Lab1_Bayesian

Alejo Perez Gomez(alepe026), Martynas Lukosevicius(marlu207)

07/04/2021

Daniel Bernoulli

a) Draw random numbers from posterior

Given:

- $y_1, ..., y_n|\theta \sim Bern(\theta)$
- s = 8
- n = 24
- prior = $Beta(\alpha_0, \beta_0)$
- $\alpha_0 = \beta_0 = 3$

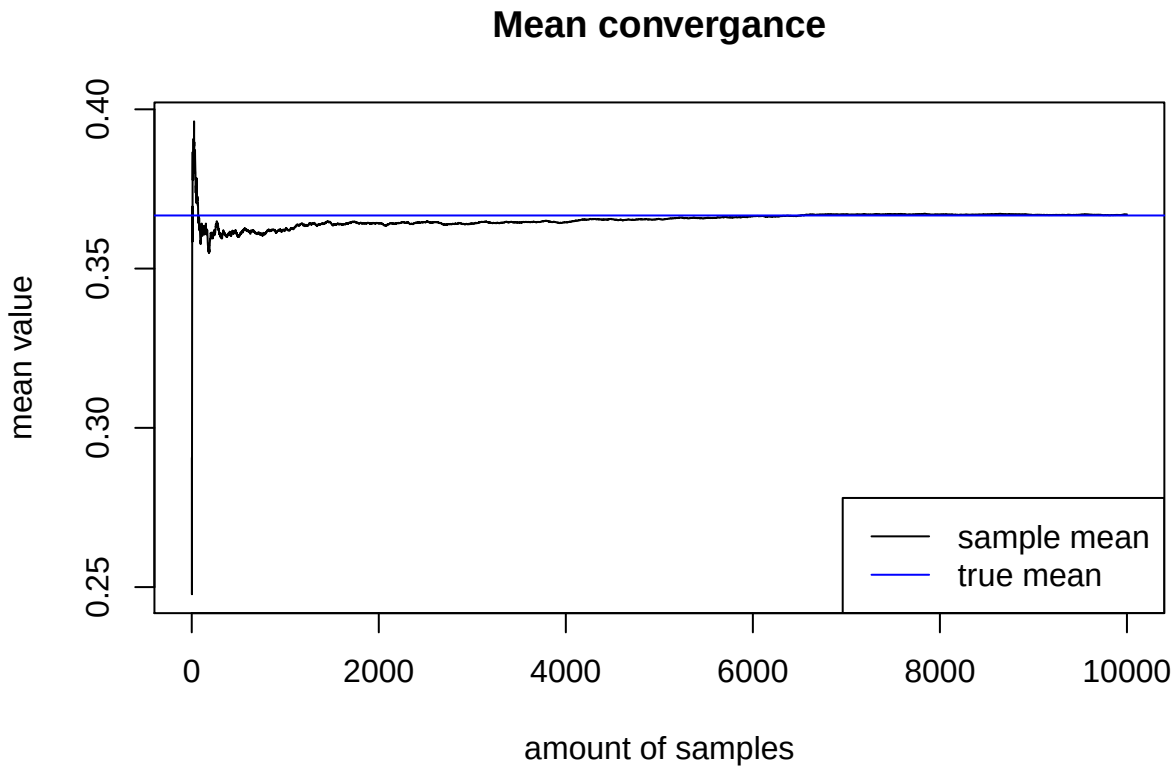
And

Posterior $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$

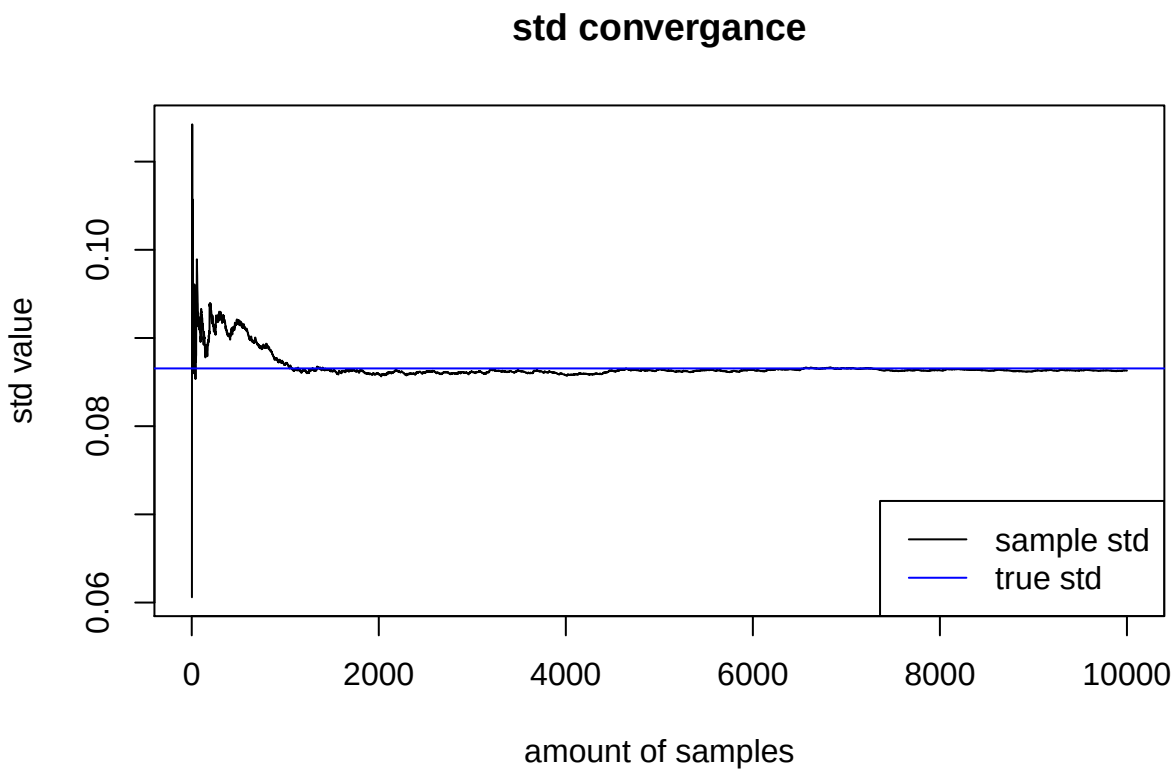
We will be drawing samples from the posterior and assessing convergence of parameters.

```
nDraws <- 10000
alpha_a <- 3
beta_a <- 3
s <- 8
n <- 24
f <- n - s

theta <- rbeta(n = nDraws,  alpha_a + s , beta_a + f)

beta_post <- beta_a + f
alpha_post <- alpha_a + s
```

```
mean_v <- vector(length = nDraws)
std_v <- vector(length = nDraws)

for(i in 1:nDraws){
  mean_v[i] <- mean(theta[1:i])
  std_v[i] <- sd(theta[1:i])
}

true_mean <- alpha_post/(alpha_post + beta_post)

plot(mean_v, type = "l", main = "Mean convergance", ylab = "mean value", xlab = "amount of samples")
abline(h = true_mean, col = "blue")
legend("bottomright", legend = c("sample mean", "true mean"), col = c("black", "blue"), lty = 1)
```




```
true_std <- (alpha_post*beta_post/((alpha_post+beta_post+1)*(alpha_post+beta_post)^2))^0.5

plot(std_v, type = "l", main = "std convergance", ylab = "std value", xlab = "amount of samples")
abline(h = true_std, col = "blue")
legend("bottomright", legend = c("sample std", "true std"), col = c("black", "blue"), lty = 1)
```



From the plots we can see that as a number of random draws from distribution increases, posterior mean and standard deviation converges to true value

b) compute the posterior probability $Pr(\theta > 0.4/y)$

The procedure for this probability calculation will be counting the ratio of appearance in the sample such that the parameter is greater than 0.4.

