Bayesian Learning 01

Rakhshanda Jabeen

01/04/2020

Contents

Bernoulli Distribution

In this task, we let that $y_1, y_2, ..., y_n|\theta \sim Bern(\theta)$. We have obtained a sample with $s = 5$ successes in $n = 20$ trials. Assuming that, $\theta \sim Beta(2, 2)$ is a prior for θ.

- **Model:** $y_1, y_2, ..., y_n|\theta \sim Bern(\theta)$
- **Pror:** $\theta \sim Beta(2, 2)$
- **Likelihood Function:** Here $n = 20$, $s = 5$, $f = n - s = 15$. Then the likelihood function will be,

$$\begin{aligned} p(y_1, y_2, ..., y_n|\theta) &= p(y_1|\theta)\ p(y_2|\theta)...p(y_n|\theta) \\ &= \theta^5(1-\theta)^{15} \end{aligned}$$

- **Posterior:**

$$\begin{aligned} p(\theta|y_1, y_2, ..., y_n) &\propto p(y_1, y_2, ..., y_n|\theta)\ p(\theta) \\ &\propto \theta^5(1-\theta)^{15}\theta^{2-1}(1-\theta)^{2-1} \\ &\propto \theta^{(5+2)-1}(1-\theta)^{(15+2)-1} \\ &\sim Beta(7, 17) \end{aligned}$$

Thus the prior to posterior mapping:

$$\theta \sim Beta(2, 2) \quad \xrightarrow{y_1, y_2, ..., y_n} \quad \theta|y_1, y_2, ..., y_n \ \sim Beta(7, 17)$$

1.1 Draw Random Samples from Posterior Distribution $Beta(\alpha, \beta)$

The posterior distribution $Beta(\alpha, \beta)$ mean and standard deviation can be derived using followin formulas,

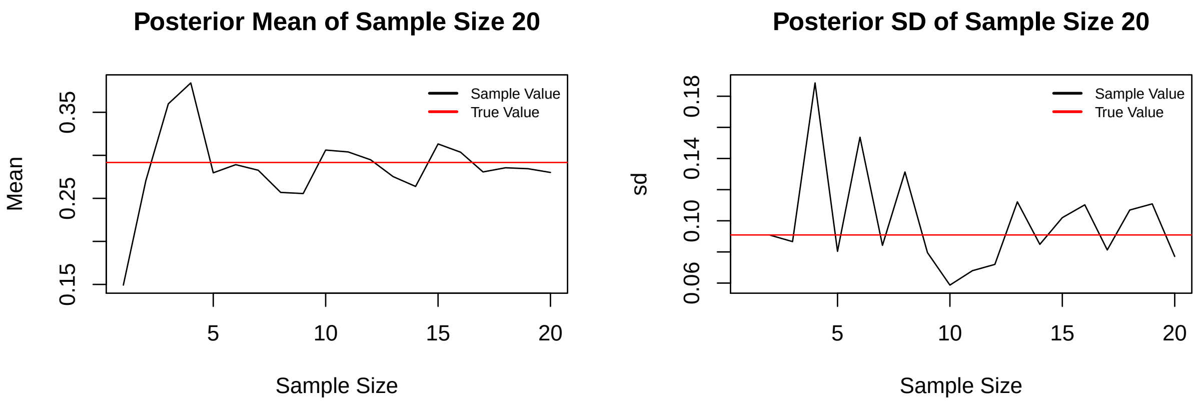
$$Mean = \frac{\alpha}{\alpha + \beta}$$

$$Standard\ Deviation = \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}$$

