Bayesian Learning Lab 1

Shipeng Liu Jin Yan

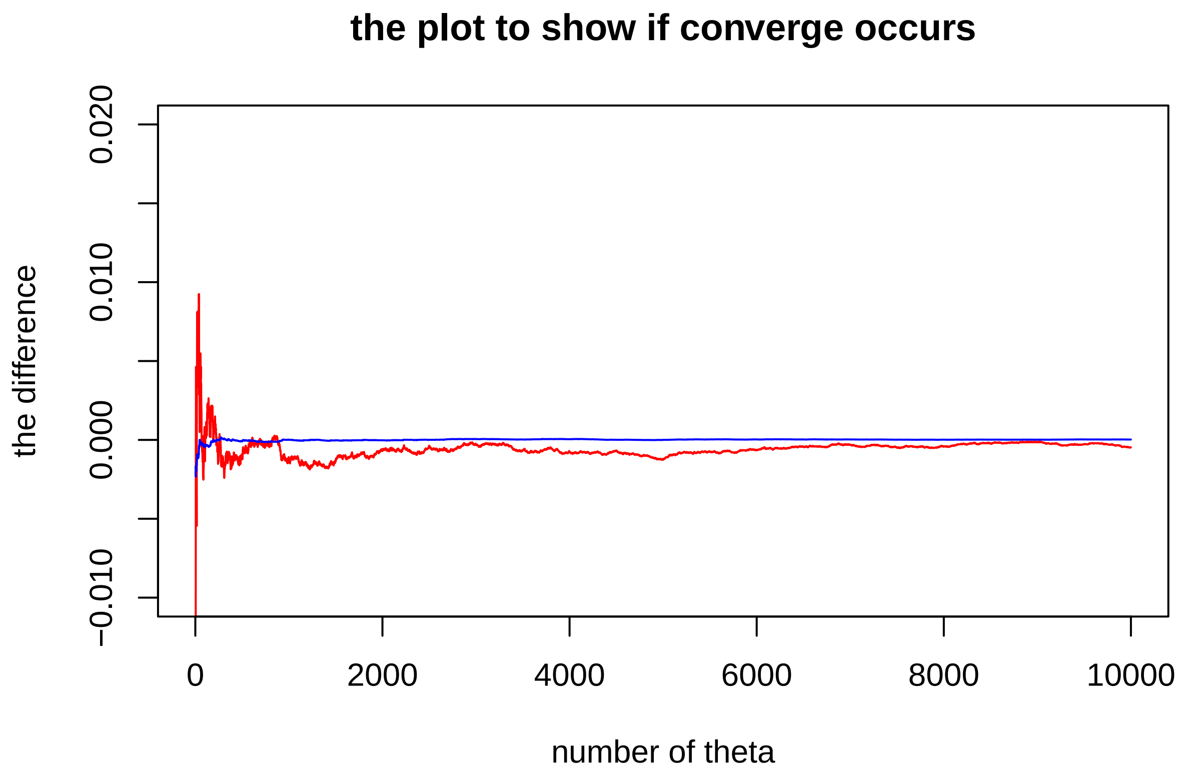
2023-04-06

Contents

Assignment 1

a) Draw 10000 random values from the posterior and verify graphically that the posterior mean and standard deviation converges to the true values as the number of random draws grows large.

```
alpha <- 8 + 22
beta <- 8 + 48
mean_true <- alpha / (alpha + beta)
var_true <- alpha * beta / ((alpha + beta)**2 * (alpha + beta + 1))

vec_theta <- rbeta(10000,alpha,beta)

vec_mean_dif <- c()
vec_var_dif <- c()


for(i in 1:10000){
  theta_used <- vec_theta[1:i]
  mean_used <- mean(theta_used)
  var_used <- var(theta_used)
  vec_mean_dif[i] <- mean_used - mean_true
  vec_var_dif[i] <- var_used - var_true
}

x <- 1:10000
plot(x,vec_mean_dif,type = "l",col = "red",ylim = c(-0.01,0.02) ,xlab = "number of theta", ylab = "the 
```




```
## integer(0)
```

From the plot we can see that with the number of random draws increasing, the differences converge into zeros in the end.

b)Draw 10000 random values from the posterior to compute the posterior probability Pr and compare with the exact value from the Beta psoterior.

```
prob <- length(which(vec_theta < 0.3)) / 10000
prob_2 <- pbeta(0.3,alpha,beta)
prob
```

```
## [1] 0.1752
```