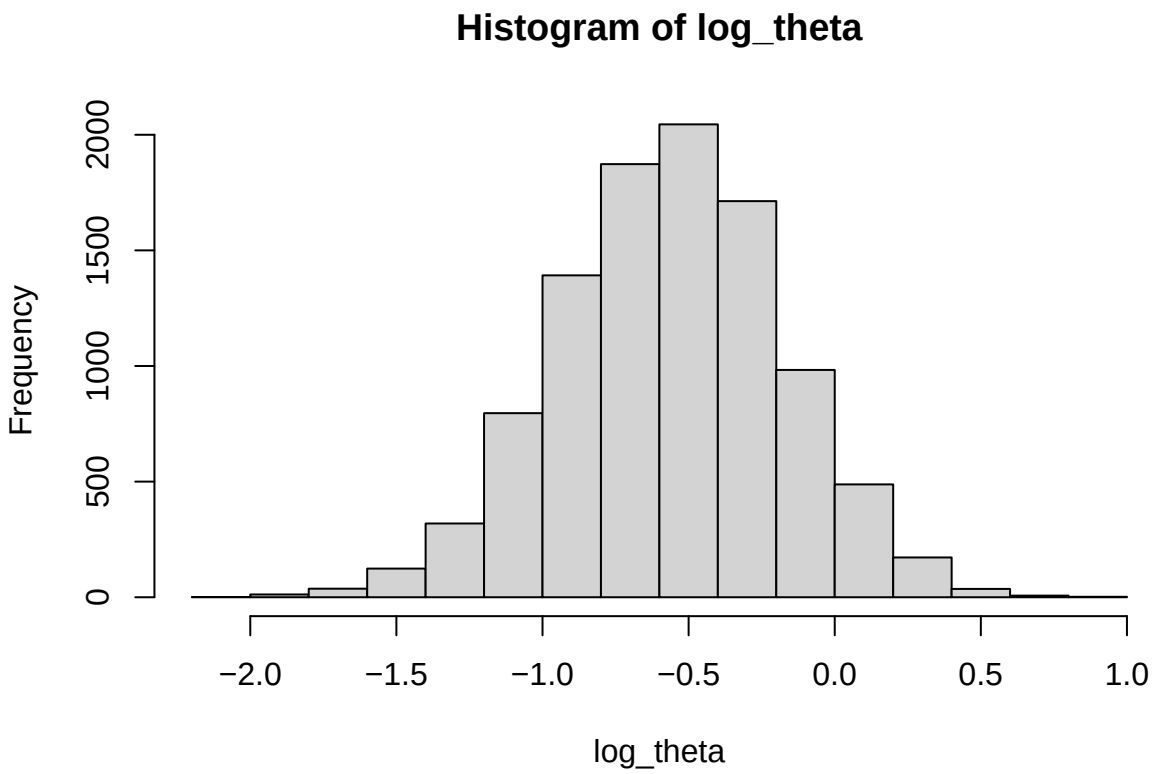
```
Pr_simul <- sum(theta>0.4)/nDraws
Pr_true <- pbeta(q = 0.4, shape1 = alpha_post, shape2 = beta_post, lower.tail = FALSE)
```

Simulated $Pr(\theta > 0.4|y) = 0.3452$ true value is 0.3426654

c) log-odds computation

The logg-odds distribution is to be computed by menas of the formula given.

```
log_theta <- log(theta/(1-theta))
hist(log_theta)
```



from the histogram we can see that log-odds are normally distributed with mean -0.5643239 and variance 0.1477607

Log-normal distribution and the Gini coeffcient.

a) simulation from posterior of variance

Given

- prior $p(\sigma^2)\alpha\frac{1}{\sigma^2}$
- posterior for σ^2 is $Inv-\chi^2(n,\tau^2)$
- where $\tau^2=\frac{\sum_{i=1}^{n}(logy_i-\mu)^2}{n}$ and $n=$ degrees of freedom
- posterior theoretical for scaled inverse chi squared is

