

Bayesian Learning_Lab1

Zahra Jalilpour

2021-04-01

Question1: Daniel Bernouli

Exercise: Let $y_1, \dots, y_n | \theta \sim \text{Bern}(\theta)$, and assume that you have obtained a sample with $s = 8$ successes in $n = 24$ trials. Assume a $\text{Beta}(\alpha_0, \beta_0)$ prior for θ and let $\alpha_0 = \beta_0 = 3$

a) Draw random numbers from the posterior $\theta | y \sim \text{Beta}(\alpha_0 + s, \beta_0 + f)$, where $y = (y_1, \dots, y_n)$ and verify graphically that the posterior mean and standard deviation converges to the true values as the number of randoms grows large.

- PDF for bernouli distribution without order, n fixed, s random:

$$p(s|\theta) = \binom{n}{s} \theta^s \cdot (1 - \theta)^{n-s}, \quad s \text{ is the number of success and } n \text{ is the number of trails}$$

- PDF for Beta distribution :

$$f(\theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} \theta^{\alpha-1} \cdot (1 - \theta)^{\beta-1}$$

- Mean in Beta distribution:

$$\text{Mean} : \frac{\alpha}{\alpha + \beta}$$

- Standard deviation:

$$SD : \sqrt{\frac{\alpha \cdot \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}}$$

- Likelihood:

$$p(s|\theta) = \binom{n}{s} \theta^s \cdot (1 - \theta)^{n-s}$$

$$\text{Prior} \propto \theta^{\alpha-1} \cdot (1 - \theta)^{\beta-1}$$

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

$$\text{Posterior} \propto \text{Beta}(\alpha_0 + s, \beta_0 + f)$$

```

n<-24
s<-8
f<-(n-s)
alpha_0 <- 3
beta_0 <-3
alpha<- alpha_0 + s
beta<- beta_0 + f
mean_posterior<- alpha/(alpha+beta)
sd_posterior <- sqrt((alpha*beta)/
                    ((alpha+beta)^2*(alpha + beta +1)))
paste("mean of posterior is :",mean_posterior)

```

```
## [1] "mean of posterior is : 0.366666666666667"
```

```
paste("standard deviation of posterior is: ", sd_posterior)
```

```
## [1] "standard deviation of posterior is: 0.0865507910219387"
```

For different random numbers from the posterior:

```

beta_stat<-function(n,alpha,beta){
  rand_gen <-rbeta(n, shape1 = alpha, shape2 = beta)
  out<- c(i=n, rand_mean=mean(rand_gen),rand_sd=sd(rand_gen))
}

