732A73: Bayesian Learning

Computer Lab 1

Oriol Garrobé, Dávid Hrabovszki

11 April 2020

Question 1. Bernoulli. . . again

(a)

Given the posterior $\theta|y \sim Beta(\alpha = 7, \beta = 17)$ we can compute the expected value and the standard deviation of the distribution.

The Expected value is:

$$E(\theta) = \frac{\alpha}{\alpha + \beta} = \frac{7}{7 + 17} = 0.2917$$

The standard deviation is:

$$Sd(\theta) = \sqrt{\frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}} = \sqrt{\frac{7 \cdot 17}{(7 + 17)^2(7 + 17 + 1)}} = 0.0909$$

Once we have the posterior we can draw random samples from it. We draw samples with different number of draws in ascending order. For each sample we compute the mean and standard deviation. Then we plot the values and study whether the mean and standar deviation get close to the true values.

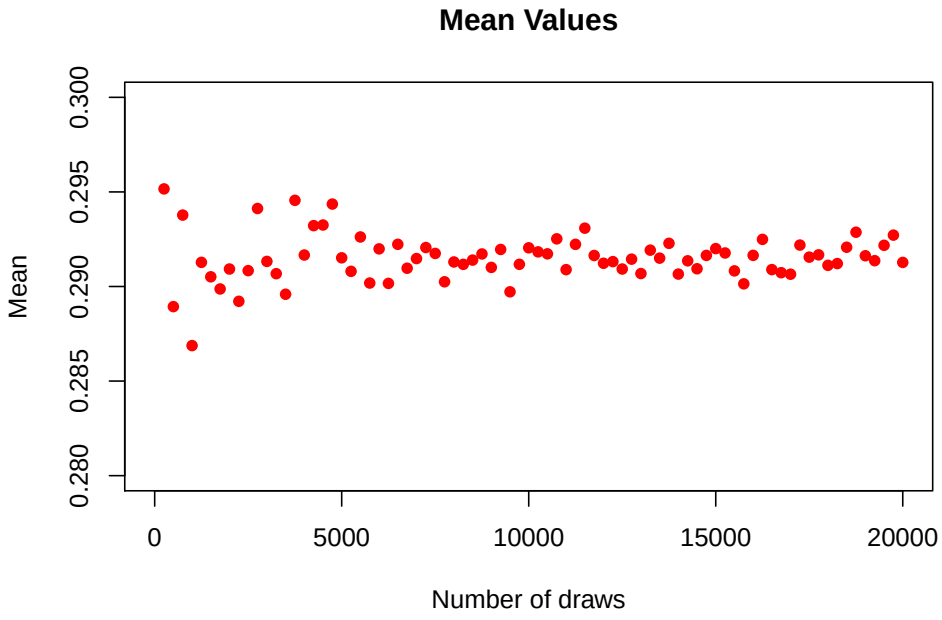

Figure 1: Mean values for different size samples.

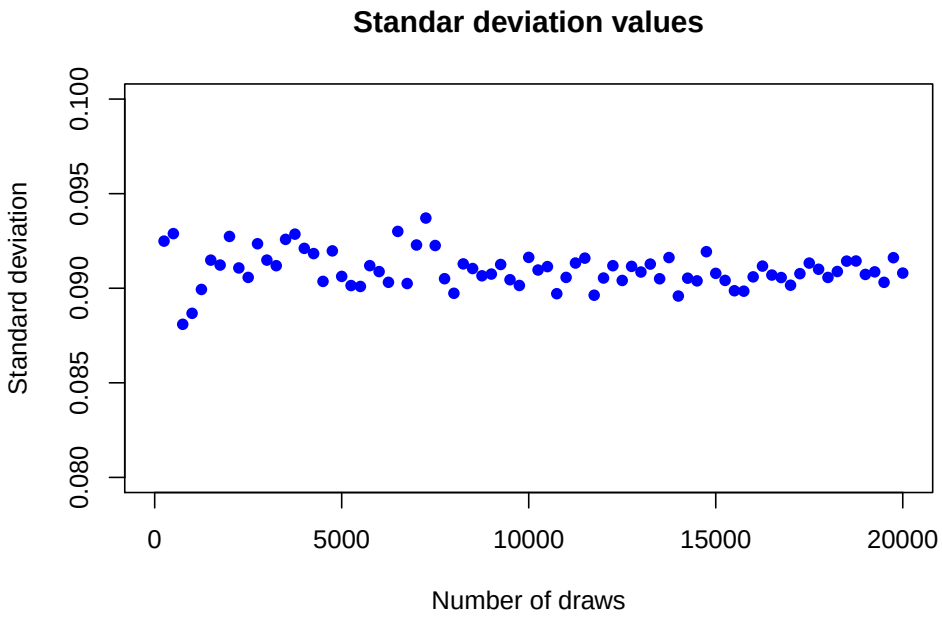

Figure 2: Standard deviation values for different size samples.