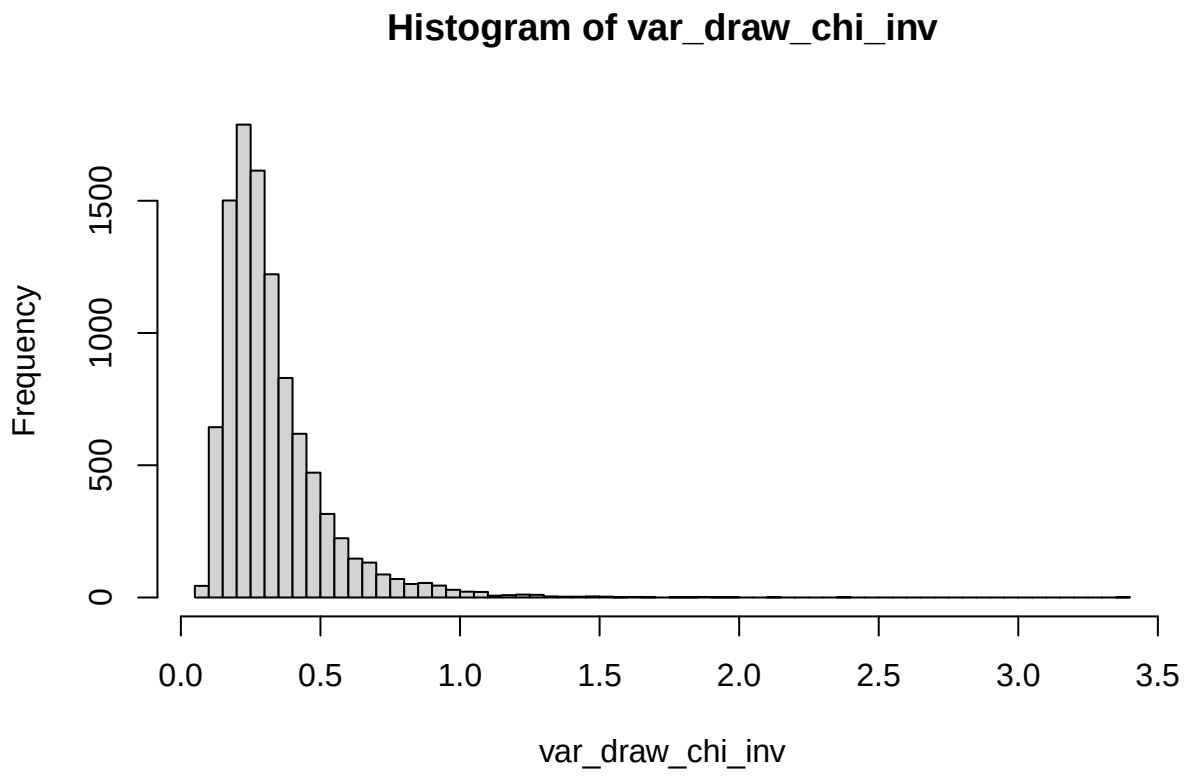
$$f(x;\nu,\tau^2)=\frac{(\tau^2\nu/2)^{\nu/2}}{\Gamma(\nu/2)}\ \frac{\exp\left[\frac{-\nu\tau^2}{2x}\right]}{x^{1+\nu/2}}$$

Let us first draw 1000 samples from the method proposed as:

- Draw
$$X\sim\chi^2(n-1)$$
- Compute
$$\sigma^2=\frac{(n-1)s^2}{X}$$
(draw from Inv scaled chi squared)

We will repeat this pattern as many times as we have draws

```
sample <- c( 38,20, 49, 58, 31, 70, 18, 56, 25, 78)
n <- length(sample)
nDraws <- 10000
mu <- 3.8
thau_sqr <- sum((log(sample) - mu)^2)/n

X <- rchisq(nDraws, n)
var_draw_chi_inv <- n*thau_sqr/X
sample_mean <- mean(var_draw_chi_inv)
true_mean <- (n*thau_sqr)/(n-2)
hist(var_draw_chi_inv, breaks = 100)
```

