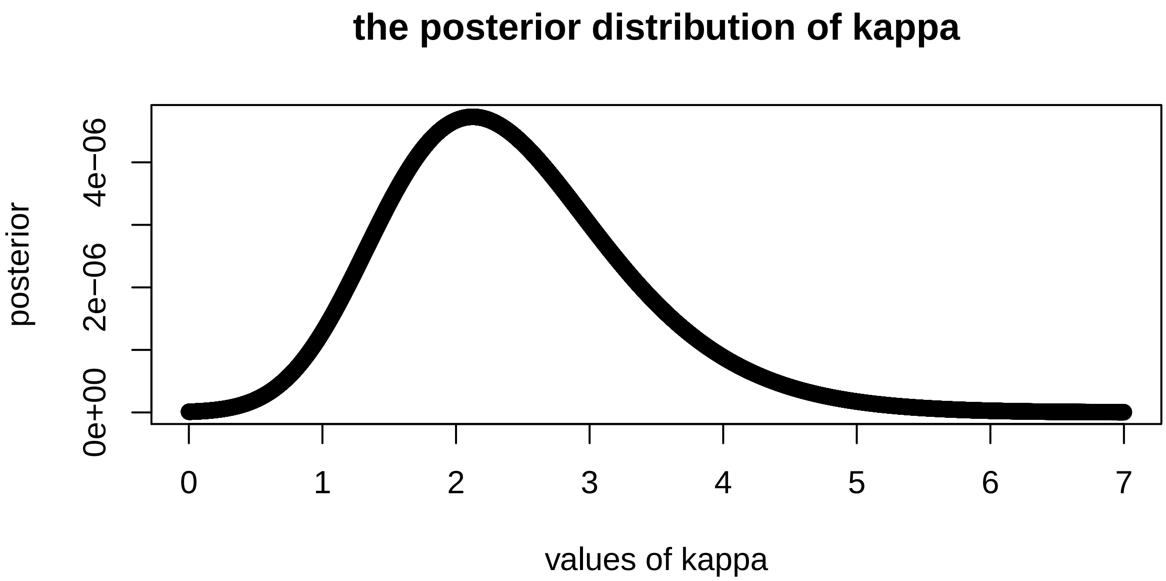
The parameter of interest κ is the shape parameter of von Mises distribution and $\kappa >= 0$. When $\kappa = 0$ the distribution is uniform.

The prior of κ is subjected to $exp(-\kappa)$, which means $\kappa = 7$ only has propability $exp(-7) = 0.00091$ from our knowledge before seeing the data.

From above information, we decide to choose 10001 grids from 0 to 7 by using function

seq(0,7,by=(7-0)/10000).

```
mu=2.39
y=c(-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02)
fine_grid=seq(0,7,by=(7-0)/10000)
post=vector(length = length(fine_grid))
j=1
for (i in fine_grid) {
  likelihood=prod(exp(i*cos(y-mu))/(2*pi*besselI(i,0)))
  prio=exp(-i)
  post[j]=likelihood*prio
  j=j+1
}
plot(fine_grid,post,main = "the posterior distribution of kappa")
```



(b) The approximate posterior mode of kappa

The posterior mode of *kappa* is the *kappa* value with the biggest posterior probability. Since we use grid method to calculate and plot the posterior distribution, we can only find the approximate posterior mode of κ.

```
mode_pos=which(post==max(post))
mode_of_k=fine_grid[mode_pos]
```

```
## The approximate posterior mode of kappa is: 2.1245
```