In Figure 1 we can see that with a small number of samples the sample mean is already close to the true value, but the bigger the sample the closer it gets. It happens the same thing in Figure 2 with the standard deviation values.

(b)

We can compute the probability of $\theta > 0.3$ using the drawn sample and compare it to the exact number computed using `pbeta()`. The value obtained from the sample is 0.4333 and the exact probability is 0.4399472 - see code in Appendix. We can conclude that with this sample size both probabilities are very similar.

(c)

From this point we compute the log-odds $\phi = log\frac{\theta}{1-\theta}$ from the posterior sample.

In Figure 3 we can see the distribution of the generated sample.

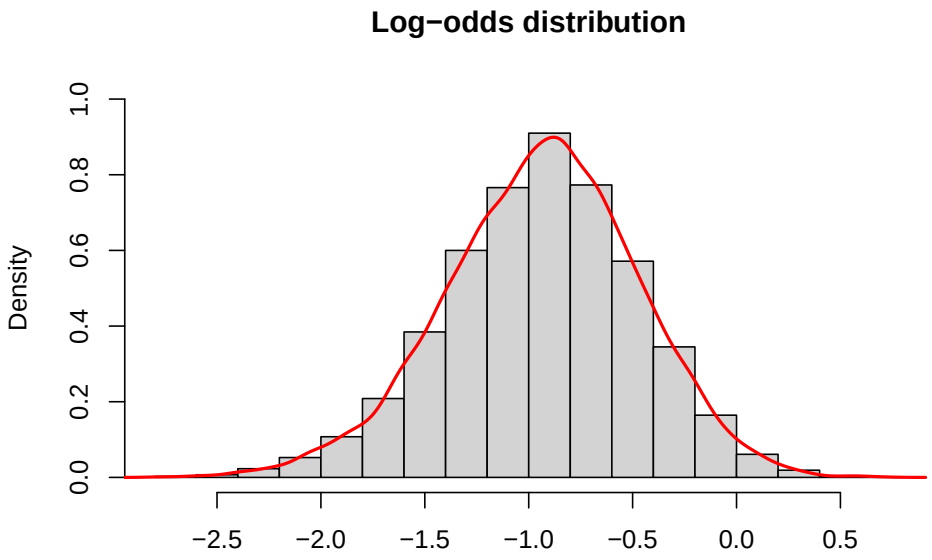

Figure 3: Histogram of the simulated log-odds.

Question 2. Log-normal distribution and the Gini coefficient.

In this question we are given data regarding random people income. As the data is continuous non-negative, we use the log-normal distribution with density function:

$$p(y|\mu,\sigma^2) = \frac{1}{y\cdot\sqrt{2\pi\sigma^2}} exp\left[-\frac{1}{2\sigma^2}(log(y)-\mu)^2\right],$$

where $\mu = 3.7$ is assumed to be known and σ^2 follows the $Inv-\chi^2(n,\tau^2)$ distribution, where:

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

(a)

From this point we can simulate 10,000 draws from the posterior of σ^2 using `rchisq()` that draws sample from the $\chi^2(n)$ distribution, given that:

$$X\sim\chi^2(n) \rightarrow \frac{1}{X}\sim Inv-\chi^2(n) \rightarrow \frac{\tau^2 n}{X}\sim Scaled-Inv-\chi^2(n),$$

since we have the scale parameter τ^2.

Therefore, in Figure 4 we can see the distribution of the simulated sample.

In Figure 5 there is the distribution of 10,000 draws from the theoretical $Inv-\chi^2(n,\tau^2)$ distribution that we simulated using `rinvchisq()`. We can see that both samples are equal, then we conlude that the generated sample from the χ^2 distribution is correct.