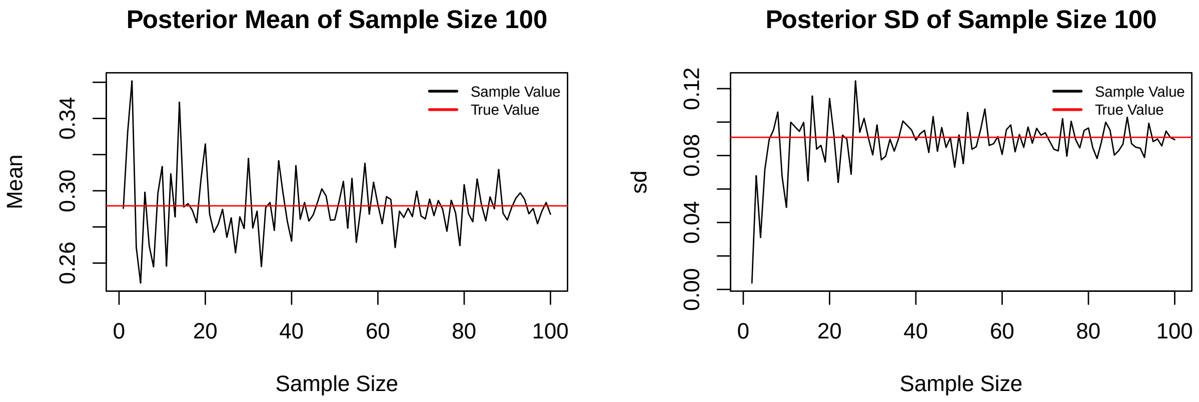
```r
BetaPosterior = function(n, alpha=7, beta=17){
  trueMean = alpha/(alpha+beta)
  trueSD = sqrt((alpha*beta)/((alpha+beta)^2*(alpha+beta+1)))
    Mean = numeric()
    sd = numeric()
    for (i in 1:n) {
      posterior = rbeta(i, alpha, beta)
      Mean[i] = mean(posterior)
      sd[i] = sd(posterior)
    }
    par(mfrow = c(1,2))
    ts.plot(Mean, main = paste('Posterior Mean of Sample Size',n),
            xlab = 'Sample Size')
    abline(h = trueMean, col = 'red')
    legend('topright', legend = c('Sample Value', 'True Value'),bty = 'n',
           col = c('black','red'), lty = c(1,1),cex=0.7, lwd = 2,xjust = 0)

    ts.plot(sd, main = paste('Posterior SD of Sample Size',n),
             xlab = 'Sample Size')
    abline(h = trueSD, col = 'red')
    legend('topright', legend = c('Sample Value', 'True Value'),bty = 'n',
           col = c('black','red'), lty = c(1,1), cex = 0.7, lwd = 2,xjust = 0)
}
```

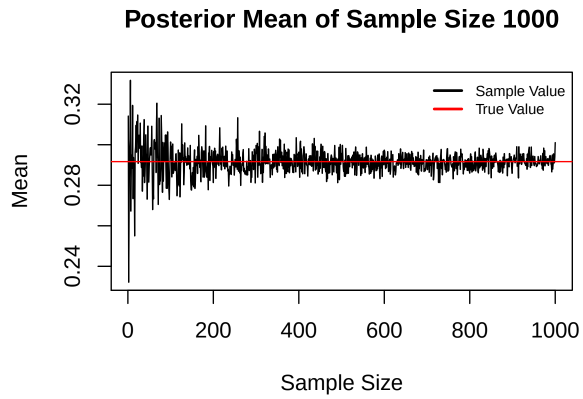
```
BetaPosterior(n = 20)
```

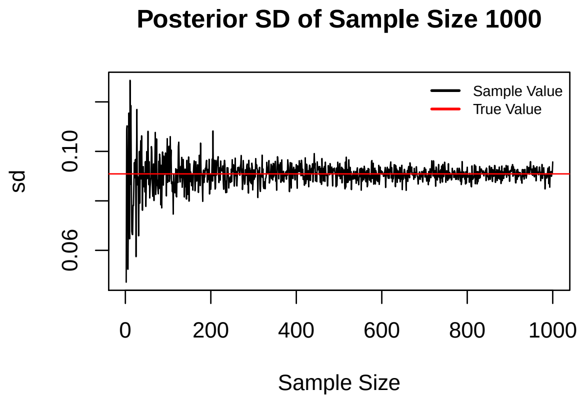


```
BetaPosterior(n = 100)
```



```
BetaPosterior(n = 1000)
```