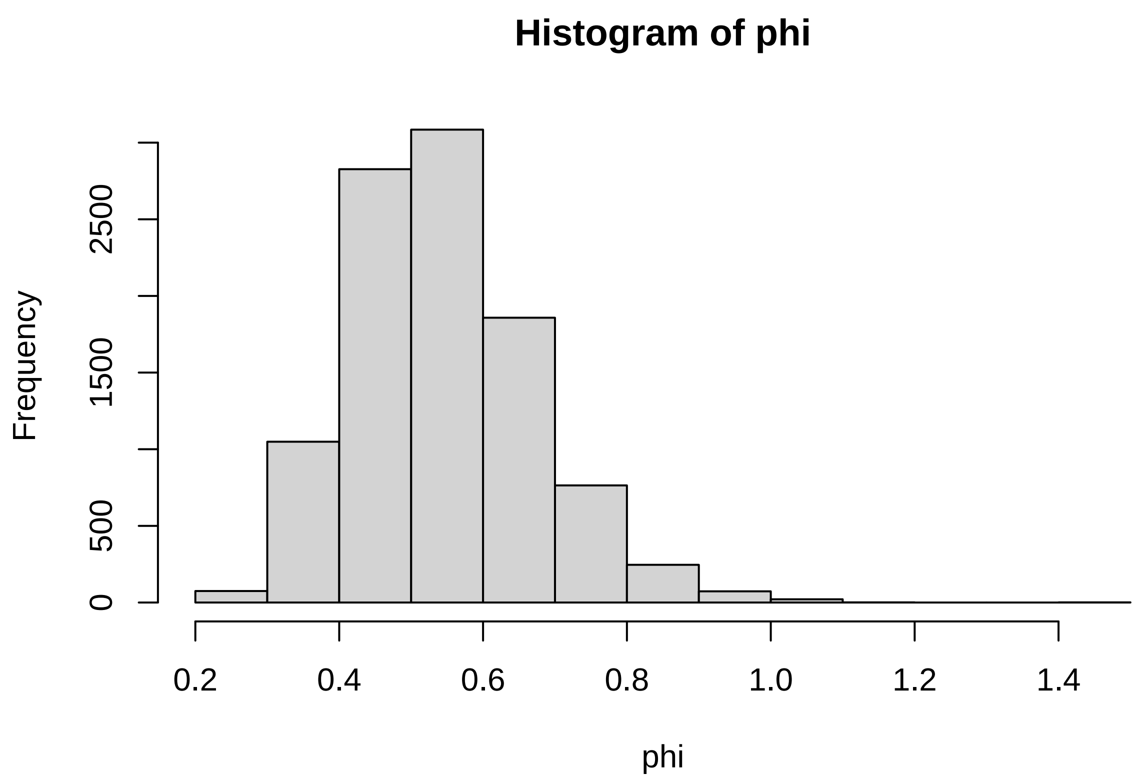
```
prob_2
```

```
## [1] 0.1714064
```

By comparison, we can see the difference is pretty small between two values.

c)Draw 10000 random values from the posterior of the odds phi, by using the previous random draws from the Beta psoterior for theta and plot the posterior distribution of phi.

```
phi <- vec_theta / (1-vec_theta)
hist(phi)
```



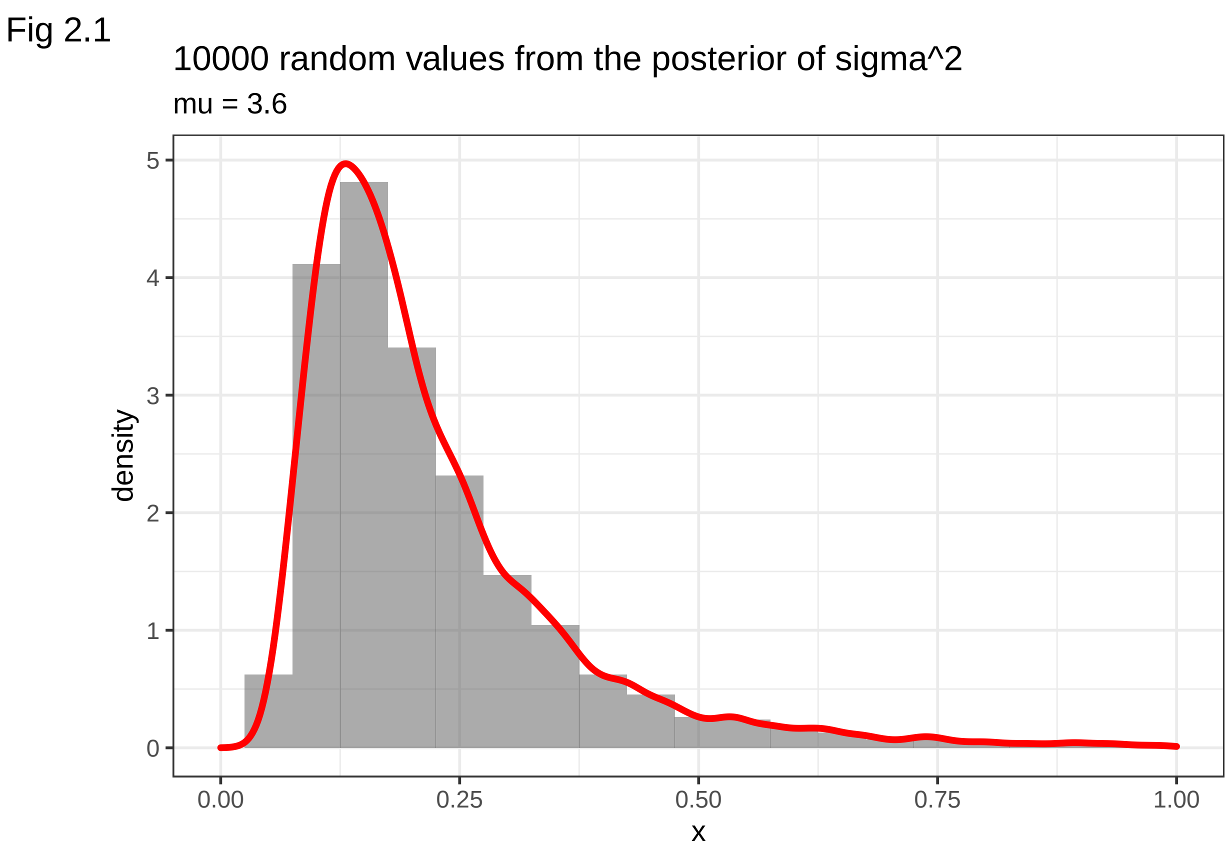

```
density(phi)
```

```
##
## Call:
##  density.default(x = phi)
##
## Data: phi (10000 obs.);  Bandwidth 'bw' = 0.01743
##
##        x                y
##  Min.   :0.1743   Min.   :0.000000
##  1st Qu.:0.4990   1st Qu.:0.001572
##  Median :0.8237   Median :0.105793
##  Mean   :0.8237   Mean   :0.769152
##  3rd Qu.:1.1484   3rd Qu.:1.308972
##  Max.   :1.4731   Max.   :3.258410
```