```

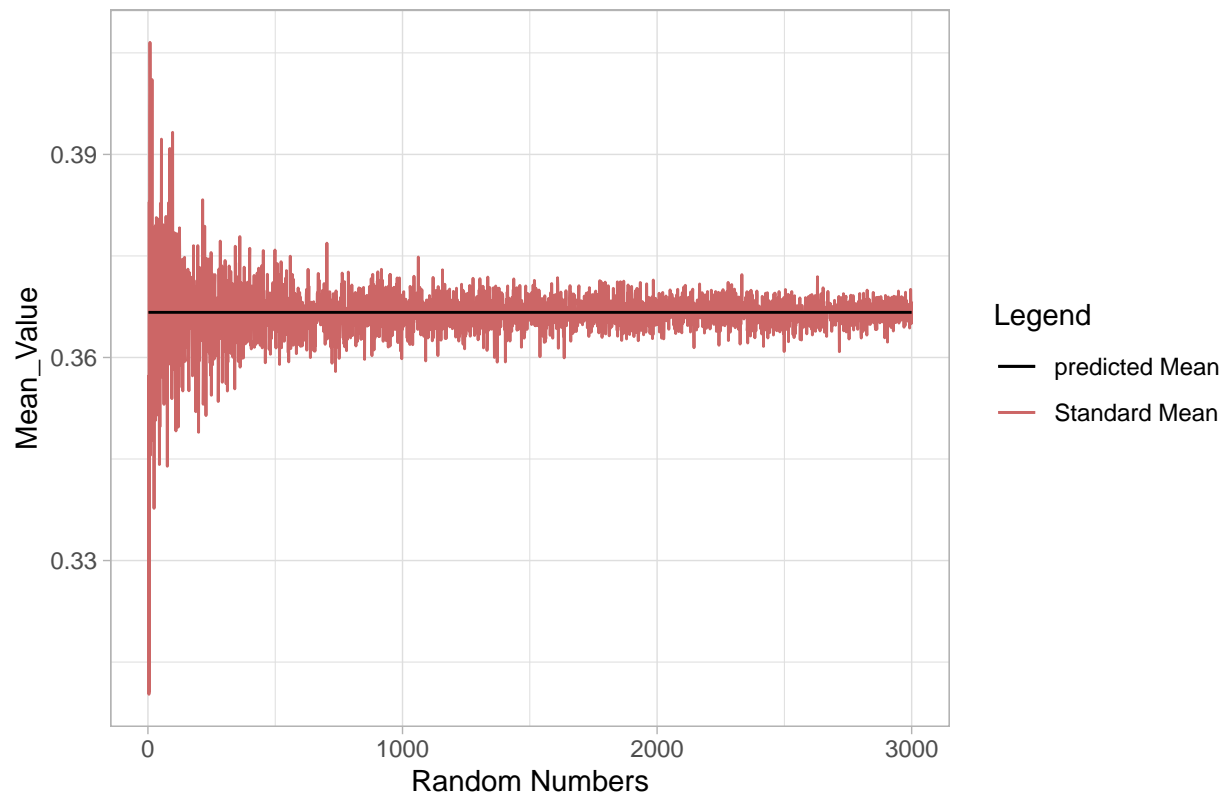
```
df1 = data.frame(t(sapply(2:3000, beta_stat, alpha, beta)))
```

```

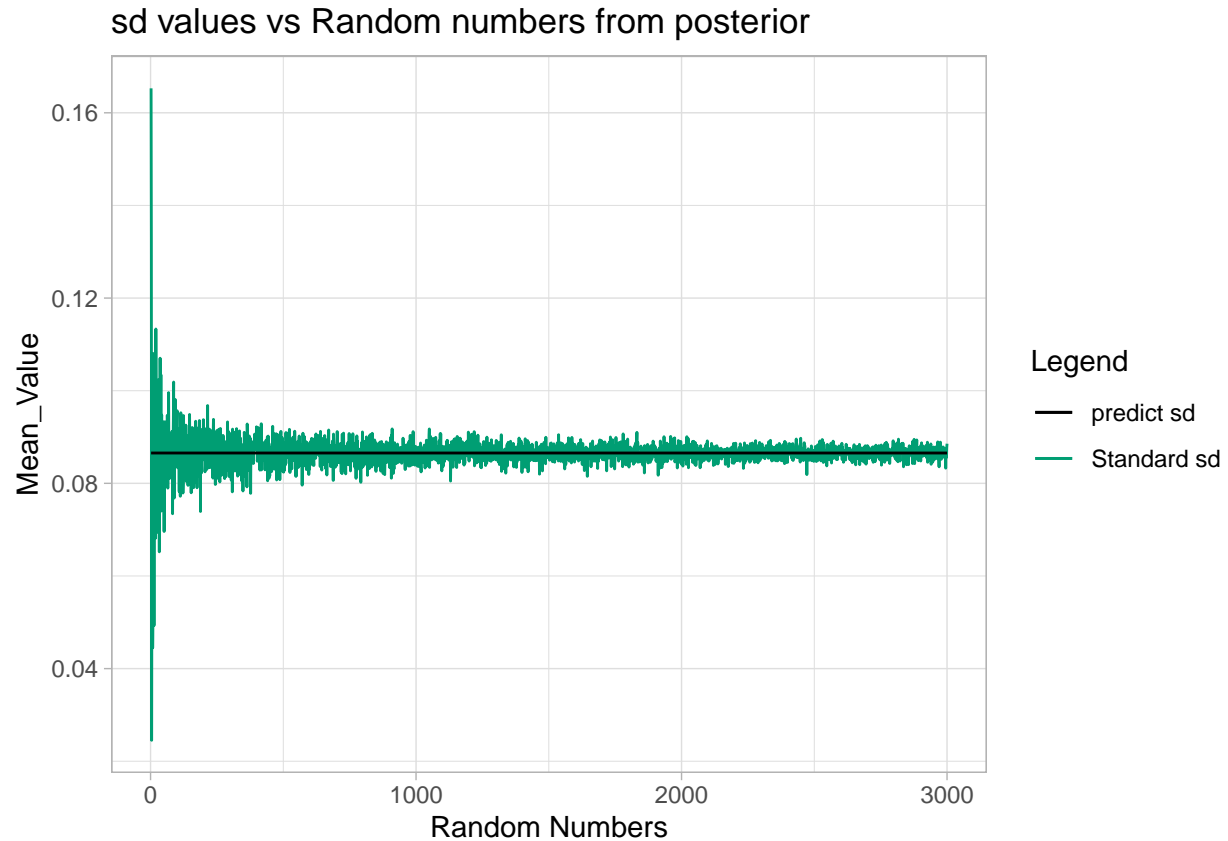
library(ggplot2)
ggplot(df1) +
  geom_line(aes(x = i, y = rand_mean, color = "Standard Mean")) +
  geom_line(aes(x = i, y = mean_posterior, color = "predicted Mean")) +
  labs(title = "Mean values vs Random numbers from posterior", y = "Mean_Value", x = "Random Numbers") +
  scale_color_manual("Legend", values = c("#000000", "#CC6666")) +
  theme_light()

```

Mean values vs Random numbers from posterior



```
ggplot(df1) +  
  geom_line(aes(x = i, y = rand_sd, color = "Standard sd")) +  
  geom_line(aes(x = i, y = sd_posterior, color = "predict sd")) +  
  labs(title = "sd values vs Random numbers from posterior", y = "Mean_Value", x = "Random Numbers") +  
  scale_color_manual("Legend", values = c("#000000", "#009E73")) +  
  theme_light()
```



As we see, by increasing the random numbers drawn from posterior, mean and standard deviation converge to the true value.

b) Use `simulation(nDraws=10000)`, to compute the posterior probability $\Pr(\theta > 0.4|y)$ and compare with the exact value [Hint:`pbeta()`]

To calculate the exact value of probability , we can use `pbeta` function.

```
true_probability<- pbeta(q=0.4,shape1 = alpha, shape2 = beta, lower.tail = TRUE)
paste("True value of probability is :",true_probability )
```

```
## [1] "True value of probability is : 0.657334638118954"
```

```
paste("The calculated value of probability is:",mean(rbeta(10000,alpha,beta)<0.4))
```

```
## [1] "The calculated value of probability is: 0.6581"
```