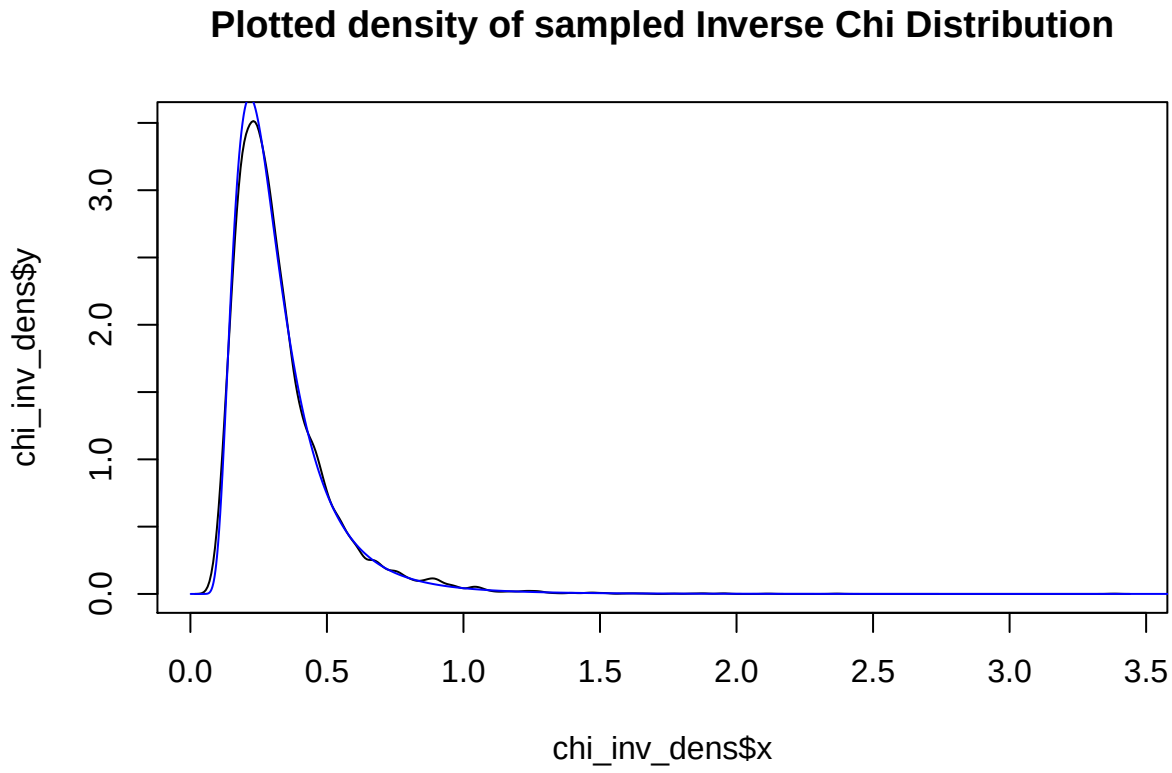


Above you can observe the histogram using this posterior sampling method. Sample mean of variance is 0.3280829 and true mean of Scaled inverse chi-squared distribution is 0.3263046

Now let us extract density points of our sample in order to plot the sample density and compare it with the teoretical distribution function for σ^2. As we can see, the posterior sampled distribution and the theoretical curve look alike.

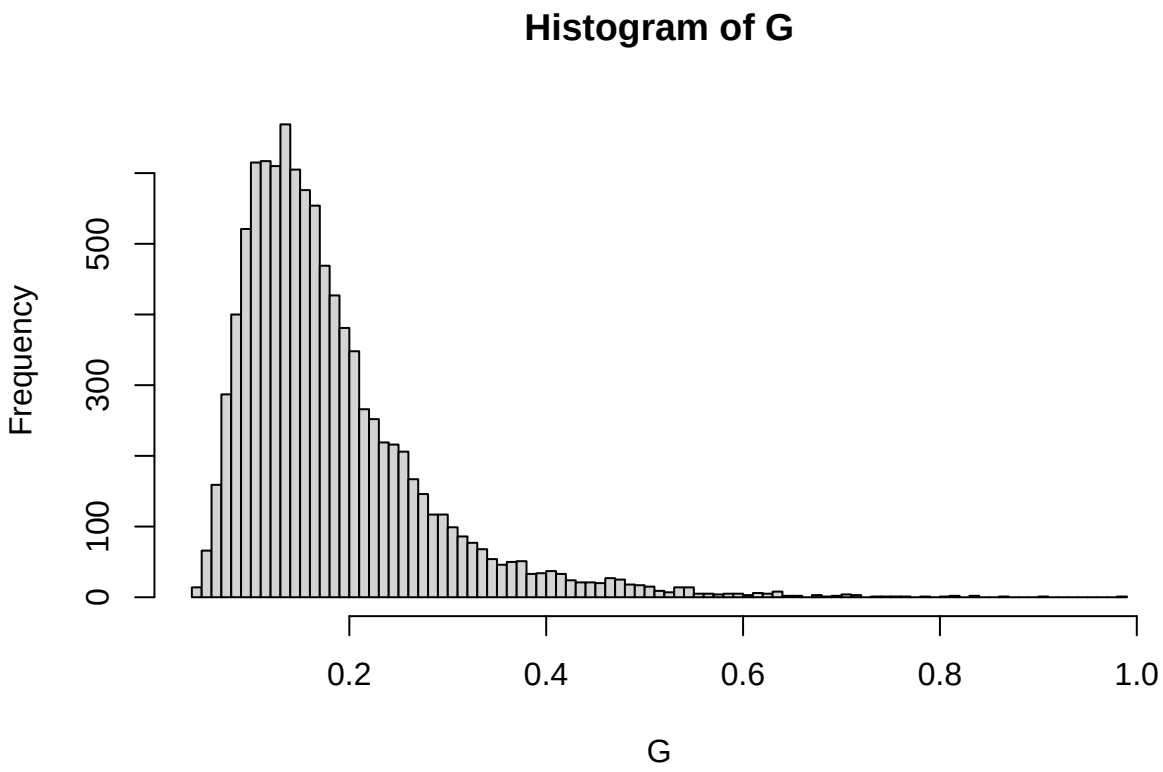
```
chi_inv_dens <- density(var_draw_chi_inv)

chi_inv_dens_func <- function(x, v, tau_sqr){
  num <- (((tau_sqr)*v/2)^(v/2)) * exp(-(v*tau_sqr)/(2*x))
  den <- gamma(v/2)*x^(1+v/2)
```

```
  return(num/den)
}

x <- seq(0,4,0.001)

y_chi_inv_dens  <- sapply(x, chi_inv_dens_func, v=n, tau_sqr=thau_sqr)

plot(x=chi_inv_dens$x , y=chi_inv_dens$y, main = "Plotted density of sampled Inverse Chi Distribution",

lines(x=x, y_chi_inv_dens, col= "blue")
```