Assignment 2

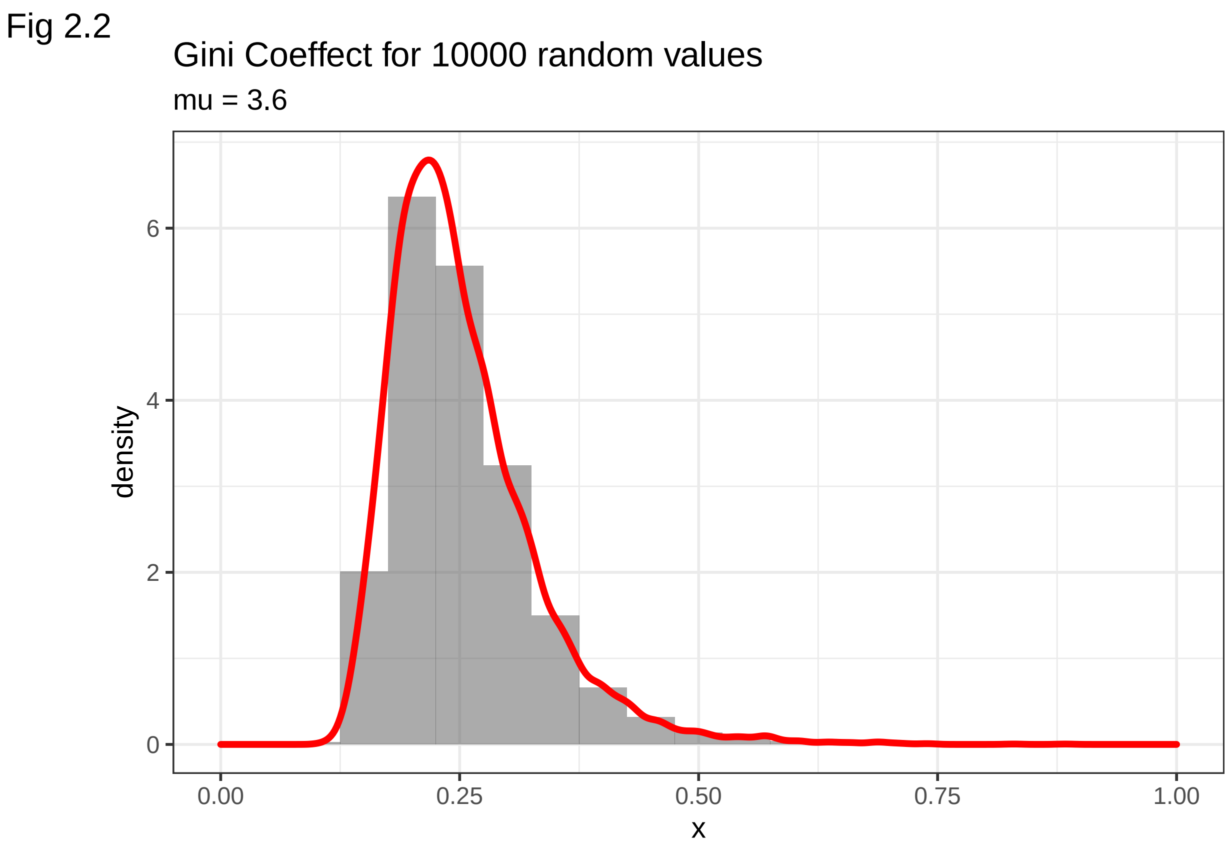
a) Draw 10000 random values from the posterior of σ^2 by assuming $\mu = 3.6$ and plot the posterior distribution.

```
# tau^2 function
tau_2=function(y,mu){
  return(sum((log(y)-mu)^2)/(length(y)))
}

y=c(33,24,48,32,55,74,23)
n=10000
mu=3.6
tau2=tau_2(y,mu)

randomPosterior=rinvchisq(n, length(y),tau2)

ggplot(data=data.frame(x=randomPosterior),aes(x))+
  geom_histogram(aes(y=..density..),binwidth=0.05,alpha=0.5) +
  geom_density(color='red',size=1.2) +
  scale_x_continuous(limits = c(0, 1))+labs(title='10000 random values from the posterior of sigma^2',
                                            subtitle = 'mu = 3.6',tag='Fig 2.1')+
  theme_bw()
```



b) Compute the Gini coeffect

```
Gini=2*pnorm(sqrt(randomPosterior)/sqrt(2))-1

ggplot(data=data.frame(x=Gini),aes(x))+
  geom_histogram(aes(y=..density..),binwidth=0.05,alpha=0.5) +
  geom_density(color='red',size=1.2)  +
  scale_x_continuous(limits = c(0, 1))+labs(title='Gini Coeffect for 10000 random values',
                                            subtitle = 'mu = 3.6',tag='Fig 2.2')+
  theme_bw()
```