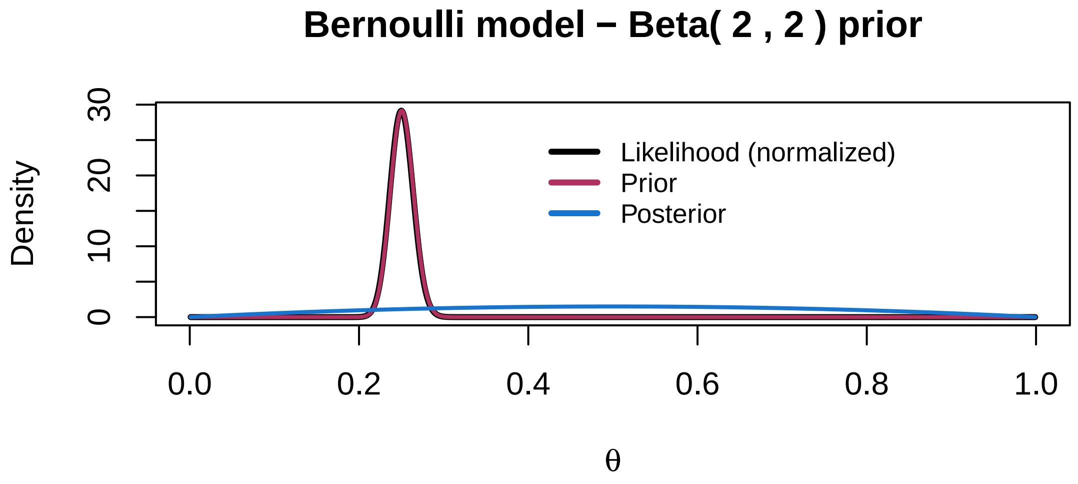




It is evident from all the above plots that as the size of random sample is increasing the posterior mean and standard deviation converges to the true values of mean and standard deviation.

```
BetaPriorPostPlot = function(a,b,n,p){
  xGrid = seq(0.001, 0.999, by=0.001)
  normalizedLikelihood = dbeta(xGrid, n*p+1, n*(1-p)+1)
  prior = dbeta(xGrid, a, b)
  posterior = dbeta(xGrid, a+n*p, b+n*(1-p))
  maxDensity = max(normalizedLikelihood, prior, posterior) # Use to make the y-axis high enough
  plot(xGrid, normalizedLikelihood, type = 'l', lwd = 3, xlim = c(0,1), ylim = c(0, maxDensity),
       xlab = expression(theta), ylab = 'Density',
       main = paste("Bernoulli model - Beta(",a,",",b,") prior"))
  lines(xGrid, posterior, lwd = 2, col = "maroon")
  lines(xGrid, prior, lwd = 2, col = "dodgerblue3")
  legend(x = 0.4, y = maxDensity*0.95, legend = c("Likelihood (normalized)", "Prior", "Posterior"),
         col =  c("black","maroon","dodgerblue3"), lwd = c(3,3,3), cex = 0.8, bty = 'n')
}

BetaPriorPostPlot(a = 2, b = 2, n = 1000, p = 0.25)
```



1.2. Posterior Probability $P(\theta > 0.3|y)$

In this task we have to compute $p(\theta > 0.3|y)$ using a random sample of size 10000 from posterior probability distribution $Beta(7, 17)$.

```
theta = 0.3
Sample = rbeta(n = 10000, 7, 17)
posterior_prob = mean(Sample>theta)
actual_prob = pbeta(theta, 7,17, lower.tail = F)
```

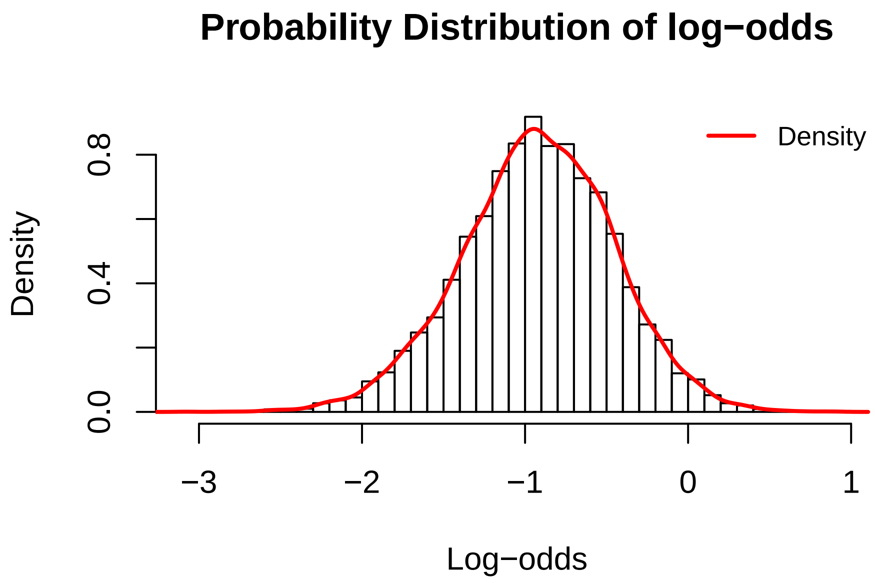
The probability $p(\theta > 0.3|y)$ computed simulated random sample is 0.44 which is approximately equal to the actual prabability 0.4399472 computed using pbeta() function of **R**.

1.3. Posterior Distribution of the Log-odds

We can compute the log-odds using the following formula,

$$\phi = log\frac{\theta}{1 - \theta}$$

```
log_odds = log(Sample/(1-Sample))
hist(log_odds,xlab = 'Log-odds', probability = T,
     main = 'Probability Distribution of log-odds', breaks = 40)
lines(density(log_odds), col = 'red', lwd = 2)
legend('topright', legend = c('Density'), lty = 1, cex = 0.8,
       col = c('red'), lwd = 2, bty = 'n')
```



It is evident from the histogram that posterior distribution of log-odds has a bell-curve centered around −1. Thus we can infer that log-odds of randomly selected samples from posterior distribution are normally distributed.