b) Gini coefficient distribution

Let us calculate he Gini distribution according to the formula given used over the posterior variance draws and further histogram it. As observed, it resembles a log Normal distribution



c) Gini Confidence intervals

We will calculate the 90% equal tail by means of the *quantile* R in-built function.

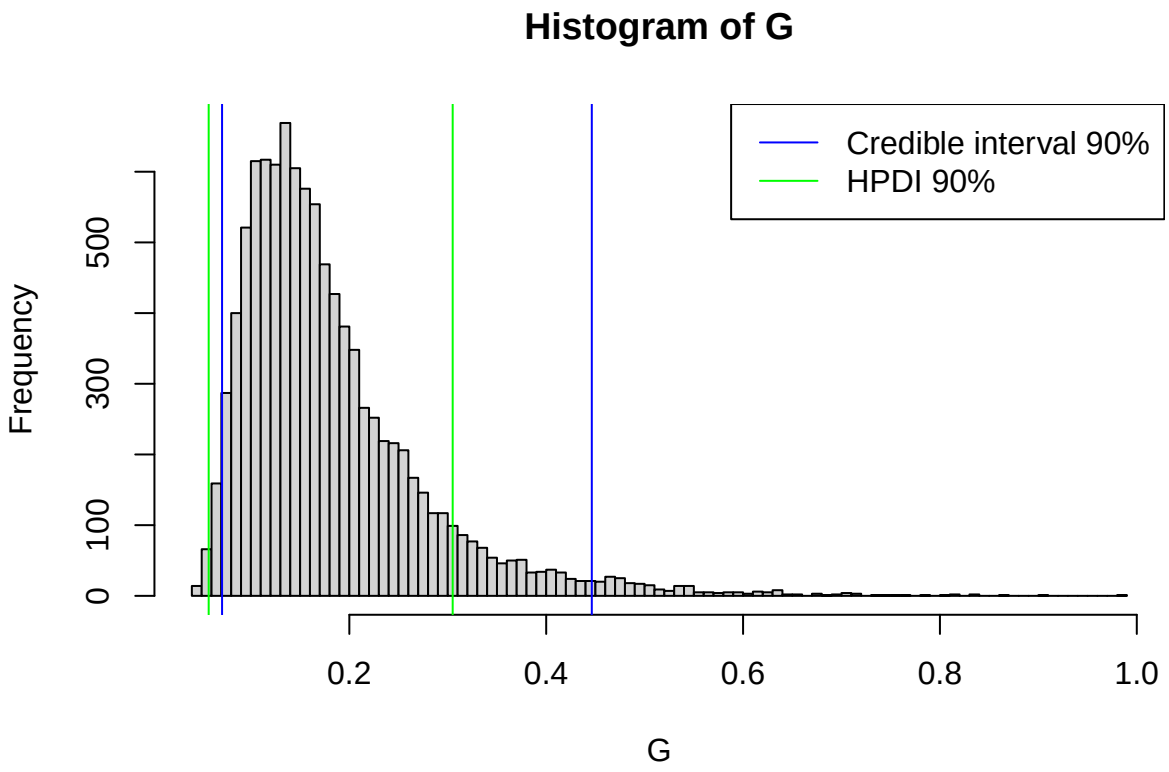
```
CI = quantile(x = G, probs = c(0.025, 0.975))
```

After that, to calculate de HPDI interval, we extract the density points, sort them by density values in decreasing order, and subset them such that gather the 90% of density value. Finally we will report the lower and higher limits of the interval.

```
dens <-  density(G)
dens_data <- data.frame(dens$x, dens$y/sum(dens$y))

dens_sum <- 0
ordered_dens <- dens_data[order(-dens_data$dens.y), ]

i <- 1
while(dens_sum<0.9){
  dens_sum <- dens_sum+ordered_dens$dens.y[i]
  i <- i+1
}

ordered_dens <- ordered_dens[1:(i-2),]

CI_dens <- c(min(ordered_dens$dens.x), max(ordered_dens$dens.x))
```

- 90% credible interval: 0.0707142, 0.4463191
- 90% HPDI: 0.0571009, 0.3049926

Values for both intervals are close although they differ by an order of magnitude of 0.01 and 0.1. That can be caused due to different approach of calculating the respecting interval. They estimate the underlying distribution in a different way.



