TDDE07 - Lab 1

Martin Friberg - marfr370

2021/04/17

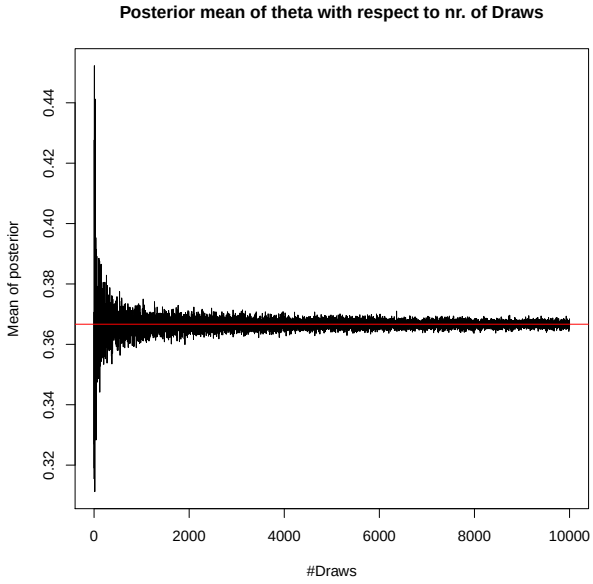
1 Assignment 1.

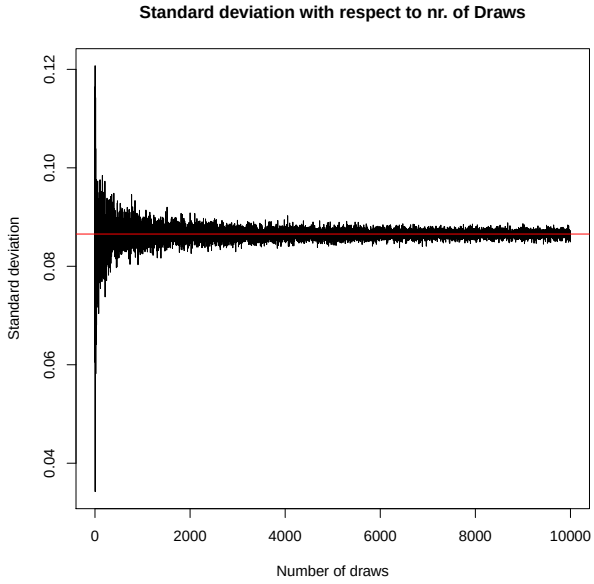
1.1 Question a)

$y, ..., y1 \sim Bern(\theta)$ and we assume that the sample obtained has 8 successes and 16 failures in 24 trials. Assuming a Beta(α_0, β_0) prior for θ and we let $\alpha_0 = \beta_0 = 3$.

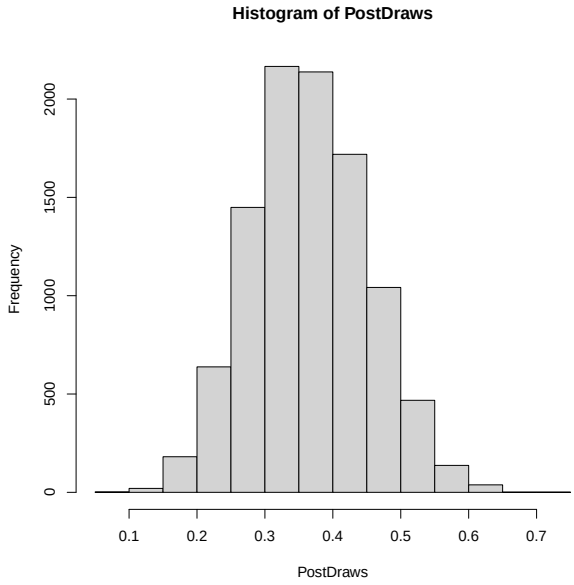
Since the prior belongs to the same family as the model, the prior is a conjugate prior, which results in that the prior and the posterior are conjugate distributions. The posterior distribution is proportional to the $B(\alpha + s, \beta + f)$ density.

The true mean of the beta distribution is calculated as $\frac{\alpha}{(\alpha+\beta)}$ and the true standard deviation is calculated as $\sqrt{\frac{\alpha*\beta}{(\alpha+\beta)^2*(\alpha+\beta+1)}}$) By drawing random numbers from the posterior beta distribution we obtain different means and standard deviations for different number of draws. As seen in the graphs below, the posterior mean and standard deviation converges to the true values as the number of draws grow large.