2. Log-Normal Distribution and the Gini Coefficient

In this task, we have data of ten randomly selected people saleries. A common model for non-negative continuous variables is the log-normal distribution. The log-normal distribution $log\mathcal{N}(\mu, \sigma^2)$ has density function.

- **Model:** $y_1, y_2, ..., y_n|\theta \sim log\mathcal{N}(\mu, \sigma^2)$
- **Pror:** $p(\sigma^2) \propto \frac{1}{\sigma^2}$
- **Likelihood Function:** Here assumed values of μ is 3.7. Then the likelihood function will be,

$$\begin{aligned} p(y_1, y_2, ..., y_n|\sigma^2) &= p(y_1|\sigma^2)\ p(y_2|\sigma^2)...p(y_n|\sigma^2) \\ &= \prod_{i=1}^{n} \frac{1}{y_i\sqrt{2\pi\sigma^2}} exp\bigg(\frac{-1}{2\sigma^2}\big(log(y_i) - \mu\big)^2\bigg) \\ &= \prod_{i=1}^{n} \bigg(\frac{1}{y_i\sqrt{2\pi\sigma^2}}\bigg) * exp\bigg(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2\bigg) \end{aligned}$$

Posterior

$$\text{Posterior } \propto \text{ prior * likelihood}$$

$$p(\sigma^2|y_1, y_2, ..., y_n) \propto \frac{1}{\sigma^2} * \prod_{i=1}^{n} \bigg(\frac{1}{y_i\sqrt{2\pi\sigma^2}}\bigg) * exp\bigg(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2\bigg)$$

$$\propto \frac{1}{\sigma^2} * \frac{1}{(\sigma^2)^{n/2}}\ exp\bigg(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2\bigg)$$

$$\propto \frac{1}{\sigma^{n+2}}\ exp\bigg(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2\bigg)$$

$$\propto \sigma^{-2(\frac{n}{2}+1)}\ exp\bigg(\frac{-n}{2\sigma^2} * \frac{\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2}{n}\bigg)$$

By the probability density function of scaled $Inv - \chi^2(\theta|v, s^2)$,

$$\begin{aligned} p(\theta) &= \frac{(\frac{v}{2})^{v/2}}{\Gamma(\frac{v}{2})}\ s^v\theta^{-(\frac{v}{2}+1)}\ exp(\frac{-v}{2\theta}s^2) \\ &\propto \theta^{-(\frac{v}{2}+1)}\ exp(\frac{-v}{2\theta}s^2) \end{aligned}$$

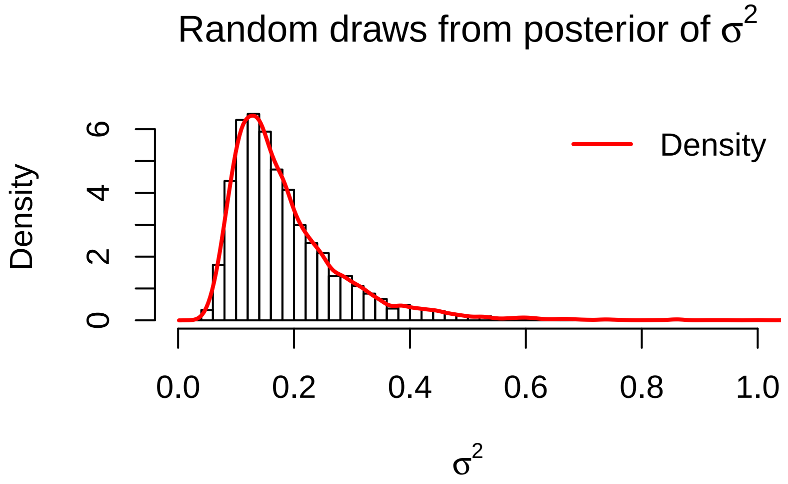
By comparing posterior distribution and scaled $Inv - \chi^2$ we can say that posterior of σ^2 is distributed as scaled $Inv - \chi^2$ with parameters $v = n$ and $s^2 = \tau^2 = \frac{\sum_{i=1}^{n}\big(log(y_i) - \mu\big)^2}{n}$.