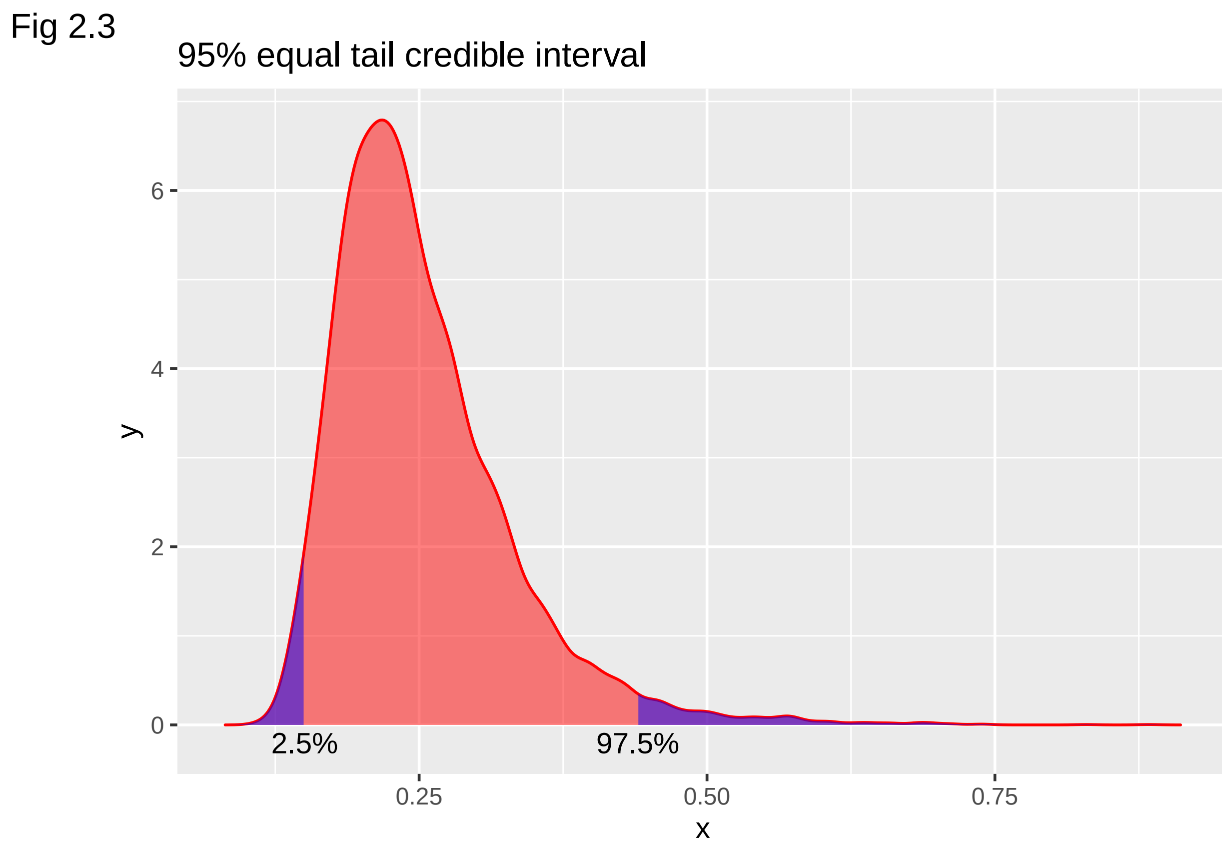
c) 95% equal tail credible interval for G

```
etcInterval=quantile(Gini,c(0.025,0.975))
etcInterval
```

```
##      2.5%     97.5%
## 0.1505392 0.4398634
```