Bayesian inference for the concentration parameter in the von Mises distribution

a)

Let us compute the posterior distribution of κ over a fine grid of this parameter.

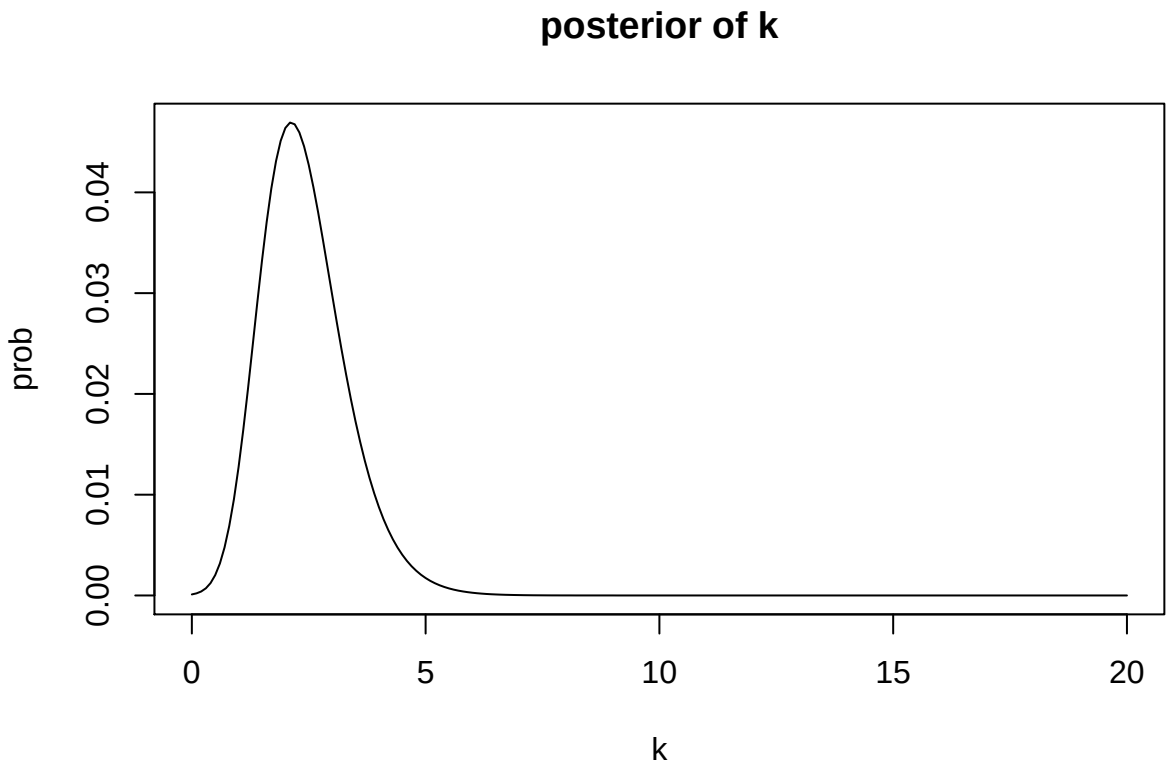
Data points follows von Mises distribution $p(y|\mu,\kappa) = \frac{exp[\kappa * cos(y-\mu)]}{2\pi I_0(\kappa)}$

- $\mu = 2.39$
- $\kappa \sim Exponential(\lambda = 1)$

$p(\kappa|\mu, y) = \prod p(y|\mu,\kappa) * Exponential(\lambda = 1)$

```
radians <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

mu <- 2.39
lambda <- 1
```

```r
kappa  <- seq(0,20,0.1)

posterior <- function(kappa){
  prior <- dexp(kappa, rate = 1/lambda)
  likelihood <- prod(exp(kappa*cos(radians-mu))/(2*pi*besselI(x = kappa, nu = 0)))
  return(prior*likelihood)
}


posterior_grid <- sapply(kappa, posterior)

posterior_grid_norm <- posterior_grid/sum(posterior_grid)

plot(kappa, posterior_grid_norm, type ="l", xlab = "k", ylab = "prob", main = "posterior of k")
```



b)