Appendix

```
# Question 1

## (a) Draw random numbers from the posterior
set.seed(12345)
nDraws=seq(1000,10000,1000)
mean_draws=rep(0,length(nDraws))
sd_draws=rep(0,length(nDraws))
j=1
alpha0=2
beta0=2
s=5
n=20
f=n-s
for (i in nDraws) {
  mean_draws[j]=mean(rbeta(i,alpha0+s,beta0+f))
  sd_draws[j]=sd(rbeta(i,alpha0+s,beta0+f))
  j=j+1
}

true_mean=(alpha0+s)/(alpha0+s+beta0+f)
true_var=((alpha0+s)*(beta0+f))/(alpha0+s+beta0+f)^2/(alpha0+s+beta0+f+1)
true_sd=sqrt(true_var)

plot(nDraws,mean_draws,ylim = c(0.2,0.3))
abline(h=true_mean)

plot(nDraws,sd_draws,ylim = c(0,0.1))
abline(h=true_sd)

##(b) compute Pr(theta>0.3|y) by 10000 draws and compare it with the exact value

set.seed(12345)
draws=rbeta(10000,alpha0+s,beta0+f)
prob=mean(draws>0.3)
# exact value of Pr(theta>0.3)=1-Pr(theta<=0.3)
pbeta(0.3,alpha0+s,beta0+f,lower.tail = FALSE)
```

```
## (c) log-odd

log_odd=log(draws/(1-draws))
hist(log_odd,prob=TRUE)
lines(density(log_odd))



# Question 2

##(a) simulate 10000 draws from the posterior of sigma^2
n=10
y=c(44,25,45,52,30,63,19,50,34,67)
mu=3.7
tao2=sum((log(y)-mu)^2)/n
nDraws=10000
set.seed(12345)
draw_chisq=rchisq(nDraws,df=n)
postdraw=n*tao2/draw_chisq
hist(postdraw,breaks = 100,freq = FALSE)
lines(density(postdraw),col="blue")
mean(postdraw)
var(postdraw)

### compare with the theoretical distribution

x=seq(0.01,2,0.001)
scal_inv_chisq=(tao2*n/2)^(n/2)*exp(-n*tao2/(2*x))/(gamma(n/2)*x^(1+n/2))

mean_scaled=(n*tao2)/(n-2)
var_scaled=(2*n^2*tao2^2)/((n-2)^2*(n-4))
plot(density(postdraw),col="blue",main = "Comparison of two distributions")
lines(x,scal_inv_chisq, type = "l",col="red")


## (b) Compute the posterior distribution of G

G_postdraw=2*pnorm(sqrt(postdraw/2))-1
hist(G_postdraw,freq = FALSE,breaks = 100,main ="Histogram of Gini coefficient G" )
lines(density(G_postdraw),col="blue")


## (c) Credible interval

### equal tail interval

lower_equ=quantile(G_postdraw,0.05)
upper_equ=quantile(G_postdraw,0.95)
cat("The 90% equal tail credible interval is [", lower_equ, upper_equ, "]")

### Highest Posterior Density interval

###### Calculate the HPD interval by ourselves
```

```
den_x=density(G_postdraw)$x
den_y=density(G_postdraw)$y
data=data.frame(x=den_x,y=den_y,cum=rep(0,length(den_x)))
order_data=data[order(data$y,decreasing = TRUE),]
order_data$cum=cumsum(order_data$y)/sum(order_data$y)
pos=min(which(order_data$cum>=0.9))
hpd_int=sort(order_data$x[1:pos])
lower_hpd=hpd_int[1]
upper_hpd=hpd_int[pos]
cat("The 90% HPD interval is [",lower_hpd,upper_hpd,"]")

###### call function hdi() from package HDInterval
library(HDInterval)
HPD_region=hdi(density(G_postdraw), credMass = 0.90, allowSplit = TRUE)
height=attr(HPD_region,"height")
hpd_lower=HPD_region[1,1]
hpd_upper=HPD_region[1,2]
cat("The 90% HPD interval from hdi() is [",hpd_lower,hpd_upper,"]")

### Compare the two credible intervals
library(knitr)
cred_int=data.frame(lower=c(lower_equ,lower_hpd),upper=c(upper_equ,upper_hpd))
rownames(cred_int)=c("equal_tail","HPD")
kable(caption = "The two credible intervals for G",cred_int)

plot(density(G_postdraw),col="violet",main ="The two credible intervals for G")
abline(v=lower_equ,col="blue")
abline(v=upper_equ,col="blue")
abline(v=lower_hpd,col="red")
abline(v=upper_hpd,col="red")


# Question 3

##(a) plot the posterior distribution over a fine grid
mu=2.39
y=c(-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02)
fine_grid=seq(0,7,by=(7-0)/10000)
post=vector(length = length(fine_grid))
j=1
for (i in fine_grid) {
  likelihood=prod(exp(i*cos(y-mu))/(2*pi*besselI(i,0)))
  prio=exp(-i)
  post[j]=likelihood*prio
  j=j+1
}
plot(fine_grid,post)

## (b) The approximate posterior mode of kappa

mode_pos=which(post==max(post))
mode_of_k=fine_grid[mode_pos]
```