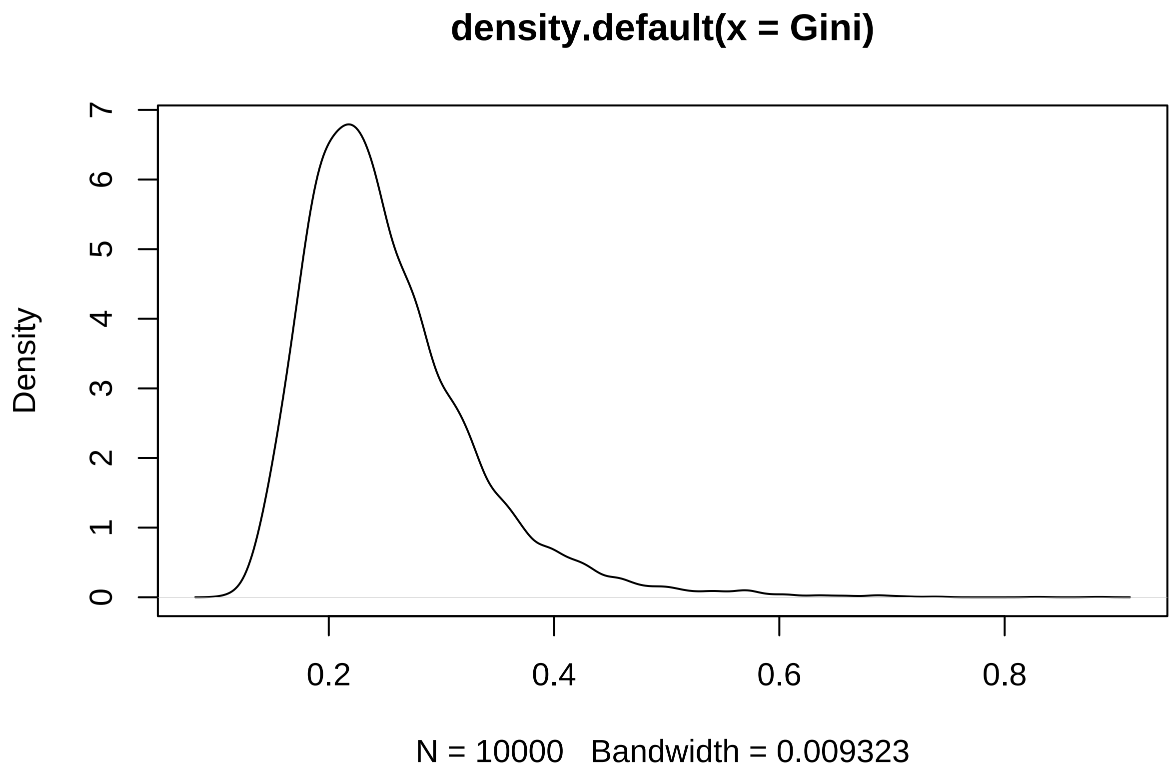
Plot

```
df=data.frame(x=Gini)
dat=with(density(df$x),data.frame(x,y))
dat1=dat%>%filter(x<etcInterval[1])
dat2=dat%>%filter(x>etcInterval[2])
ggplot()+
  geom_density(data=df,aes(x=x),,color='red',fill="red",alpha=0.5)+
  geom_area(data=dat1,aes(x=x,y=y),fill="blue",alpha=0.5)+
  geom_area(data=dat2,aes(x=x,y=y),fill="blue",alpha=0.5)+
  annotate(geom='text',x=etcInterval[1],y=-0.2,label='2.5%')+
  annotate(geom='text',x=etcInterval[2],y=-0.2,label='97.5%')+
  labs(title="95% equal tail credible interval",tag='Fig 2.3')
```



d) 95% Highest Posterior Density Interval (HPDI) for G

```
dgini=density(Gini)
plot(dgini)
```




```
sortDgini=sort(dgini$y,decreasing = TRUE)

sumDgini=sum(dgini$y)
cumsum=0
i=1
while(cumsum<0.95){
  cumsum=cumsum+(sortDgini[i]/sumDgini)
  i=i+1
}
cat('The highest density in the HPDI:',sortDgini[i])
```

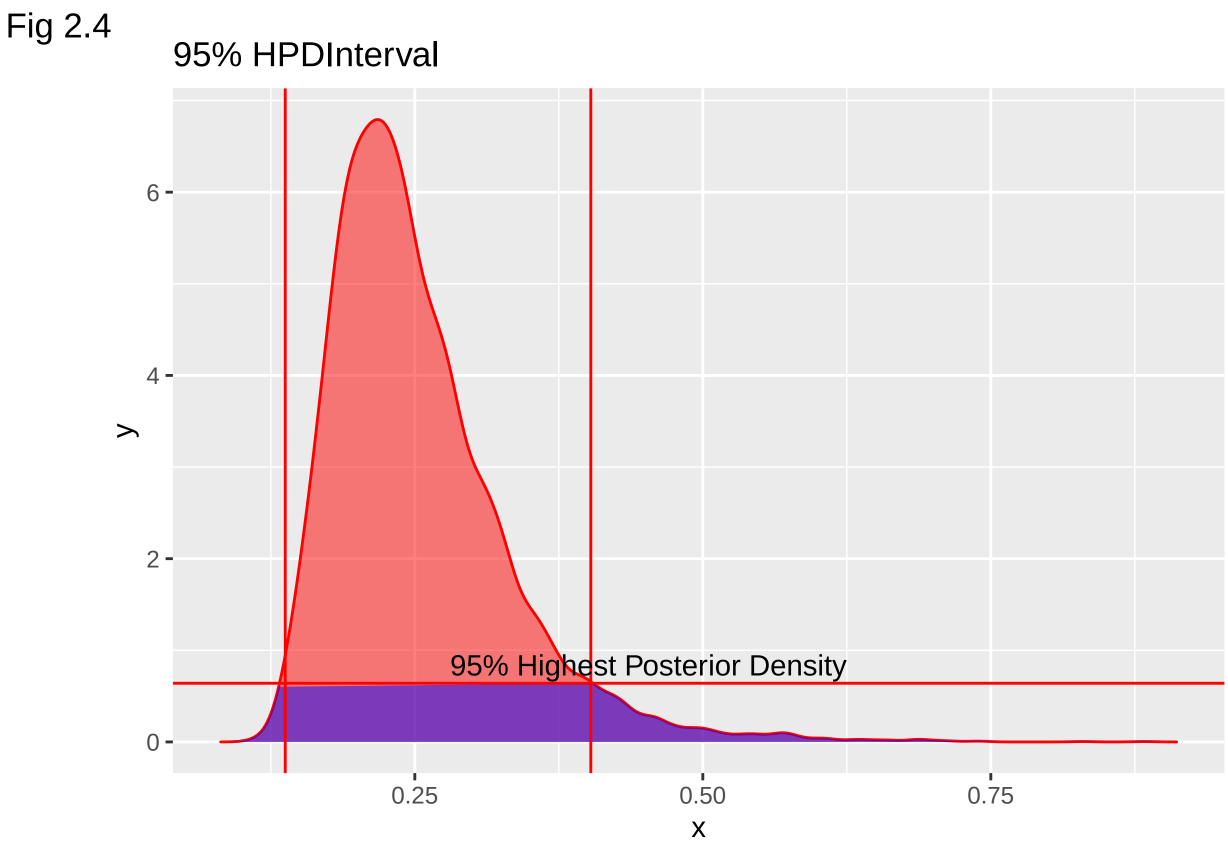
```
## The highest density in the HPDI: 0.6404965
```

```
HPDI=HPDinterval(as.mcmc(Gini),prob=0.95)
HPDI
```

```
##          lower     upper
## var1 0.1375105 0.4027781
## attr(,"Probability")
## [1] 0.95
```