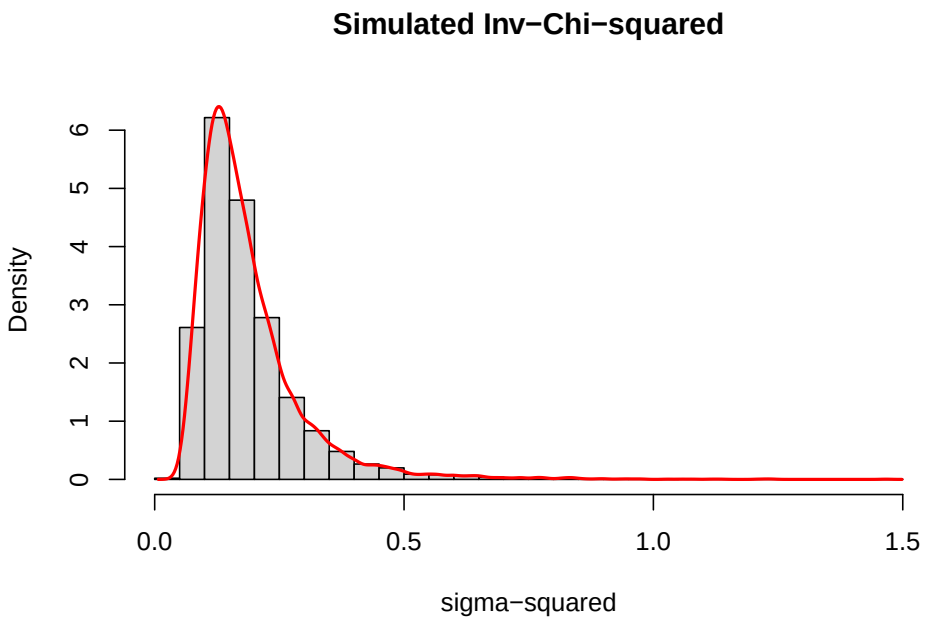

Figure 4: Density distribution of the simulated Inv-Chi-squared.

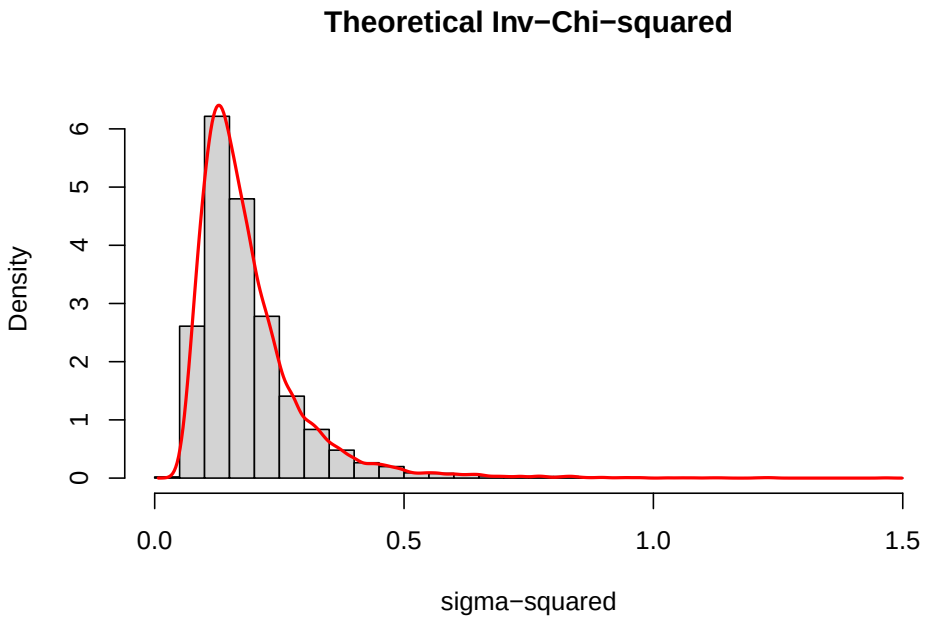

Figure 5: Density distribution of the Theoretical Inv-Chi-squared.

(b)

Using the the drawn sample from the posterior of σ^2 we then compute the posterior distribution of the Gini coefficient G, that is a measure of inequality, for the data set. To do so we use the Cumulative Distribution

Function (CDF) for the standard normal distribution and apply the following:

$$G = 2\phi\left(\frac{\sigma}{\sqrt{2}}\right) - 1,$$

we then plot the result in Figure 6. From the plot we conclude that the Gini coefficient is closer to 0 than to 1, therefore the income of the people in the sample are close to being equal.