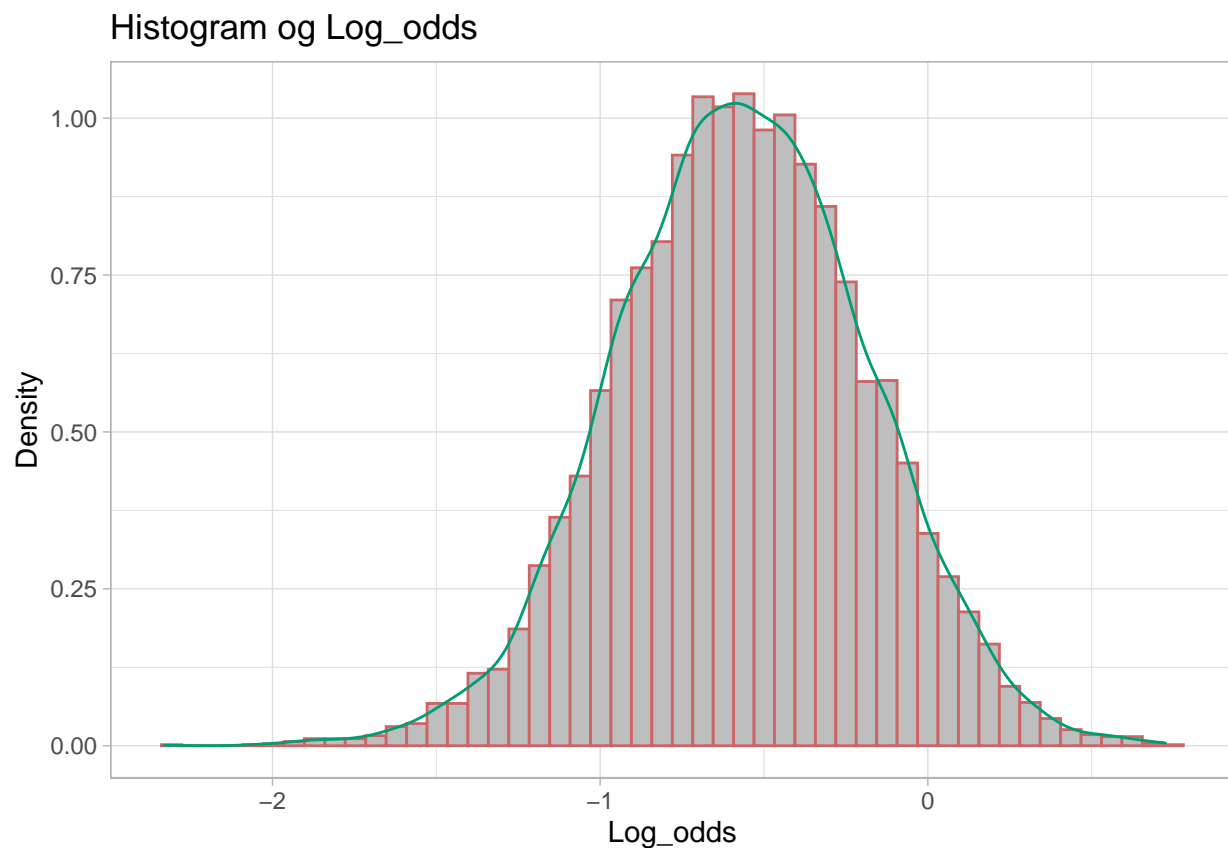
c) Compute the posterior distribution of the log-odds, $\Phi = \log \frac{\theta}{1-\theta}$ by `simulation(nDraws=10000)`

Here θ is the value drawn from posterior.

```

n_draws<- 10000
round_num<-rbeta(10000,alpha,beta)
phi<- log(round_num/(1-round_num))
df2<- data.frame(phi)
colnames(df2)<- "phi"
ggplot(df2)+
  geom_histogram(aes(x=phi, y=..density..),color="#CC6666",
                 fill="grey",bins=50,position='identity')+
  geom_density(aes(x=phi, y=..density..),color="#009E73")+
  labs(title="Histogram og Log_odds",
       y="Density",
       x="Log_odds",color="Legend")+
  theme_light()

```



Question2:Log-normal distribution and the Gini coefficient.

Assume that you have asked 10 randomly selected persons about their monthly income(in thousands Swedish Krona) and obtained the following ten observations:38,20,49,58,31,70,18,56,25 and 78.A common model for non-negative continuous variables is the log-normal distribution. The log normal distribution $\text{logN}(\mu, \sigma^2)$ has density function

$$p(y|\mu, \sigma^2) = \frac{1}{y \cdot \sqrt{2\pi\sigma^2}} e^{-\frac{(\log y - \mu)^2}{2\sigma^2}}$$

where $y > 0$, $\mu > 0$ and $\sigma^2 > 0$. The log-normal distribution is related to the normal distribution as follows: if $y \sim \text{logN}(\mu, \sigma^2)$ then $\log y \sim N(\mu, \sigma^2)$. Let $y_1, \dots, y_n | \mu, \sigma^2 \sim \text{logN}(\mu, \sigma^2)$, where $\mu = 3.8$ is assumed to be known but σ^2 is unknown with non-informative prior $p(\sigma^2) \propto 1/\sigma^2$. The posterior for σ^2 is the $\text{Inv-}\chi^2(n, \tau^2)$ distribution where

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

a) Simulate 10,000 draws from the posterior of σ^2 (assuming $\mu = 3.8$) and compare with the theoretical $\text{Inv-}\chi^2(n, \tau^2)$ posterior distribution.

The scaled inverse chi squared distribution is the distribution for $x = 1/s^2$ where s^2 is a sample mean of the squares of v independent normal random variables that have mean 0 and inverse variance $1/\sigma^2$. The scaled inverse chi square distribution has extra parameter τ^2 , which scales the distribution horizontally and vertically. The two distribution of inverse chi square and the scaled inverse chi square have the relation as below:

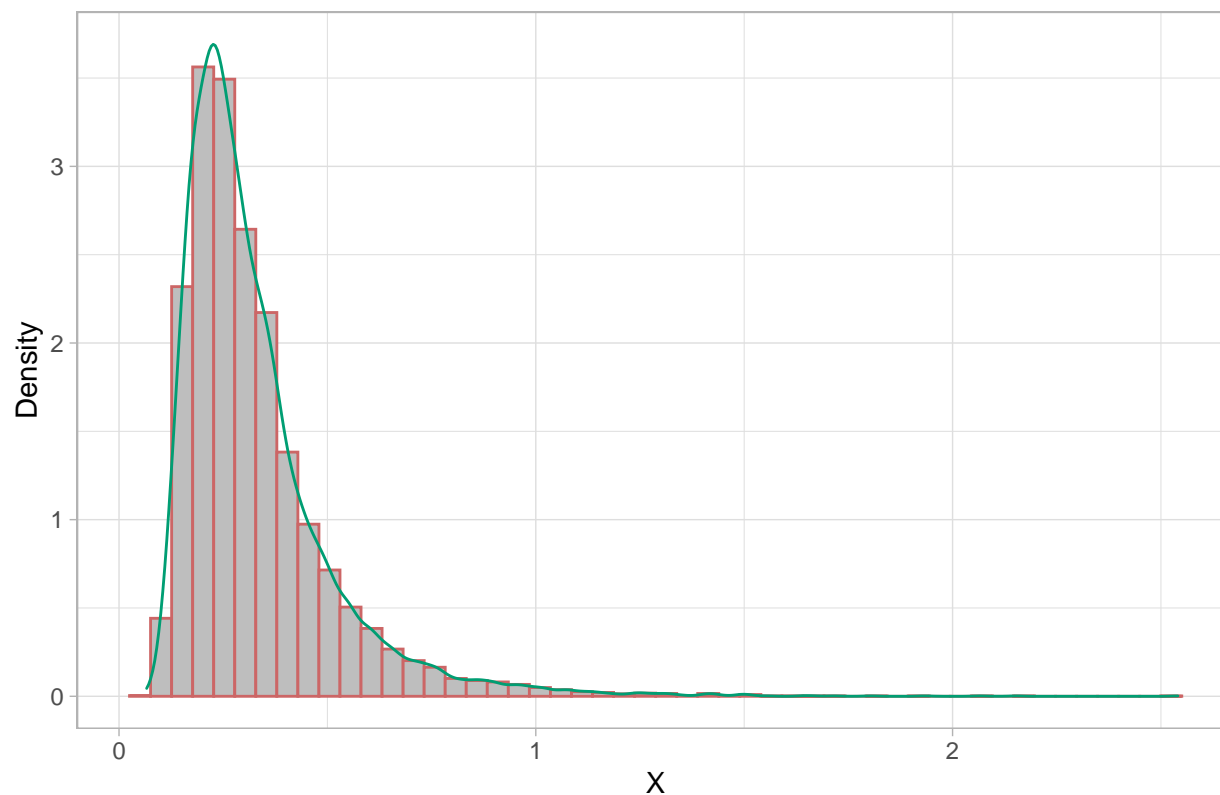
$X \sim \text{Scale-Inv-}\chi^2(n, \tau^2)$, then

$$\frac{X}{n\tau^2} \sim \text{Inv-}\chi^2(n)$$

Hence, at first we draw (nDraws) by $X \sim \chi^2(n)$, then we should calculate $\sigma^2 = \frac{n\tau^2}{X}$, in which we now how to achieve the τ^2 . After that we can calculate the mean and Var of σ^2 , and then we compare these values with the theoretical $\text{Inv-}\chi^2(n, \tau^2)$ posterior distribution. The mean and Standard deviation of $\text{Inv-}\chi^2(n, \tau^2)$ are obtained by $\frac{n\tau^2}{n-2}$ for $n > 2$ and $\sqrt{\frac{2n^2\tau^4}{(n-2)^2(n-4)}}$ for $n > 4$

```
y<-c(38,20,49,58,31,70,18,56,25,78)
n<- length(y)
mu=3.8
tau_2<- sum((log(y)-mu)^2)/n
sigma_sq<- c()
for(i in 1:10000){
  x<-rchisq(n=1,df=n)
  sigma_sq[i]<- (n*tau_2)/x
}
library(ggplot2)
ggplot(as.data.frame(sigma_sq), aes(x = sigma_sq)) +
  geom_histogram(aes(y=..density..),colour="#CC6666",fill="grey",bins=50,position='identity') +
  geom_density(aes(x = sigma_sq),color="#009E73") +
  labs(title = "Drawn Samples and Inverse-Chi-Squared Density ",
       y = "Density", x = "X",color="Legend") +
  theme_light()
```

Drawn Samples and Inverse–Chi–Squared Density



Now we should compare the result of simulation with theoretical values of mean and Sd of posterior distribution.

```
sim_mean <- mean(sigma_sq)
sim_sd<- sd(sigma_sq)
theo_mean <-(n*tau_2)/(n-2)
theo_sd = sqrt((2*n^2*tau_2^2)/(((n-2)^2)*(n-4)))
sim_df<- data.frame(Posterior="Simulatin",Mean=sim_mean, Sd=sim_sd)
theo_df<-data.frame(Posterior="Theoretical",Mean=theo_mean, Sd=theo_sd)
main_df<-rbind(sim_df,theo_df)
knitr::kable(main_df)
```