Plot

```
datInterval=dgini$x[dgini$y<=sortDgini[i]]


df=data.frame(x=Gini)
dat=with(density(df$x),data.frame(x,y))
dat1=dat%>%filter(x%in%datInterval)
ggplot()+
  geom_density(data=df,aes(x=x),,color='red',fill="red",alpha=0.5)+
  geom_area(data=dat1,aes(x=x,y=y),fill="blue",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 = sortDgini[i],color='red')+
  annotate(geom='text',x=HPDI[2]+0.05,y=sortDgini[i]+0.2,label='95% Highest Posterior Density')
```



Compare the 95% equal tail credible interval and Highest Posterior Density Interval

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

```
##                                  lower     upper
## equal tail credible interval 0.1505392 0.4398634
## HPDI                         0.1375105 0.4027781
```

Choosing the narrowest interval, which for a unimodal distribution will involve choosing those values of highest probability density including the mode (the maximum a posteriori). This is sometimes called the highest posterior density interval (HPDI).

Choosing the interval where the probability of being below the interval is as likely as being above it. This interval will include the median. This is sometimes called the equal-tailed interval.

(https://en.wikipedia.org/wiki/Credible_interval)

Assignment 3

a) Derive the posterior

Prior: $K \sim Exponential(\lambda)$

Hence,

$$p(K) = \begin{cases} \lambda & K \geq 0 \\ 0 & K < 0 \end{cases}$$

Likelihood:

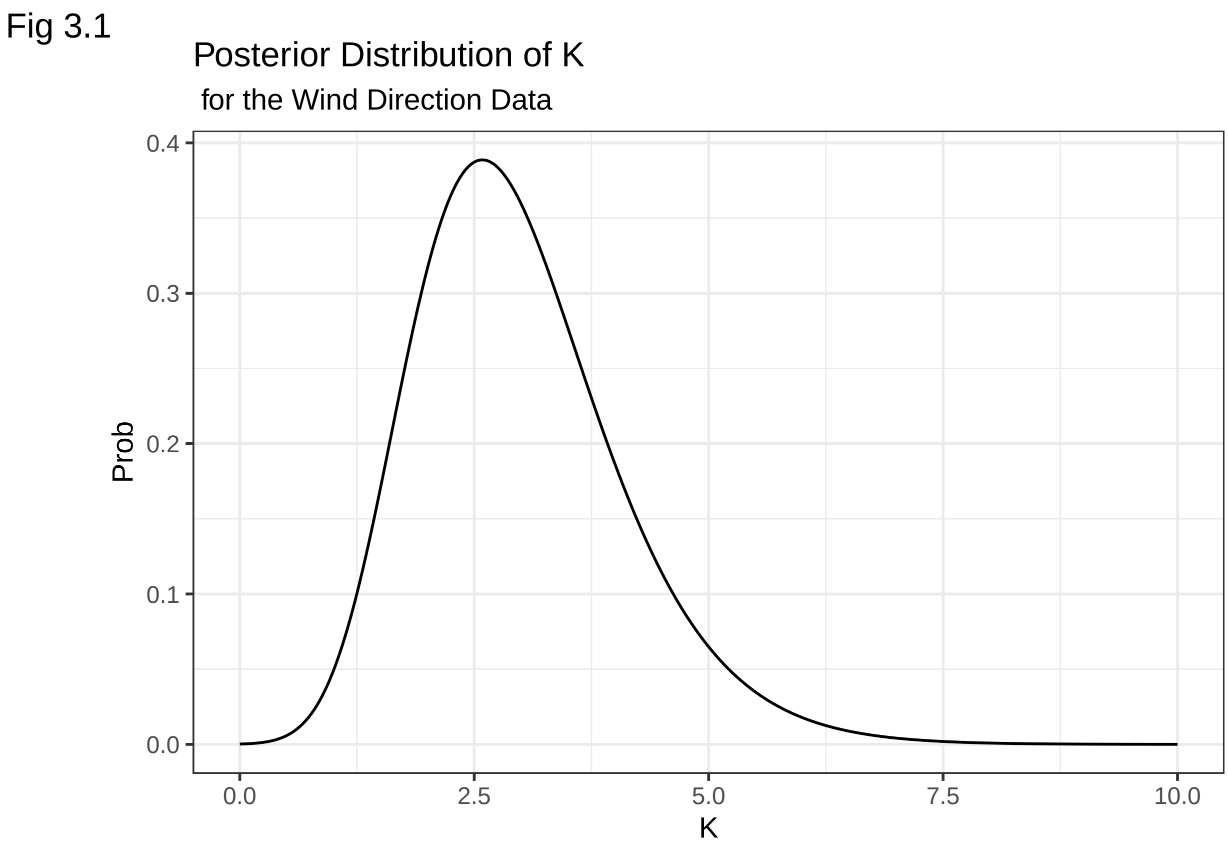
$p(y|\mu, k) = p(y_1, \ldots, y_n|\mu, k) = \frac{\exp[K \cdot \sum_{i=1}^{n} \cos(y_i - \mu)]}{(2\pi I_0(K))^n}$