For 10 000 draws, the θ-values are distributed as follows.



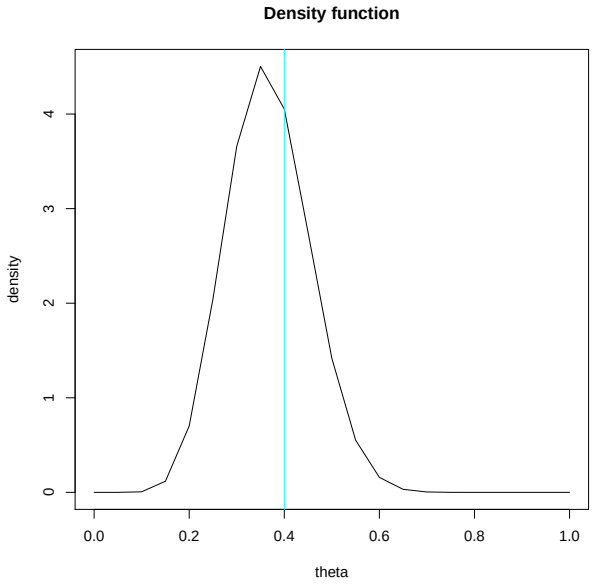
1.1.1 Code for question 1a

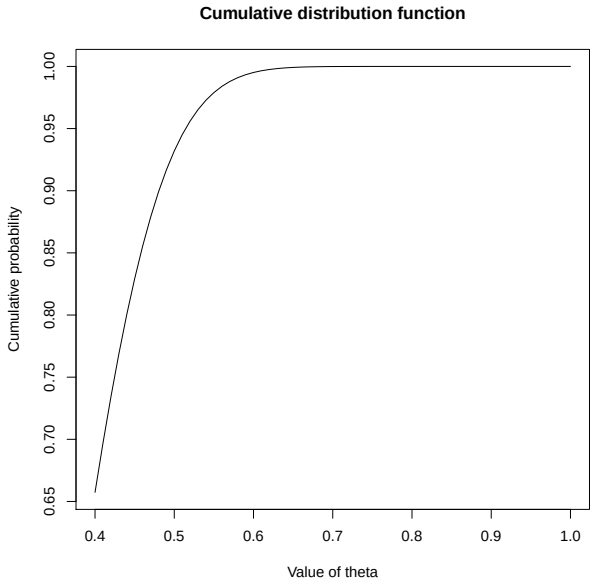
```
#Mean of beta distribution is calculated as alpha/(alpha+beta)
meanBeta <- function(a, b){
  return(a/(a+b))
}

#Standard dev of  beta distribution
stdBeta <- function(a,b){
  sqrt(a*b/((a+b)^2*(a+b+1)))
}

BetaPrior <- function(NrDraws, s, f, a ,b){
  PostDraws <- matrix(0,NrDraws,1)  ## Setting 0 to a one-Ndraws dimensional matrix
  PostDraws <- rbeta(NrDraws, a+s, b+f) #Drawing values from the beta distribution
  return(PostDraws)
}

#Function for calculating standard deviation from sample
# Variance = alpha*beta/((alpha+beta)^2*(alpha+beta+1))
std <- function(nrDraws, mean, postdraws){
  return(sqrt(sum((postdraws-mean)^2)/(nrDraws-1)))
}

set.seed(12345)
a <- 3
b <- 3
successes <- 8
failures <- 16
prob <- successes/(successes+failures)
```

```r
postBetaMean <- meanBeta(a+successes, b+failures)
postBetaStd<- stdBeta(a+successes, b+failures)
vecMean <- c()
vecStd <- c()
for (i in seq(1, 10000, 1)){
  # Generating draws from the joint posterior of theta
  PostDraws <- BetaPrior(i, successes, failures, a, b)
  #Adding the mean of different posterior draws with different number of draws
  vecMean <- append(vecMean, mean(PostDraws))

  vecStd <- append(vecStd, std(i, mean(PostDraws), PostDraws))
}

plot(seq(1, 10000, 1),
     vecMean,
     main="Mean with respect to nr. of Draws",
     xlab="#Draws",
     ylab="Mean of posterior",
     type="l")
abline(h=postBetaMean, col="blue")

plot(seq(1, 10000, 1),
     vecStd,
     main="Standard deviation with respect to nr. of Draws",
     xlab="Number of draws",
     ylab="Standard deviation",
     type="l")
abline(h=postBetaStd, col="blue")

hist(PostDraws)             # Plotting the histogram of theta-draws
```

1.2 Question b)

Plotting the posterior density of θ aswell as the limit 0.4. For calculating the true probability of obtaining a θ-value larger than 0.4 the pbeta function was used with the posterior Beta(α_0+s, β_0+f). The obtained true probability was 0.3427.. and the obtained probability of θ being larger that 0.4 from the density function was 0.3406 which is very similar to the real probability.