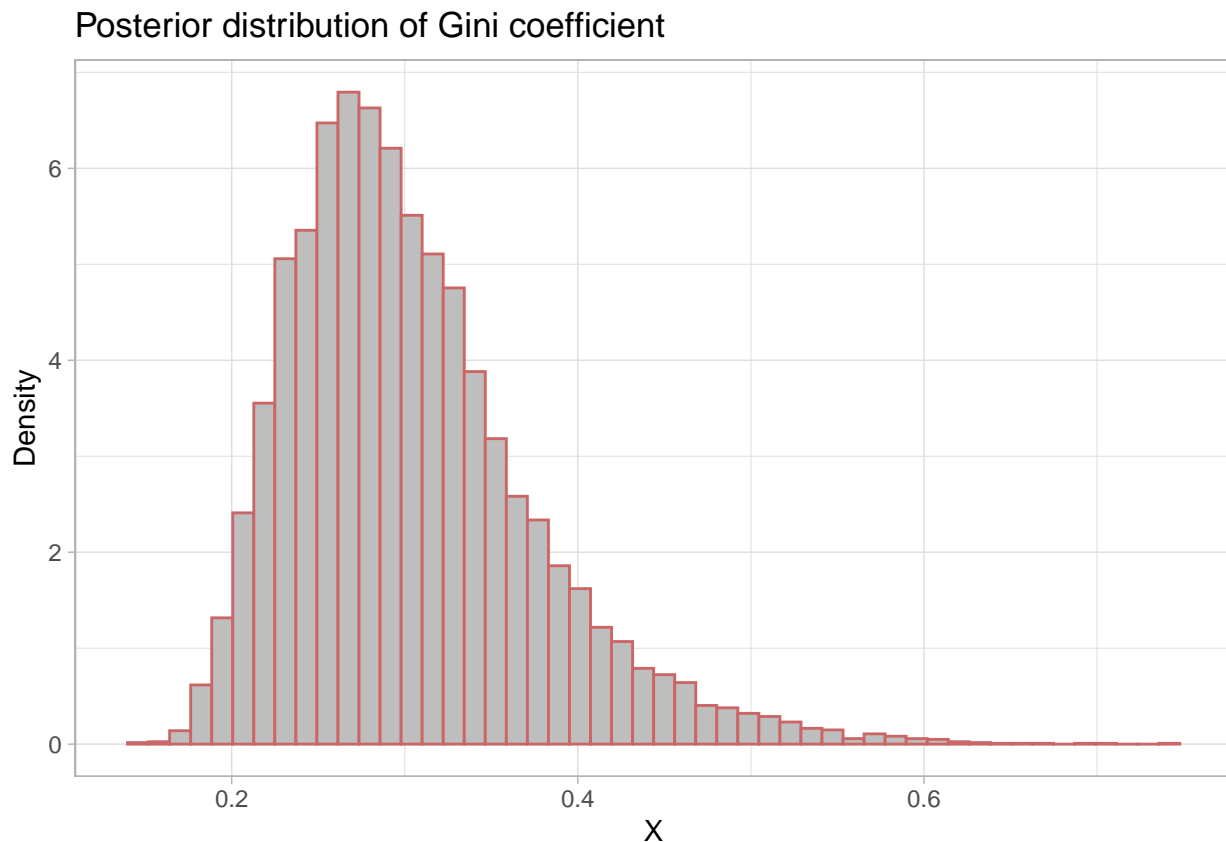
Posterior	Mean	Sd
Simulatin	0.3264036	0.1834342
Theoretical	0.3263046	0.1883920

b) The most common measure of income inequality is the Gini coefficient, G , where $0 \leq G \leq 1$. $G = 0$ means a completely equal income distribution, whereas $G = 1$ means complete income inequality (See Wikipedia for more information about the Gini coefficient). It can be shown that $G = 2\phi(\frac{\sigma}{\sqrt{2}}) - 1$ when incomes follow a $\log N(\mu, \sigma^2)$ distribution. $\phi(z)$ is the cumulative distribution function (CDF) for the standard normal distribution with mean zero and unit variance. Use the posterior draws in a) to compute the posterior distribution of the Gini coefficient G for the current data set.

As we know $\phi(z)$ is the CDF for the standard normal distribution with mean=0 and variance=1. In previous task we obtained the posterior distribution for σ^2 , so we use from that posterior draws in the $\phi(z)$ to compute the posterior distribution of the Gini Coefficient.

```
p_gini <- 2* pnorm(sqrt(sigma_sq)/sqrt(2), mean=0, sd=1)-1
G_df = data.frame(p_gini)
ggplot(G_df) +
  geom_histogram(aes(x = p_gini, y=..density..),
                 bins = 50, color = "#CC6666", fill = "gray", position='identity') +

  labs(title = "Posterior distribution of Gini coefficient ",
       y = "Density", x = "X") +
  theme_light()
```



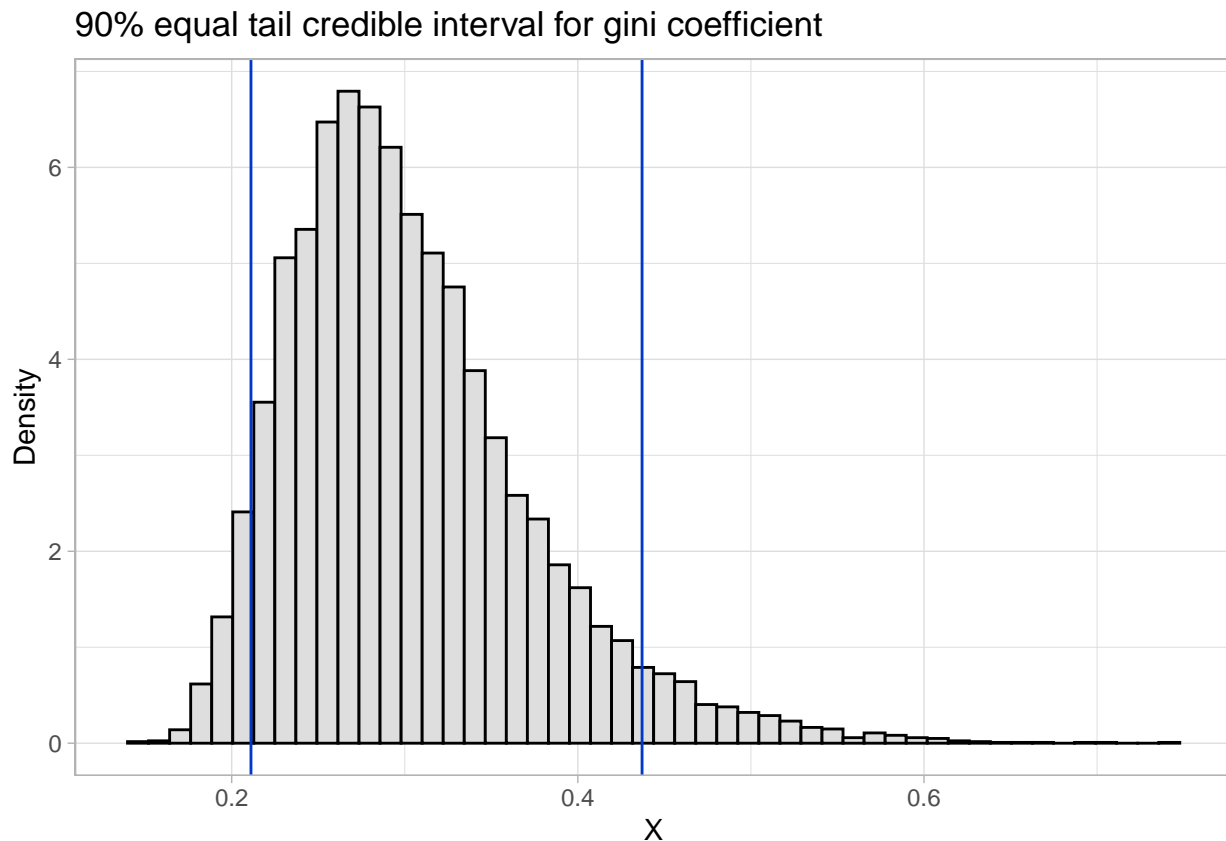
c) Use the posterior draws from b) to compute a 90% equal tail credible interval for G. An 90% equal tail interval (a,b) cuts off 5% percent of the posterior probability mass to the left of a, and 5% to the right of b. Also, do a kernel density estimate of the posterior of G using the density function in R with default settings, and use that kernel density estimate to compute a 90% Highest Posterior Density interval(HPDI) for G. Compare the two intervals.