Posterior:

$p(k|y, \mu) \propto p(y|\mu, k) \cdot (K) \propto \frac{\exp[K \cdot (\sum_{i=1}^{n} \cos(y_i - \mu) - \lambda)]}{(2\pi I_0(K))^n}$

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

# Normalize

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
lam=0.5
```

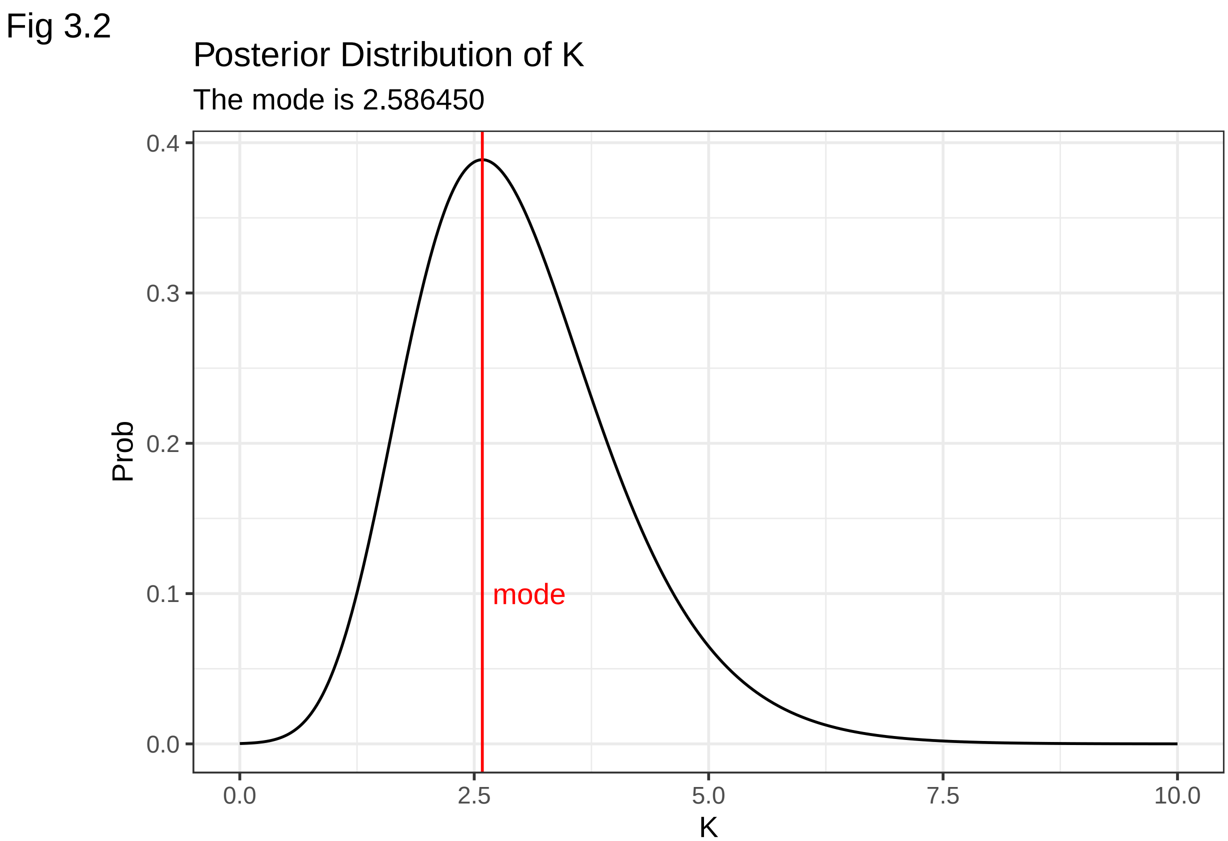
```
prob=sapply(seq(0,10,0.01), posteriorK,y=y,mu=mu,lam=lam)
prob=prob/sum(prob)*100
plotData=data.frame(x=seq(0,10,0.01),y=prob)

ggplot(data=plotData)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=' for the Wind Direction Data',tag = 'Fig 3.1',
       x='K',y='Prob')+
  theme_bw()
```



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

```
optimPosteriorK=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
  lam=0.5
  cumprod=1
  for(i in 1:length(y)){
    cumprod=cumprod*(exp(k*cos(y[i]-mu))/(2*pi*besselI(k,0)))
  }
  p=cumprod*lam*exp(-lam*k)
  return(-p)
}
mode=optim(par=2.5,fn=optimPosteriorK,method=c('Brent'),lower=0,upper=10)
```

```
ggplot(data=plotData)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=sprintf('The mode is %f',mode$par),tag = 'Fig 3.2',
       x='K',y='Prob')+
  geom_vline(xintercept = mode$par,color='red')+
  annotate(geom='text',x=mode$par+0.5,y=0.1,label='mode',color='red')+
  theme_bw()
```



Appendix