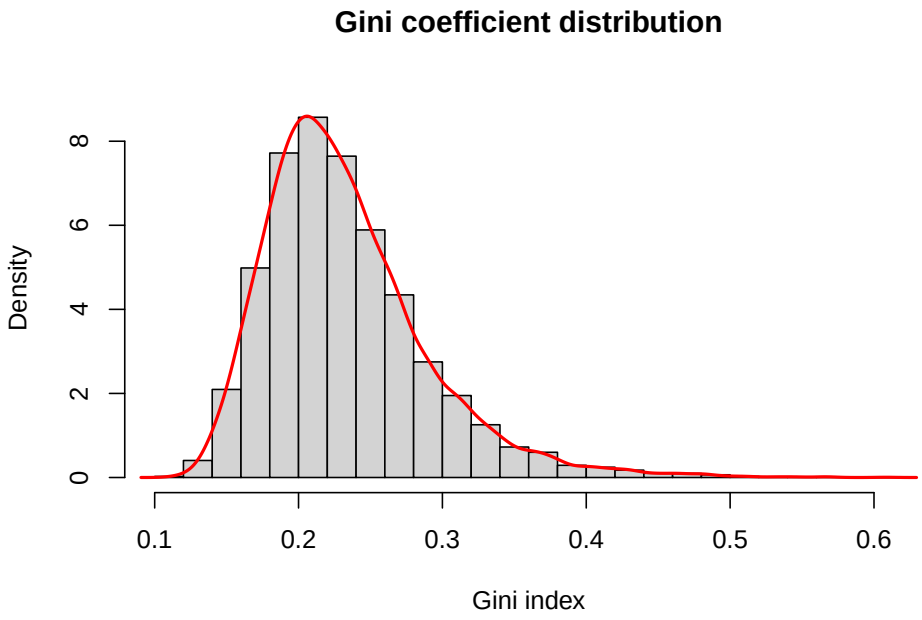

Figure 6: Posterior distribution of the Gini coefficient.

(c)

Now we use the posterior draws from b) to compute a 90% equal tail credible interval and a 90% Highest Posterior Density interval for G.

For the equal tail interval, we cut of 5% of the probability mass on each side using the `quantile()` function. We calculate the HPD interval by estimating the kernel density with the `density()` function. We sort the y values in descending order and calculate the area under the curve in a while loop, where we gradually increase the interval until the ratio of the current area and the total area reaches 0.9. Then we have a cut-off value, which we compare to the unsorted y values to find the x values on both sides of the distribution. These two are the interval limits. We also use the package `HDInterval` to check our results.

In Figure 7 we can see the kernel density estimates of G and its 90% intervals. The intervals are as follows:

Type	90% Interval
Equal Tail	$(0.1599, 0.3373)$
HPD Interval (Analytical)	$(0.1464, 0.3162)$
HPD Interval (Package)	$(0.1499, 0.3171)$

Table 1: *The 90% intervals for G.*

We can observe that the Equal Tail interval cuts the density curve at different heights, and it is lower on the right side of the mode. This is because of the skewedness of the distribution, which means that there is

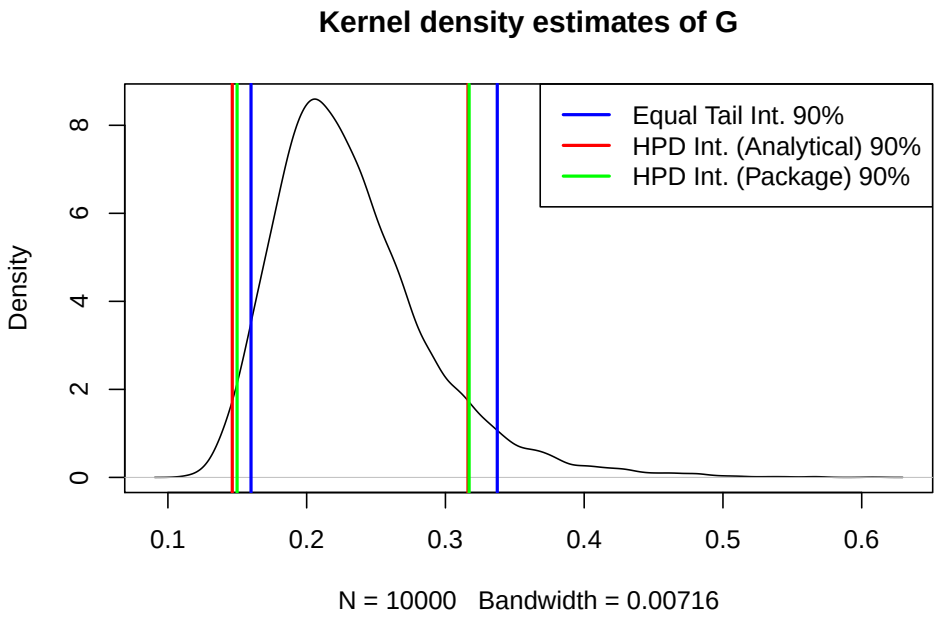

Figure 7: Comparison of equal tail credible intervals and highest posterior density intervals.

more probability mass on one side than on the other. The HPD interval by definition has to cut the density curve at the same height on both sides, hence it is a bit more to the left compared to the Equal Tail interval. There does not seem to be a significant difference between our implementation and the function imported from the packege `HDInterval`.