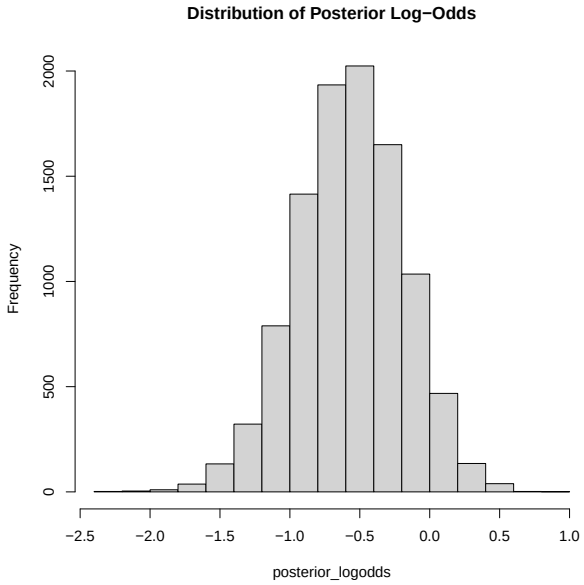
1.2.1 Code for question 1b

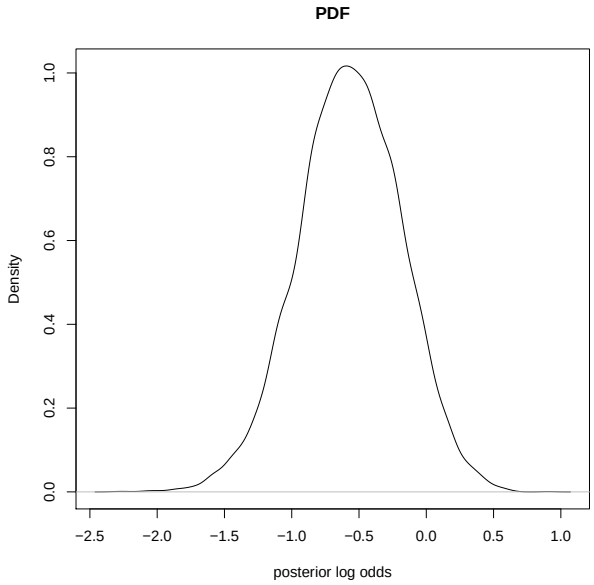
```
# I_x(a,b) is pbeta(x, a, b).
# pbeta is the cumulative distribution function
# so pbeta(0.4, alpha, beta) calculates the area under the graph for values
# below 0.4
trueProb <- pbeta(0.4, a+successes, b+failures, lower=FALSE) #lower=FALSE

#dbeta returns the density (to evaluate the beta density)
pdf('ThetaDensity.pdf')
plot(seq(0, 1, by = 0.05),
     dbeta(seq(0, 1, by = 0.05),a+successes, b+failures),
     main="Density function",
     xlab="theta",
     ylab="density",
     type="l")
abline(v=0.4, col="cyan")
dev.off()

# Gives us the complement of the cumulative distribution function
pdf("CumulativeTheta.pdf")
cumulative <- pbeta(seq(0.4, 1, by = 0.01) , a+successes, b+failures)
plot(seq(0.4, 1, by = 0.01),
     cumulative,
     main="Cumulative distribution function",
     xlab="Value of theta",
     ylab="Cumulative probability",
```

```
     type="l")
dev.off()

# Approximate posterior probability of theta>0.4
post_prob <- mean(PostDraws>0.4)
```

1.3 Question c)

The posterior log-odds $\phi = \log \frac{\theta}{1-\theta}$ was calculated and the histogram as well as the density of the posterior log-odds was plotted.