```
library(extraDistr)
library(ggplot2)
library(dplyr)
library(coda)
alpha <- 8 + 22
beta <- 8 + 48
mean_true <- alpha / (alpha + beta)
var_true <- alpha * beta / ((alpha + beta)**2 * (alpha + beta + 1))

vec_theta <- rbeta(10000,alpha,beta)

vec_mean_dif <- c()
vec_var_dif <- c()


for(i in 1:10000){
  theta_used <- vec_theta[1:i]
  mean_used <- mean(theta_used)
  var_used <- var(theta_used)
  vec_mean_dif[i] <- mean_used - mean_true
  vec_var_dif[i] <- var_used - var_true
}

x <- 1:10000
plot(x,vec_mean_dif,type = "l",col = "red",ylim = c(-0.01,0.02) ,xlab = "number of theta", ylab = "the d

prob <- length(which(vec_theta < 0.3)) / 10000
prob_2 <- pbeta(0.3,alpha,beta)
prob
prob_2
phi <- vec_theta / (1-vec_theta)
hist(phi)
density(phi)
# tau^2 function
tau_2=function(y,mu){
  return(sum((log(y)-mu)^2)/(length(y)))
}

y=c(33,24,48,32,55,74,23)
n=10000
mu=3.6
tau2=tau_2(y,mu)

randomPosterior=rinvchisq(n, length(y),tau2)

ggplot(data=data.frame(x=randomPosterior),aes(x))+
  geom_histogram(aes(y=..density..),binwidth=0.05,alpha=0.5) +
  geom_density(color='red',size=1.2) +
  scale_x_continuous(limits = c(0, 1))+labs(title='10000 random values from the posterior of sigma^2',
                                            subtitle = 'mu = 3.6',tag='Fig 2.1')+
  theme_bw()
```

```
Gini=2*pnorm(sqrt(randomPosterior)/sqrt(2))-1

ggplot(data=data.frame(x=Gini),aes(x))+
  geom_histogram(aes(y=..density..),binwidth=0.05,alpha=0.5) +
  geom_density(color='red',size=1.2)  +
  scale_x_continuous(limits = c(0, 1))+labs(title='Gini Coeffect for 10000 random values',
                                            subtitle = 'mu = 3.6',tag='Fig 2.2')+
  theme_bw()


etcInterval=quantile(Gini,c(0.025,0.975))
etcInterval
df=data.frame(x=Gini)
dat=with(density(df$x),data.frame(x,y))
dat1=dat%>%filter(x<etcInterval[1])
dat2=dat%>%filter(x>etcInterval[2])
ggplot()+
  geom_density(data=df,aes(x=x),,color='red',fill="red",alpha=0.5)+
  geom_area(data=dat1,aes(x=x,y=y),fill="blue",alpha=0.5)+
  geom_area(data=dat2,aes(x=x,y=y),fill="blue",alpha=0.5)+
  annotate(geom='text',x=etcInterval[1],y=-0.2,label='2.5%')+
  annotate(geom='text',x=etcInterval[2],y=-0.2,label='97.5%')+
  labs(title="95% equal tail credible interval",tag='Fig 2.3')

dgini=density(Gini)
plot(dgini)

sortDgini=sort(dgini$y,decreasing = TRUE)

sumDgini=sum(dgini$y)
cumsum=0
i=1
while(cumsum<0.95){
  cumsum=cumsum+(sortDgini[i]/sumDgini)
  i=i+1
}
cat('The highest density in the HPDI:',sortDgini[i])


HPDI=HPDinterval(as.mcmc(Gini),prob=0.95)
HPDI
datInterval=dgini$x[dgini$y<=sortDgini[i]]


df=data.frame(x=Gini)
dat=with(density(df$x),data.frame(x,y))
dat1=dat%>%filter(x%in%datInterval)
ggplot()+
  geom_density(data=df,aes(x=x),,color='red',fill="red",alpha=0.5)+
  geom_area(data=dat1,aes(x=x,y=y),fill="blue",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')+
```

```r
  geom_hline(yintercept = sortDgini[i],color='red')+
  annotate(geom='text',x=HPDI[2]+0.05,y=sortDgini[i]+0.2,label='95% Highest Posterior Density')

Interval=data.frame("lower"=c(etcInterval[1],HPDI[1]),"upper"=c(etcInterval[2],HPDI[2]))
rownames(Interval)=c("equal tail credible interval","HPDI")
Interval
#posterior function
posteriorK=function(k,y,mu,lam){
  cumprod=1
  for(i in 1:length(y)){
    cumprod=cumprod*(exp(k*cos(y[i]-mu))/(2*pi*besselI(k,0)))
  }
  p=cumprod*lam*exp(-lam*k)
  return(p)
}

# Normalize

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
lam=0.5

prob=sapply(seq(0,10,0.01), posteriorK,y=y,mu=mu,lam=lam)
prob=prob/sum(prob)*100
plotData=data.frame(x=seq(0,10,0.01),y=prob)

ggplot(data=plotData)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=' for the Wind Direction Data',tag = 'Fig 3.1',
       x='K',y='Prob')+
  theme_bw()

optimPosteriorK=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
  lam=0.5
  cumprod=1
  for(i in 1:length(y)){
    cumprod=cumprod*(exp(k*cos(y[i]-mu))/(2*pi*besselI(k,0)))
  }
  p=cumprod*lam*exp(-lam*k)
  return(-p)
}
mode=optim(par=2.5,fn=optimPosteriorK,method=c('Brent'),lower=0,upper=10)

ggplot(data=plotData)+geom_line(aes(x=x,y=y))+
  labs(title='Posterior Distribution of K',subtitle=sprintf('The mode is %f',mode$par),tag = 'Fig 3.2',
       x='K',y='Prob')+
  geom_vline(xintercept = mode$par,color='red')+
  annotate(geom='text',x=mode$par+0.5,y=0.1,label='mode',color='red')+
  theme_bw()
```