At first we use `quantile()` function to obtain two tail 90% quantile.

```
quantiles = quantile(p_gini, c(0.05, 0.95))
cat("We can see the credible intervals", quantiles[1], quantiles[2])

## We can see the credible intervals 0.2111452 0.4371063

gini_df<- data.frame(p_gini)
ggplot(gini_df) +
  geom_histogram(aes(x = p_gini, y=..density..),
                 bins = 50, color = "black", fill = "#DEDEDE") +
  geom_vline(xintercept = quantiles, colour = "#0039C7") +
  labs(title = "90% equal tail credible interval for gini coefficient",
       y = "Density", x = "X") +
  theme_light()
```



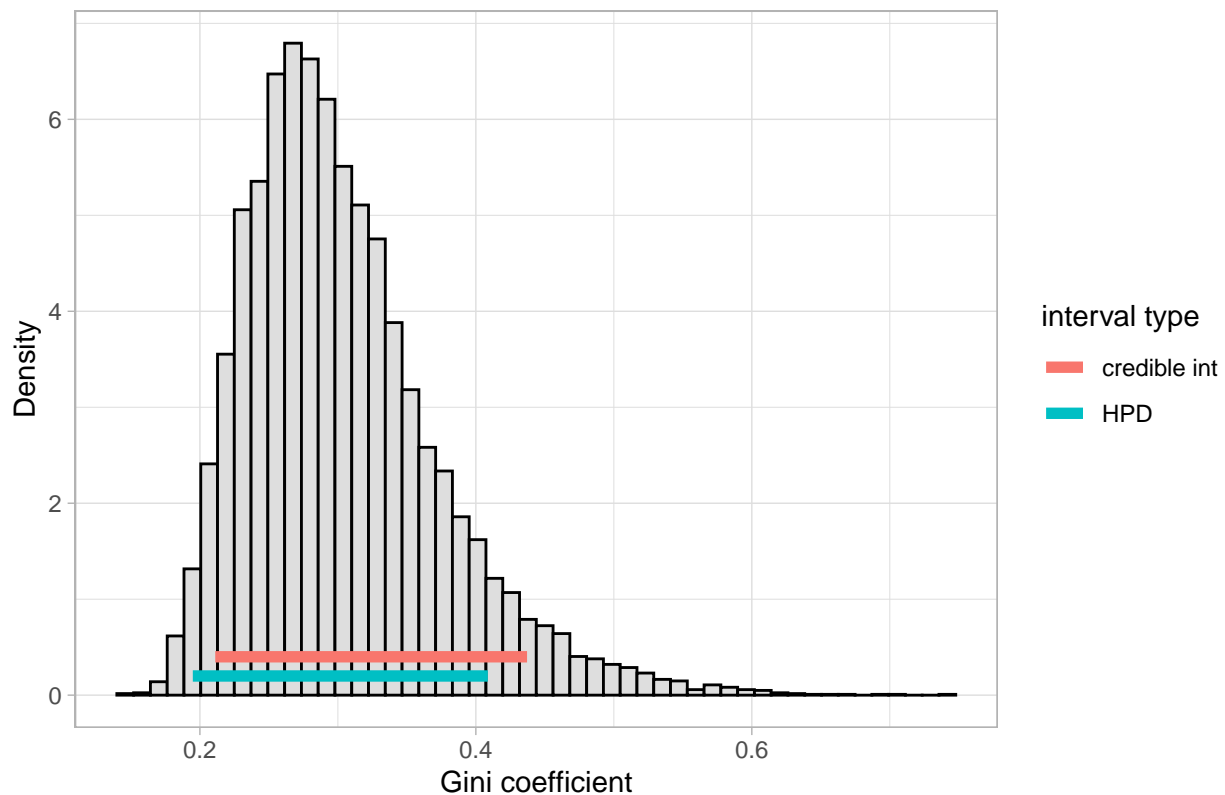
Now we should use kernel density estimate to compute a 90% Highest Posterior Density Interval(HPDI) for Gini coefficient. Here we use density from gini posterior, then take the density estimate for particular intervals, rank_order them cumulative sample until we reach our designed probability density(alpha=0.9)

```

temp<- density(p_gini)
kernel_df <- data.frame(x=temp$x, density_k= temp$y)
sorted_kernel <- kernel_df[order(kernel_df$density_k, decreasing = TRUE),]
sorted_kernel$cumulative <- cumsum(sorted_kernel$density_k)
alpha1<- sorted_kernel$cumulative[dim(sorted_kernel)[1]]*0.9
result_df <- sorted_kernel[which(sorted_kernel$cumulative < alpha1),]
lower_tail<-result_df[which.min(result_df$x),]$x
upper_tail<-result_df[which.max(result_df$x),]$x
HPD<- c(lower_tail,upper_tail)
df <- data.frame(x=c(quantiles[1],quantiles[2],lower_tail,upper_tail), y=c(0.4,0.4,0.2,0.2),label=c("cr",
ggplot() +
  geom_histogram(aes(x = p_gini, y=..density..),
                 bins = 50, color = "black", fill = "#DEDEDE") +
  geom_line(mapping=aes(df$x, y=df$y, color=df$label),size=2) +
  labs(title = "90% Highest Posterior Density Interval",
       y = "Density", x = "Gini coefficient", color="interval type") +
  theme_light()