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

In this exercise we are given observed wind directions at a given location on ten different days. We assume that the data points are independent observations following the Von Mises distribution:

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

where $I_0(\kappa)$ is the modified Bessel function of the first kind of order zero, μ is the mean direction and κ is called the concentration parameter.

(a)

Let $\kappa \sim Exponential(\lambda = 1)$ we can compute the posterior of κ like this,

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

Once we have the posterior distribution, we plot it for different values of κ. In Figure 8 we can see the posterior distribution of κ that is centered around 2 and slightly right skewed.

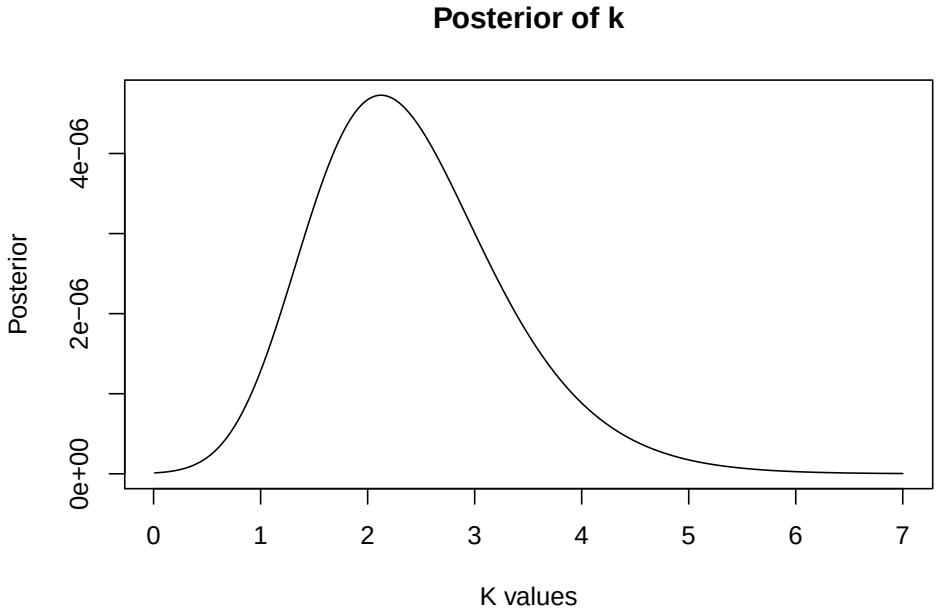

Figure 8: Posterior distribution of K.

(b)

From the posterior distribution of κ we find the mode, yielding a result of 2.12 - see code in the Appendix.