Finally, from this posterior distribution, let us take the posterior mode.

```r
mode <- kappa[which.max(posterior_grid_norm)]
```

Posterior mode is 2.1, its not very high meaning that variance around mean should be high

Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE)
nDraws <- 10000
alpha_a <- 3
beta_a <- 3
s <- 8
n <- 24
f <- n - s

theta <- rbeta(n = nDraws,  alpha_a + s , beta_a + f)

beta_post <- beta_a + f
alpha_post <- alpha_a + s

mean_v <- vector(length = nDraws)
std_v <- vector(length = nDraws)

for(i in 1:nDraws){
  mean_v[i] <- mean(theta[1:i])
  std_v[i] <- sd(theta[1:i])
}

true_mean <- alpha_post/(alpha_post + beta_post)

plot(mean_v, type = "l", main = "Mean convergance", ylab = "mean value", xlab = "amount of samples")
abline(h = true_mean, col = "blue")
legend("bottomright", legend = c("sample mean", "true mean"), col = c("black", "blue"), lty = 1)
true_std <- (alpha_post*beta_post/((alpha_post+beta_post+1)*(alpha_post+beta_post)^2))^0.5

plot(std_v, type = "l", main = "std convergance", ylab = "std value", xlab = "amount of samples")
abline(h = true_std, col = "blue")
legend("bottomright", legend = c("sample std", "true std"), col = c("black", "blue"), lty = 1)
Pr_simul <- sum(theta>0.4)/nDraws
Pr_true <- pbeta(q = 0.4, shape1 = alpha_post, shape2 = beta_post, lower.tail = FALSE)

log_theta <- log(theta/(1-theta))
hist(log_theta)
sample <- c( 38,20, 49, 58, 31, 70, 18, 56, 25, 78)
n <- length(sample)
nDraws <- 10000
mu <- 3.8
thau_sqr <- sum((log(sample) - mu)^2)/n

X <- rchisq(nDraws, n)
var_draw_chi_inv <- n*thau_sqr/X
sample_mean <- mean(var_draw_chi_inv)
true_mean <- (n*thau_sqr)/(n-2)
hist(var_draw_chi_inv, breaks = 100)
chi_inv_dens <- density(var_draw_chi_inv)

chi_inv_dens_func <- function(x, v, tau_sqr){
  num <- (((tau_sqr)*v/2)^(v/2)) * exp(-(v*tau_sqr)/(2*x))
```

```
  den <- gamma(v/2)*x^(1+v/2)
  return(num/den)
}

x <- seq(0,4,0.001)

y_chi_inv_dens  <- sapply(x, chi_inv_dens_func, v=n, tau_sqr=thau_sqr)

plot(x=chi_inv_dens$x , y=chi_inv_dens$y, main = "Plotted density of sampled Inverse Chi Distribution",

lines(x=x, y_chi_inv_dens, col= "blue")
G <- 2*pnorm(var_draw_chi_inv/sqrt(2))-1
hist(G, breaks = 100)

CI = quantile(x = G, probs = c(0.025, 0.975))


dens <-  density(G)
dens_data <- data.frame(dens$x, dens$y/sum(dens$y))

dens_sum <- 0
ordered_dens <- dens_data[order(-dens_data$dens.y), ]

i <- 1
while(dens_sum<0.9){
  dens_sum <- dens_sum+ordered_dens$dens.y[i]
  i <- i+1
}

ordered_dens <- ordered_dens[1:(i-2),]

CI_dens <- c(min(ordered_dens$dens.x), max(ordered_dens$dens.x))

hist(G, breaks = 100)
abline(v = CI[1], col="blue")
abline(v = CI[2],col="blue")
abline(v = CI_dens[1], col="green")
abline(v = CI_dens[2],col="green")

legend("topright", legend = c("Credible interval 90%", "HPDI 90%"),
       col=c("blue", "green"), lty=1)

radians <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

mu <- 2.39
lambda <- 1

kappa  <- seq(0,20,0.1)

posterior <- function(kappa){
  prior <- dexp(kappa, rate = 1/lambda)
  likelihood <- prod(exp(kappa*cos(radians-mu))/(2*pi*besselI(x = kappa, nu = 0)))
  return(prior*likelihood)
```

```
}


posterior_grid <- sapply(kappa, posterior)

posterior_grid_norm <- posterior_grid/sum(posterior_grid)

plot(kappa, posterior_grid_norm, type ="l", xlab = "k", ylab = "prob", main = "posterior of k")

mode <- kappa[which.max(posterior_grid_norm)]
```