```

90% Highest Posterior Density Interval



```

equal_df<- data.frame(Interval="Equal tailed interval",Lower=quantiles[1], Upper=quantiles[2])
hpd_df<-data.frame(Interval="HPD",Lower=HPD[1], Upper=HPD[2])

main_df1<-rbind(equal_df,hpd_df)
knitr::kable(main_df1)

```

	Interval	Lower	Upper
5%	Equal tailed interval	0.2111452	0.4371063
1	HPD	0.1948676	0.4086629

As we can see HPD interval is shorter than equal tailed credible interval. In fact HPDI is the shortest interval among all of the Bayesian Credible Interval. In symmetric credible interval(equal tailed credible interval), we exclude 5% of the density from each tail of distribution. If the posterior distribution is not unimodal and symmetric, there are point outside of the equal-tailed credible interval that have a higher posterior density than some points of the interval. If we want to choose the credible interval so that this not happen, we can do it by using the highest posterior density criterion for choosing it.

Question3: Bayesian Inference

Bayesian inference for the concentration parameter in the von Mises distribution. This exercise is concerned with directional data. The point is to show you that the posterior distribution for somewhat weird models can be obtained by plotting it over a grid of values. The data points are observed wind directions at a given location on ten different days. The data are recorded in degrees: (40,303,326,285,296,314,20,308,299,296), where North is located at zero degrees. The 10 observations in radians are: (-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

Assume that these data points are independent observations following the von Mises distribution:

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

where $I_0(k)$ is the modified Bessel function of the first kind of order zero [see ?besselI in R]. The parameter μ ($-\pi \leq y \leq \pi$) is the mean direction and $k > 0$ is called the concentration parameter. Large k gives a small variance around μ , and vice versa. Assume that μ is known to be 2.39. Let $k \sim \text{Exponential}(\lambda = 1)$ a priori, where λ is the rate parameter of the exponential distribution (so that the mean is $1/\lambda$).

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

At first we should obtain the posterior formula by this way:

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

Here we have:

$$p(y|\mu, \kappa) \propto p(\kappa|\mu, y) \times p(\kappa)$$

Prior=

$$p(\kappa) \propto \text{Exp}(\lambda = 1) = \lambda \cdot e^{-\lambda \cdot \kappa}$$