Appendix

```r
knitr::opts_chunk$set(echo=FALSE, eval=TRUE)
# R version
RNGversion('3.5.1')

# libraries
library(geoR)
library(HDInterval)
library(MESS)
library(ggplot2)
# Question 1. Bernoulli...again

# Parameters
n = 20
s = 5
f = n-s
alpha_0 = 2
beta_0 = 2

# (a)

# Seed
set.seed(1234567890)

draw_values <- seq(from=0, to=20000, by=250)
mean_values <- numeric(length(draw_values))
sd_values <- numeric(length(draw_values))
for (n_Draws in draw_values) {
  posterior_draw <- rbeta(n = n_Draws, shape1 = alpha_0 + s, shape2 = beta_0 + f)
  mean_values[which(draw_values==n_Draws)] <- mean(posterior_draw)
  sd_values[which(draw_values==n_Draws)] <- sd(posterior_draw)
}

# Plot - Number of draws Vs. mean
plot(draw_values, mean_values, main = "Mean Values", xlab="Number of draws",
     ylab="Mean", col="red", pch=16, ylim = c(0.28,0.3))

#Plot - Number of draws Vs. Sd
plot(draw_values, sd_values, main="Standar deviation values", ylim = c(0.08,0.1),
     xlab="Number of draws", ylab="Standard deviation", col="blue", pch=16)

#b

# Seed
set.seed(1234567890)

nDraws = 10000
posterior_draw = rbeta(n = nDraws, shape1 = alpha_0 + s, shape2 = beta_0 + f)
larger_than_03_sim = sum(posterior_draw > 0.3) / length(posterior_draw) #0.4333
larger_than_03_true = 1 - pbeta(q = 0.3, shape1 = alpha_0 + s, shape2 = beta_0 + f) #0.4399472

#c
```

```r
nDraws = 10000
log_odds = log(posterior_draw/(1-posterior_draw))

#Plot
hist(log_odds, freq = F, main="Log-odds distribution", xlab = "",
     ylim = c(0,1), breaks = 20, col="lightgrey")
lines(density(log_odds), col="red", lwd=2)

# Question 2. Log-normal distribution and the Gini coeficcient

Y = c(44, 25, 45, 52, 30, 63, 19, 50, 34, 67)
logY = log(Y)
n = length(Y)
mu = 3.7
tau2 = sum((logY - mu)^2) / n
#a

# simulate from posterior for sigma^2

# Seed
set.seed(1234567890)

X = rchisq(nDraws,n)
sigma2_sim = n*tau2/X
hist(sigma2_sim, freq = F, ylim=c(0,6.5), breaks=30, xlim=c(0,1.5),
     main = "Simulated Inv-Chi-squared", xlab="sigma-squared", col="lightgrey")
lines(density(sigma2_sim), col="red", lwd=2)

# theoretical posterior for sigma^2?

# Seed
set.seed(1234567890)

sigma_posterior_draw = rinvchisq(n = nDraws, df = n, scale = tau2)
hist(sigma_posterior_draw, freq = F, ylim=c(0,6.5), xlim=c(0,1.5), breaks = 30,
     main = "Theoretical Inv-Chi-squared", xlab="sigma-squared", col="lightgrey")
lines(density(sigma_posterior_draw), col="red", lwd=2)

#b

#Gini coefficient distribution
G = 2 * pnorm(sqrt(sigma2_sim / 2)) - 1
hist(G, freq = F, ylim=c(0,9), breaks=30, main="Gini coefficient distribution",
     xlab="Gini index", col="lightgrey")
lines(density(G), col="red", lwd=2)

#c

#90% equal tail
equal_tail = quantile(G, probs = c(0.05, 0.95)) #0.1599018, 0.3373707

#HPD
densx = density(G)$x
```

```r
densy = density(G)$y
# sort the density values (y) in descending order and add an id column
sorted = cbind(seq(1,length(densy)), sort(densy, decreasing = T))
plot(sorted, type = 'l')
total_area = auc(sorted[,1], sorted[,2]) # calculate the total area under the sorted curve

area = 0
i = 1
while (area/total_area < 0.9) {
  # calculate the area under the curve of the current data
  area = auc(sorted[1:(i+1),1], sorted[1:(i+1),2])
  i = i+1
}

cutoff = sorted[(i-1),2] # 1.743313 (subtract 1, because of the last unwanted loop iteration)
#cbind(density(G)$x, density(G)$y) # find the x values closest to cutoff on both sides
# the closest G-s from both tails from densx are 0.14638598, 0.31624232
HPD = c(0.14638598,0.31624232)

# alternative using package HDInterval
hdi(G, credMass = 0.9) #0.1498581, 0.3170765
hdi = c(0.1498581, 0.3170765)

# plot of intervals
plot(density(G), main = "Kernel density estimates of G")
abline(v = equal_tail, lwd = 2, col = "blue")
abline(v = HPD, lwd = 2, col = "red")
abline(v = hdi, lwd = 2, col = "green")
legend(x = "topright", legend=c("Equal Tail Int. 90%",
                                "HPD Int. (Analytical) 90%",
                                "HPD Int. (Package) 90%"),
       col = c("blue", "red", "green"), lwd = 2)

# 3 Bayesian inference for the concentration parameter in the von Mises distribution.

y = c(40, 303, 326, 285, 296, 314, 20, 308, 299, 296)
y_rad = c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
n = length(y_rad)
mu = 2.39


#a
k = seq(0.01,7,by = 0.01)

posterior_k = 1 / ((2*pi*besselI(k, nu = 0))^n) *
  exp(k * (sum(cos(y_rad-mu))-1))

plot(k,posterior_k, type = 'l', main = 'Posterior of k', xlab="K values", ylab="Posterior")

#b
k[which.max(posterior_k)] #2.12
```