2.1. Simmulation from Posterior of σ^2

We have simmulated a sample of size 10000 from posterior of σ^2 using the following steps:

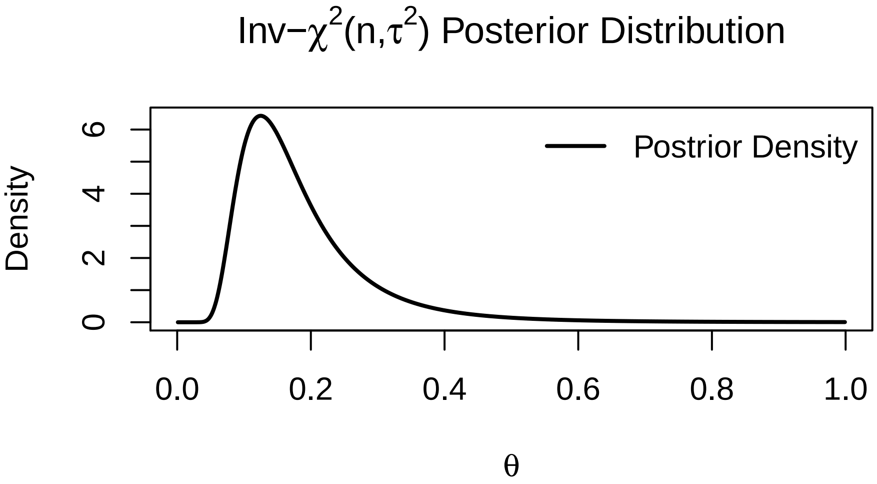
- Drwa $X \sim \chi^2(n)$
- Compute $\sigma^2 = \frac{n\tau^2}{X}$ (this draw is from scaled $Inv - \chi^2(n, \tau^2)$).

```r
library(LaplacesDemon)
y = c(44,25,45,52,30,63,19,50,34,67)
n = 10
mu = 3.7
rInvchisq = function(n, df, scale) (df*scale)/rchisq(n,df=df)
tau2 = sum((log(y)-mu)^2)/n
varPosterior = rInvchisq(n = 10000, df = n, scale = tau2)
hist(varPosterior, breaks = 80, probability = TRUE,xlim = c(0,1),
     xlab = TeX('$\\sigma^2$'),main = TeX('Random draws from posterior of $\\sigma^2$'))
lines(density(varPosterior), col = 'red', lwd = 2)
legend('topright', legend = 'Density', col = 'red', lwd = 2, bty = 'n')
```



Theoretical $Inverse - \chi^2(n, \tau^2)$ Posterior Distribution

```r
xGrid <- seq(0.001, 0.999, by=0.001)
posterior = dinvchisq(xGrid, n, tau2, log=FALSE)
plot(xGrid, posterior, type = 'l', lwd = 2, xlim = c(0,1), ylim = c(0, max(posterior)),
     xlab = expression(theta),
     ylab = 'Density',
     main = TeX(paste("$Inv-\\chi^2(n,\\tau^2)$ Posterior Distribution")))
legend('topright', legend = 'Postrior Density', col = 'black', lwd = 2, bty = 'n')
```

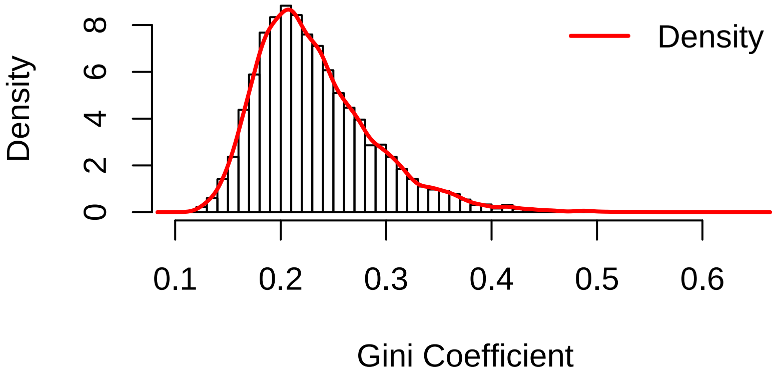


We can see that random draws from the posterior of σ^2 and the theoretical scaled $Inv - \chi^2(n, \tau^2)$ represents the similar density curves.

2.2. Gini Coefficient

Gini coefficient can be calculated using the formula, $G = 2\Phi(\sigma/\sqrt{2}) - 1$. When the data follows $log\mathcal{N}(\mu, \sigma)$ distribution. Here $\Phi(z)$ is the cumulative distribution function (CDF) for the standard normal distribution with mean zero and unit variance.

```
gini_coef = 2*pnorm(sqrt(varPosterior/2))-1
hist(gini_coef, breaks = 50, probability = TRUE,
     xlab = 'Gini Coefficient',main = '')
lines(density(gini_coef), col = 'red', lwd = 2)
legend('topright', legend = 'Density', col = 'red', lwd = 2, bty = 'n')
```



2.3. 90% Credible Interval for Gini Coefficient

90% Equal Tail Interval

We have used R's quantile() function to find the 90% equal tail credible interval.

```
CI = quantile(gini_coef, c(0.05,0.95))
```