1.3.1 Code for question 1c

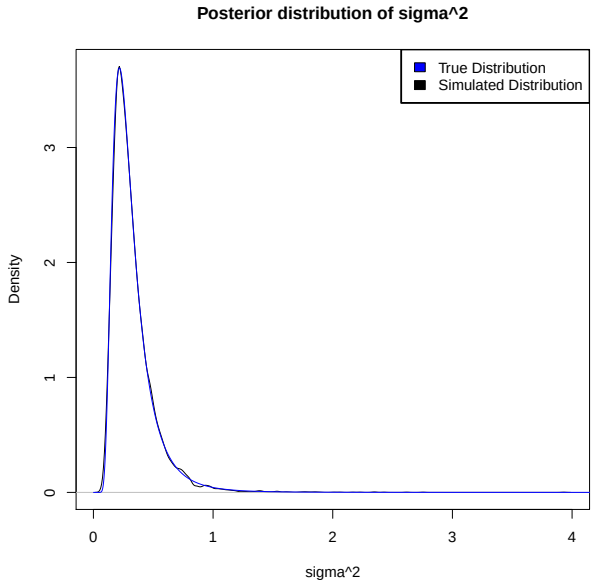
```
posterior_logodds <- log(PostDraws/(1-PostDraws))
pdf("HistogramPostLogOdds.pdf")
hist(posterior_logodds, main ="Distribution of Posterior Log-Odds")
dev.off()
pdf("DensityPostLogOdds.pdf")
plot(density(posterior_logodds), main = "PDF",
     xlab="posterior log odds")
dev.off()
```

2 Assignment 2.

2.1 Question a)

Income of swedish people in thousands SEK (38, 20, 49, 58, 31, 70, 18, 56, 25 and 78). For these kinds of values, a log normal distribution are often used since the variables are continuous and non-negative. The mean of the log normal distribution is $\mu = 3.8$. The posterior of σ^2 is the $Inv - \chi^2(n, \tau^2)$. Where $\tau^2 = \frac{\sum_{i=1}^{n}(log(y_i - \mu)^2}{n}$. To obtain a value of τ^2, the y values (i.e the incomes) were logarithmized, and the obtained values were used in the function to calculate τ^2.

10 000 draws were simulated from the posterior of σ^2 with $\mu = 3.8$ by first drawing 10 000 values of X from the $\chi^2(n)$ distribution. Thereafter σ^2 was computed from $\sigma^2 = \frac{n*\tau^2}{X}$. The achieved distribution was then compared with the theoretical $Inv\ \chi^2(n, \tau^2)$ distribution in the plot below. What can be seen is that the theoretical distribution and the simulated distribution are very similar. This is thanks to the large number of draws made.



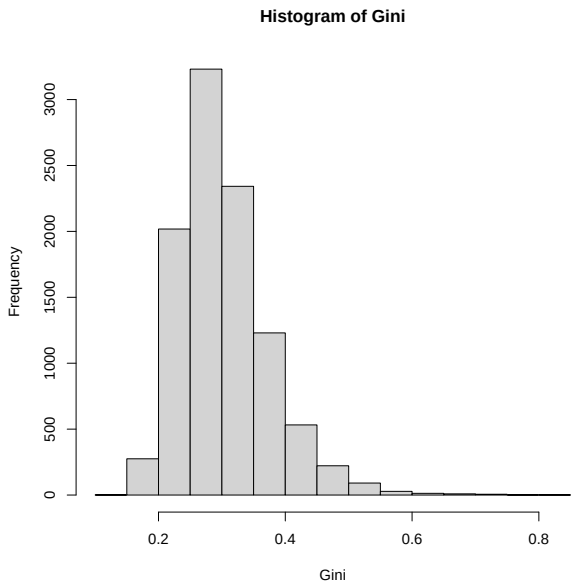
2.1.1 Code for Assignment 2a

```
set.seed(12345)
#tao squared = sum((log yi - mu))^2/n
taoSquared <- function(n, d, mean){
  logData <- log(d)
  tao_squared <- sum((logData-mean)^2)/n
  return(tao_squared)
}

# Calculating the PDF of the scaled inverse chi squared distribution
# n = degrees of freedom, squaredTao=scaling parameter
# The gamma function gamma(x) is defined as the integral over t from 0 - infinity
# of t^(x-1)*exp(-t) dt. The function returns the gamma function and the natural
# logarithm of the absolute value of the gamma function.
scaledInvChiSquare <- function(x, df, taosq){
  return(((taosq*df/2)^(df/2))/gamma(df/2) *
    (exp((-df*taosq)/(2*x))/(x^(1+df/2))))
```

```
}


Nobs <- 10000
Ndraws <- 10000
Data <- c(38,20,49,58,31,70,18,56,25,78)
Datamean <- 3.8
n <- length(Data)

# Calculating the scaling factor tao^2
squaredTao <- taoSquared(n, Data, Datamean)

degF <- n
# simulating from the posterior, normal model with unknown variance (sigma^2 unknown)
xDraw=rchisq(10000, degF) #rchisq(nr of draws, degrees of freedom)
sigmaSquared=n*squaredTao/xDraw # sigma^2 = degrees of freedom * scaling factor / X
# Probability density function extends over the domain x>0
x <- seq(0.001, 10, 0.001)
truePosterior <- scaledInvChiSquare(x, degF, squaredTao)

pdf("PostDistSigma.pdf")
plot(density(sigmaSquared),
     main="Posterior distribution of sigma^2",
     xlab="sigma^2")
lines(x,truePosterior,
      col="blue")
legend("topright",
       c("True Distribution", "Simulated Distribution"),
       fill=c("blue", "black"),
       box.lwd = 2)
dev.off()
```