Model=

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

Likelihood=

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

So in order to obtain posterior, we should multiply prior in Likelihood.

```

mu <- 2.3
lambda<-1
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
kappa <- seq(0.01, 20, 0.01)
p_model <- function(y, kappa){
  model <- (exp(kappa * cos(y-mu)))/(2*pi*besseli(kappa,0))
  return(model)
}
prior <- function(kappa){
  return(dexp(kappa))
}

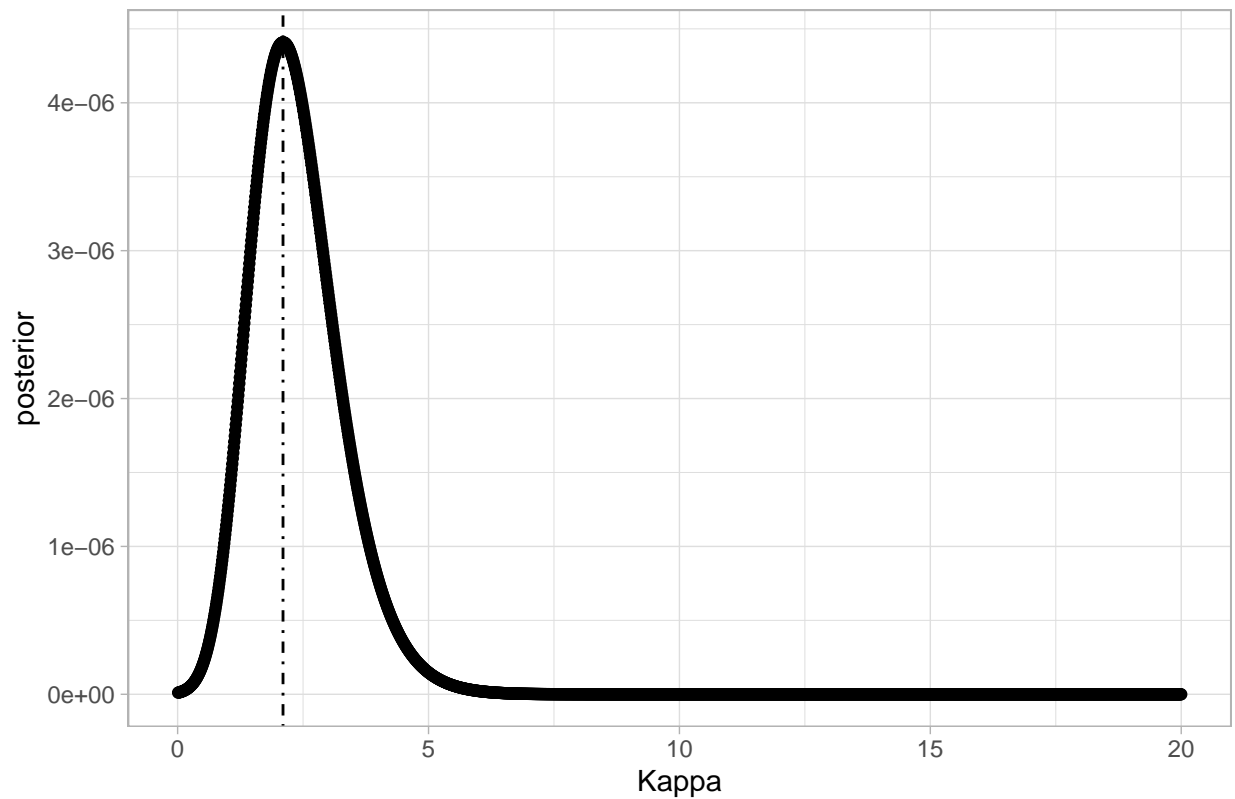
likelihood <- function(y, kappa){
  probs<-sapply(kappa, FUN=function(k,y){
    return(p_model(y,k))
  },y)
  return(apply(probs,2, prod))
}
posterior <- function(y, kappa){
  pos <- prior(kappa) * likelihood(y,kappa)
  return(pos)
}
prior_v <- prior(kappa)
like_v <- likelihood(y,kappa)
like_v1<-like_v * (1/sum(like_v)) * 100
post<-posterior(y,kappa)

#Here we want to plot prior, posterior and Likelihood based on different values of kappa, So we change
post_v <- prior_v * like_v
post_v <- post_v * (1/sum(post_v)) * 100
df_posterior <- data.frame("Kappa"=kappa, "posterior"=post)
kappa_max <- df_posterior$Kappa[which.max(df_posterior$posterior)]
library(ggplot2)

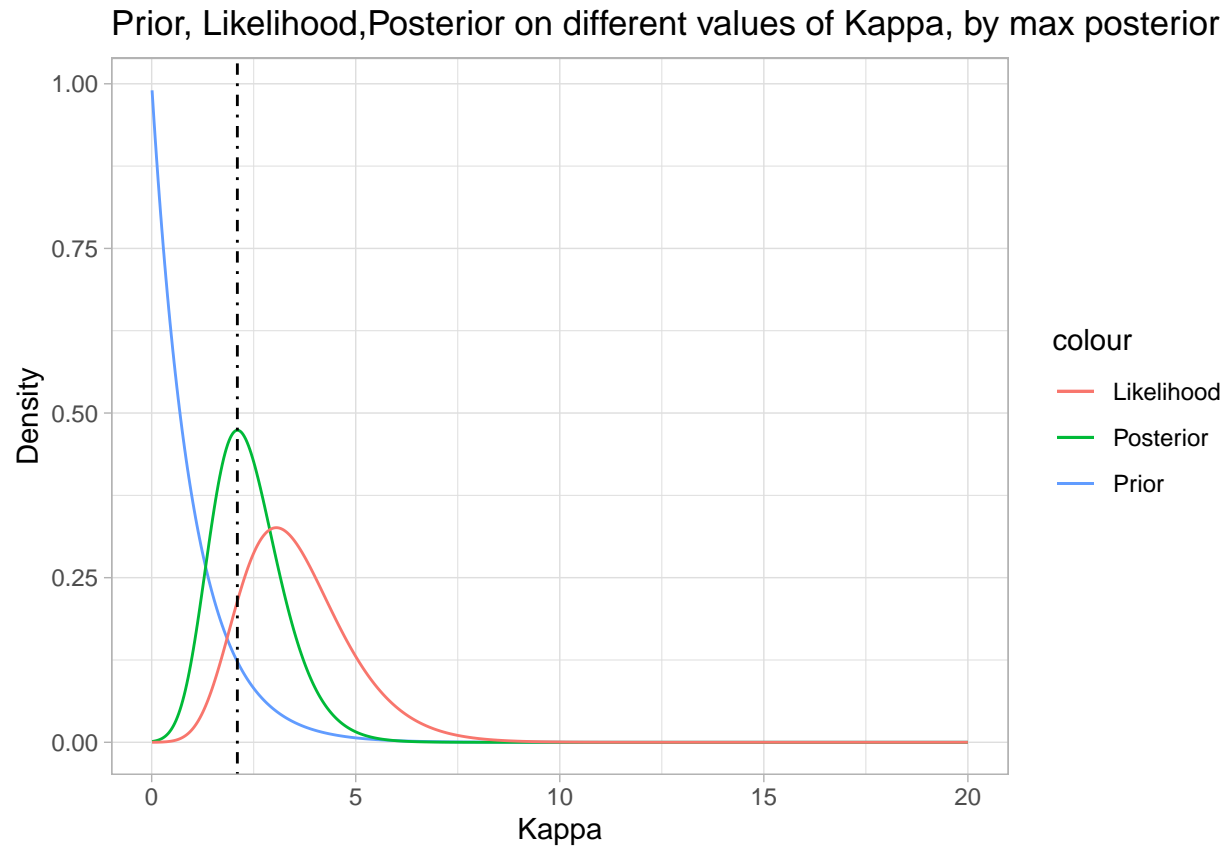
ggplot(data=df_posterior, aes(x=Kappa, y=posterior))+
  geom_point()+
  geom_vline(aes(xintercept = kappa_max), linetype= "dotdash")+
  ggtitle(paste("posterior by different k and max value=",kappa_max))+
  theme_light()

```

posterior by different k and max value= 2.1



```
df <- data.frame("Kappa"=kappa, "Prior"=prior_v, "Likelihood"=like_v1, "Posterior"=post_v)
k_max<- df$Kappa[which.max(df$Posterior)]
ggplot(df) +
  geom_line(aes(x=Kappa, y= Prior, color = "Prior"))+
  geom_line(aes(x= Kappa, y=Posterior, color="Posterior")) +
  geom_line(aes(x=Kappa, y=Likelihood, color="Likelihood"))+
  geom_vline(aes(xintercept=k_max), linetype="dotdash")+
  labs(title="Prior, Likelihood,Posterior on different values of Kappa, by max posterior", y="Density",
  theme_light()
```



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

To obtain the posterior mode, we find the max value of posterior and kappa related to this max value.

```
paste("by k = ", k_max, " the posterior would be max value")
```

```
## [1] "by k = 2.1 the posterior would be max value"
```