Table 1: Equal Tail Interval

	x
5%	0.1612809
95%	0.3401861

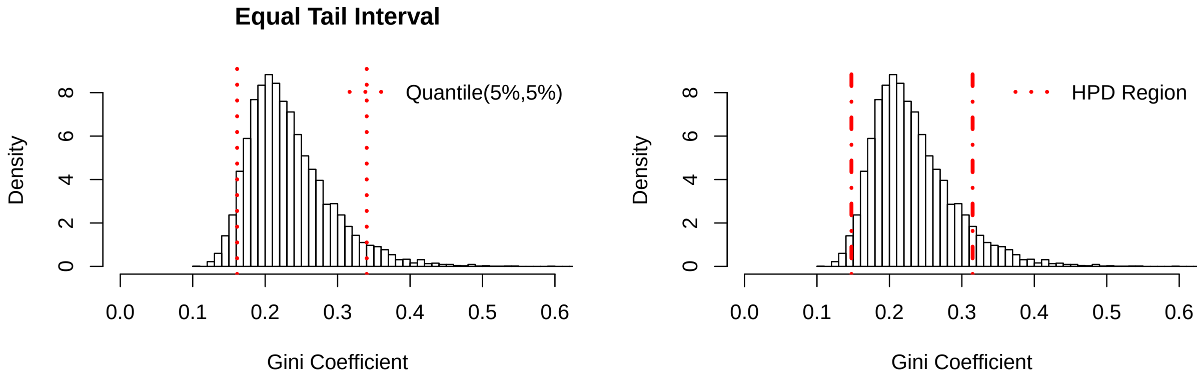
Highest Posterior Denisty Region

Highest probability density HPD interval contains gini coefficient values with the highest probability density function.

```
#_____(HPD)______#
library(HDInterval)
HPD_region = hdi(gini_coef, credMass = 0.9)[1:2]
```

Table 2: HPD Region

	x
lower	0.1475980
upper	0.3148302



3. Bayesian Inference for the Concentration Parameter in the von Mises Distribution

In this task, we let that Assume that these data points are independent observations following the von Mises distribution.

$$p(y|\mu, k) = \frac{exp\big(k\ cos(y-\mu)\big)}{2\pi I_0(k)}, \qquad -\pi \leq y \leq \pi,$$

where $I_0(k)$ is the modified Bessel function of the first kind of order zero. The parameter $\mu(-\pi \leq \mu \leq \pi)$ is the mean direction and $k > 0$ is called the concentration parameter.

- **Prior:** $k \sim exp(\lambda)$, $\qquad$ Where $\ \lambda = 1$
- **Likelihood Function:** Here $\mu = 2.39$. Then the likelihood function will be,

$$\begin{aligned} p(y_1, y_2, ..., y_n|k) &= p(y_1|k)\ p(y_2|k)...p(y_n|k) \\ &= \frac{1}{\Big(2\pi I_0(k)\Big)^n}\ exp\big(k * \sum_{i=1}^{n} cos(y_i - \mu)\big) \end{aligned}$$

3.1. Posterior Distribution of k for the Wind Direction Data

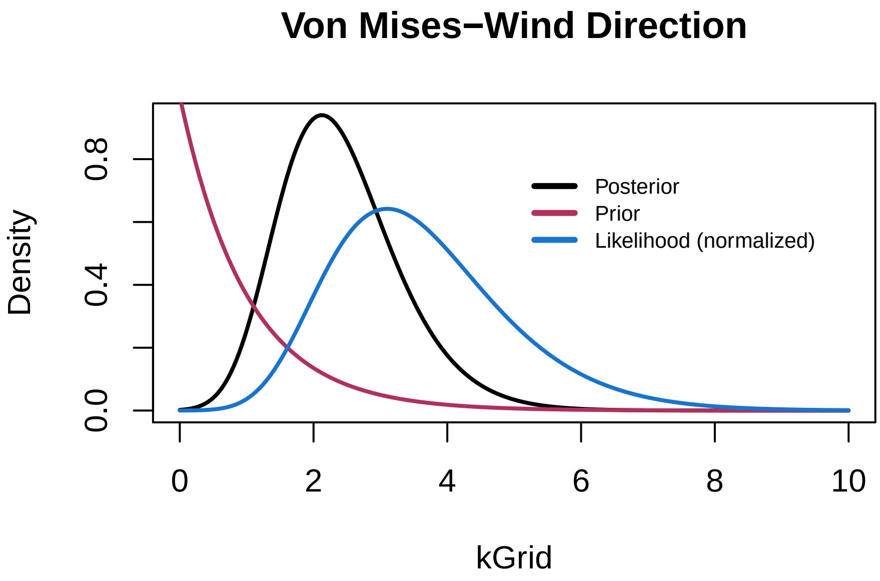
Posterior:

$$Poaterior \propto\ prior\ *\ Likelohhod$$
$$p(k|y_1, y_2, ..., y_n) \propto \frac{e^{-k}}{\Big(2\pi I_0(k)\Big)^n}\ exp\big(k * \sum_{i=1}^{n} cos(y_i - \mu)\big)$$

```r
#_______(a)______#
y = c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

dVonMises = function(y, mu=2.39, kappa){
  res = 1
  for (i in y){
    res = res*(exp(kappa*cos(i-mu))/(2*pi*besselI(kappa,0)))
  }
  return(res)
}
kGrid = seq(0, 10, length.out = 1000)
prior = dexp(kGrid)
likeli_hood = dVonMises(y, kappa = kGrid)
normal_likeli = (likeli_hood/sum(likeli_hood))/0.005
posterior = (prior*likeli_hood)
posterior = (posterior/sum(posterior))/0.005

plot(x = kGrid, y = posterior,type = 'l',lwd = 2, ylab = 'Density',
     main = 'Von Mises-Wind Direction')
lines(x = kGrid, y = prior, col = 'maroon', lwd = 2)
lines(x = kGrid, y = normal_likeli, col = 'dodgerblue3',lwd = 2)
legend(x = 5, y = 0.8, legend = c("Posterior", "Prior", "Likelihood (normalized)"),
       col = c("black","maroon","dodgerblue3"), lwd = c(3,3,3), cex = 0.7, bty = 'n')
```



3.2. Approximate Posterior Mode of k

In Bayeian Statistics, a maximum a posterior probability (MAP) estimate is an estimate of an unknown quantity, that equals the mode of the posterior distribution (wikipedia).

```
posterior_mod = kGrid[which.max(posterior)]
```

The approximate Posterior mod of kappa is 2.1221221.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(manipulate)
library(latex2exp)
library(knitr)
library(HDInterval)
setwd("E:/Bayesian Learning/LABS/lab_01")
BetaPosterior = function(n, alpha=7, beta=17){
  trueMean = alpha/(alpha+beta)
  trueSD = sqrt((alpha*beta)/((alpha+beta)^2*(alpha+beta+1)))
    Mean = numeric()
    sd = numeric()
    for (i in 1:n) {
      posterior = rbeta(i, alpha, beta)
      Mean[i] = mean(posterior)
      sd[i] = sd(posterior)
    }
    par(mfrow = c(1,2))
    ts.plot(Mean, main = paste('Posterior Mean of Sample Size',n),
            xlab = 'Sample Size')
    abline(h = trueMean, col = 'red')
    legend('topright', legend = c('Sample Value', 'True Value'),bty = 'n',
           col = c('black','red'), lty = c(1,1),cex=0.7, lwd = 2,xjust = 0)

    ts.plot(sd, main = paste('Posterior SD of Sample Size',n),
             xlab = 'Sample Size')
    abline(h = trueSD, col = 'red')
    legend('topright', legend = c('Sample Value', 'True Value'),bty = 'n',
           col = c('black','red'), lty = c(1,1), cex = 0.7, lwd = 2,xjust = 0)
}
BetaPosterior(n = 20)
BetaPosterior(n = 100)
BetaPosterior(n = 1000)
BetaPriorPostPlot = function(a,b,n,p){
  xGrid = seq(0.001, 0.999, by=0.001)
  normalizedLikelihood = dbeta(xGrid, n*p+1, n*(1-p)+1)
  prior = dbeta(xGrid, a, b)
  posterior = dbeta(xGrid, a+n*p, b+n*(1-p))
  maxDensity = max(normalizedLikelihood, prior, posterior) # Use to make the y-axis high enough
  plot(xGrid, normalizedLikelihood, type = 'l', lwd = 3, xlim = c(0,1), ylim = c(0, maxDensity),
       xlab = expression(theta), ylab = 'Density',
       main = paste("Bernoulli model - Beta(",a,",",b,") prior"))
  lines(xGrid, posterior, lwd = 2, col = "maroon")
  lines(xGrid, prior, lwd = 2, col = "dodgerblue3")
  legend(x = 0.4, y = maxDensity*0.95, legend = c("Likelihood (normalized)", "Prior", "Posterior"),
         col =  c("black","maroon","dodgerblue3"), lwd = c(3,3,3), cex = 0.8, bty = 'n')
}

BetaPriorPostPlot(a = 2, b = 2, n = 1000, p = 0.25)
theta = 0.3
Sample = rbeta(n = 10000, 7, 17)
posterior_prob = mean(Sample>theta)
```