2.2 Question b)

The most common measure of income inequality is the Gini coefficient (G) which has a value between 0 and 1. It can be shown that $G = 2 * \phi\frac{\sigma}{\sqrt{(2)}} - 1$ when incomes follow a $\log N(\mu, \sigma^2)$ distribution. The CDF was calculated for the posterior $\frac{\sigma}{\sqrt{(2)}}$ with $\mu = 0$ and $sd = 1$, whereafter the Gini coefficient was deduced for the different values of the posterior σ^2 draws. The Histogram of the Gini values and the posterior density of the Gini coefficient are seen below.

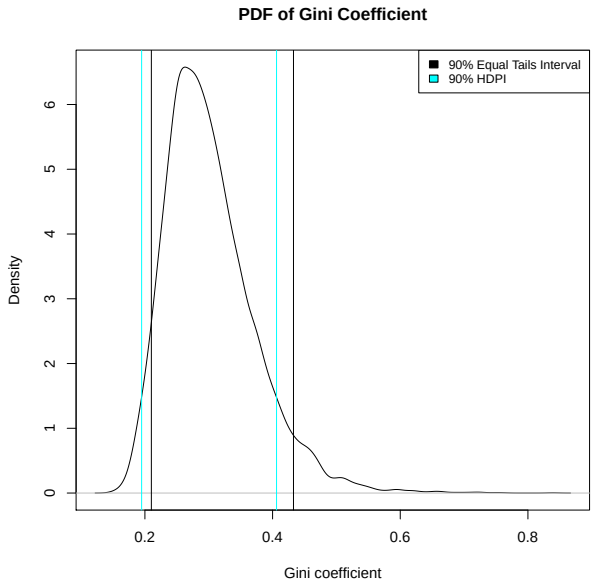


2.2.1 Code for Assignment 2b

```
# Unit variance => sd = 1
CDF <- pnorm(sqrt(sigmaSquared/2), mean=0, sd=1)
Gini <- 2 * CDF - 1
pdf("HistGini.pdf")
hist(Gini)
dev.off()
plot(density(Gini),
     main="Posterior PDF of Gini Coefficient",
     xlab="Gini value")
```

2.3 Question c)

The posterior distribution density is seen in the graph below together with the 90% equal tail interval as well as the 90% highest posterior density interval (HPDI). The two intervals are pretty similar, but a difference can be seen in that the 90% highest posterior density interval is shifted to the left since the posterior density distribution has a longer tail to the right.



2.3.1 Code for Assignment 2c

```
# 90% equal tail credible interval
ci_90 <- quantile(Gini, probs=c(0.05,0.95))
plot(density(Gini))
kernel <- density(Gini)
kernelFrame <- data.frame(x=kernel$x, density=kernel$y)
# Sorting the densities from lowest to highest
# with is used for sorting a Data Frame by Vector Name
kernelFrame <- kernelFrame[
    with(kernelFrame, order(density, decreasing=TRUE)),
    ]

kernelFrame$density <- cumsum(kernelFrame$density)/sum(kernelFrame$density)

HPDI <- kernelFrame[kernelFrame$density<0.9, ]

giniInterval <- c(min(kernel$x), max(kernel$x))
hpdiGini <- c(min(HPDI$x), max(HPDI$x))
print(hpdiGini)
print(giniInterval)

pdf("PDFGini.pdf")
plot(density(Gini),
     main="PDF of Gini Coefficient",
     xlab="Gini coefficient")
abline(v=ci_90[1], col="black")
abline(v=ci_90[2], col="black")
abline(v=min(HPDI$x), col="cyan")
abline(v=max(HPDI$x), col="cyan")
op <- par(cex = 0.8)
legend("topright",
       c("90% Equal Tails Interval ", "90% HDPI"),
```

```
       fill=c("black", "cyan"))
dev.off()
```