```r
actual_prob = pbeta(theta, 7,17, lower.tail = F)
log_odds = log(Sample/(1-Sample))
hist(log_odds,xlab = 'Log-odds', probability = T,
     main = 'Probability Distribution of log-odds', breaks = 40)
lines(density(log_odds), col = 'red', lwd = 2)
legend('topright', legend = c('Density'), lty = 1, cex = 0.8,
       col = c('red'), lwd = 2, bty = 'n')
library(LaplacesDemon)
y = c(44,25,45,52,30,63,19,50,34,67)
n = 10
mu = 3.7
rInvchisq = function(n, df, scale) (df*scale)/rchisq(n,df=df)
tau2 = sum((log(y)-mu)^2)/n
varPosterior = rInvchisq(n = 10000, df = n, scale = tau2)
hist(varPosterior, breaks = 80, probability = TRUE,xlim = c(0,1),
     xlab = TeX('$\\sigma^2$'),main = TeX('Random draws from posterior of $\\sigma^2$'))
lines(density(varPosterior), col = 'red', lwd = 2)
legend('topright', legend = 'Density', col = 'red', lwd = 2, bty = 'n')
xGrid <- seq(0.001, 0.999, by=0.001)
posterior = dinvchisq(xGrid, n, tau2, log=FALSE)
plot(xGrid, posterior, type = 'l', lwd = 2, xlim = c(0,1), ylim = c(0, max(posterior)),
     xlab = expression(theta),
     ylab = 'Density',
     main = TeX(paste("$Inv-\\chi^2(n,\\tau^2)$ Posterior Distribution")))
legend('topright', legend = 'Postrior Density', col = 'black', lwd = 2, bty = 'n')
gini_coef = 2*pnorm(sqrt(varPosterior/2))-1
hist(gini_coef, breaks = 50, probability = TRUE,
     xlab = 'Gini Coefficient',main = '')
lines(density(gini_coef), col = 'red', lwd = 2)
legend('topright', legend = 'Density', col = 'red', lwd = 2, bty = 'n')
CI = quantile(gini_coef, c(0.05,0.95))
library(kableExtra)
kable(CI,"latex", caption = "Equal Tail Interval", booktabs = T) %>%
  kable_styling(latex_options = "HOLD_position")
#_____(HPD)______#
library(HDInterval)
HPD_region = hdi(gini_coef, credMass = 0.9)[1:2]
kable(HPD_region,"latex", caption = "HPD Region", booktabs = T) %>%
  kable_styling(latex_options = "HOLD_position")
par(mfrow=c(1,2))
hist(gini_coef, breaks = 70, probability = T,freq = F,xlim = c(0,0.6),
     xlab = 'Gini Coefficient',main = 'Equal Tail Interval')
abline(v = CI[1], col = 'red', lwd = 3, lty = 3)
abline(v = CI[2], col = 'red', lwd = 3, lty = 3)
legend('topright', legend = 'Quantile(5%,5%)', lwd = 3,
       col = 'red', lty = 3, bty = 'n')

hist(gini_coef, breaks = 70, probability = T,freq = F,xlim = c(0,0.6),
     xlab = 'Gini Coefficient',main = '')
abline(v = HPD_region[1], col = 'red', lwd = 3, lty = 4)
abline(v = HPD_region[2], col = 'red', lwd = 3, lty = 4)
legend('topright', legend = 'HPD Region', lwd = 3,
       col = 'red', lty = 3, bty = 'n')
```

```r
#_______(a)______#
y = c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

dVonMises = function(y, mu=2.39, kappa){
  res = 1
  for (i in y){
    res = res*(exp(kappa*cos(i-mu))/(2*pi*besselI(kappa,0)))
  }
  return(res)
}
kGrid = seq(0, 10, length.out = 1000)
prior = dexp(kGrid)
likeli_hood = dVonMises(y, kappa = kGrid)
normal_likeli = (likeli_hood/sum(likeli_hood))/0.005
posterior = (prior*likeli_hood)
posterior = (posterior/sum(posterior))/0.005

plot(x = kGrid, y = posterior,type = 'l',lwd = 2, ylab = 'Density',
     main = 'Von Mises-Wind Direction')
lines(x = kGrid, y = prior, col = 'maroon', lwd = 2)
lines(x = kGrid, y = normal_likeli, col = 'dodgerblue3',lwd = 2)
legend(x = 5, y = 0.8, legend = c("Posterior", "Prior", "Likelihood (normalized)"),
       col = c("black","maroon","dodgerblue3"), lwd = c(3,3,3), cex = 0.7, bty = 'n')
posterior_mod = kGrid[which.max(posterior)]
``
```