2.4 Question 3a & b)

10 wind direction observations in radians: -2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2,79, 2.23, 2.07, 2.02. The points are independent and follows the von Mises Distribution which has the following likelihood function:

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

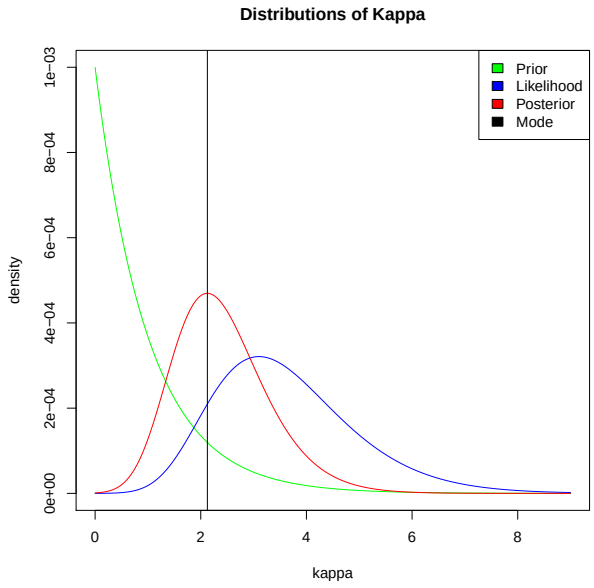
. $I_0(\kappa)$ is the modified Bessel function of the first kind of order zero. $\mu = 2.39$. κ~Exponential($\lambda = 1$) a priori. A fine grid of κ-values was created between the values 0 and 9 for the best visibility of the posterior distribution (which was derived later on) started with values between 0 and 20. The posterior is equal to the prior*likelihood. The prior was calculated by using the PDF of the exponential distribution with $\lambda = 1$. The likelihood of the Von Mises Distribution was computed by multiplication over the different observations

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

which is equal to

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

The values that was retrieved from calculations were the multiplied. This resulted in the plot below. The mode was also calculated to 2.125 for the posterior distribution of κ.




```
######################################################################
## sub-ass a)
## Plot the posterior distribution of kappa for the wind direction data over
## a fine grid of kappa values
######################################################################
```

```r
priorExponential <- function(lambda, x){
  return(lambda*exp(-lambda*x))
}


vonMisesLikelihood <- function(kappas, datap, mu){
  n <- length(datap)
  sumCos <- 0
  # Multiplication over datapoints
  for (i in datap){
    sumCos <- sumCos + cos(i - mu)
  }
  numerator <- exp(kappas*sumCos)
  denominator <- (2*pi*besselI(kappas,0))^n
  return(numerator/denominator)
}

kappaVal <- seq(0,9,0.001)
lambdaVal <- 1
priorDist <- priorExponential(lambdaVal, kappaVal)


dataPoints <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
mu <- 2.39

likelihood <- vonMisesLikelihood(kappaVal, dataPoints, mu)
posteriorExponential <- priorDist*likelihood


######################################################################
## sub-ass b)
## Find the (approximate) posterior mode of kappa from the info in a)
######################################################################

# A maximum a posteriori probability estimate is an estimate of an
# unknown quantity, that equals the mode of the posterior distribution.
# So, to find the mode, we want to maximize the MAP.
# e.g maximize the probability of kappa given the data

mode <- kappaVal[which.max(posteriorExponential)]

pdf("DistsofKappa.pdf")
plot(kappaVal, priorDist/sum(priorDist),
     col="green",
     xlab="kappa",
     ylab="density",
     type="l",
     main="Distributions of Kappa")
lines(kappaVal,
      likelihood/sum(likelihood),
      col="blue")
lines(kappaVal,
      posteriorExponential/(sum(posteriorExponential)),
```

```
      ylab="density",
      main="Posterior distribution of kappa",
      col="red")
abline(v=mode)
legend("topright",
       c("Prior", "Likelihood", "Posterior", "Mode"),
       fill=c("green", "blue", "red", "black"))
dev.off()
```