Bayesian Learning Lab1

Nicolas Taba (nicta839) & Yuki Washio (timwa902)

06/04/2021

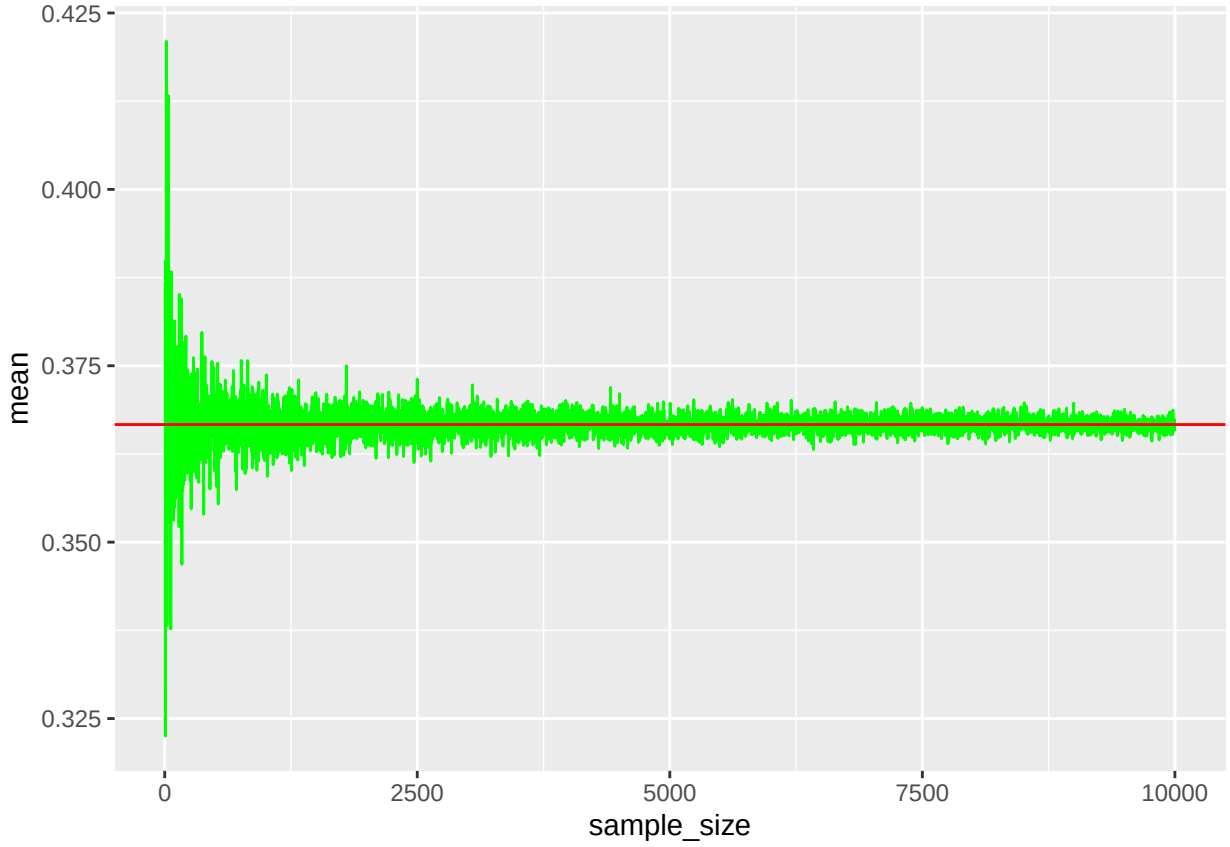
1. Daniel Bernoulli

1.a)

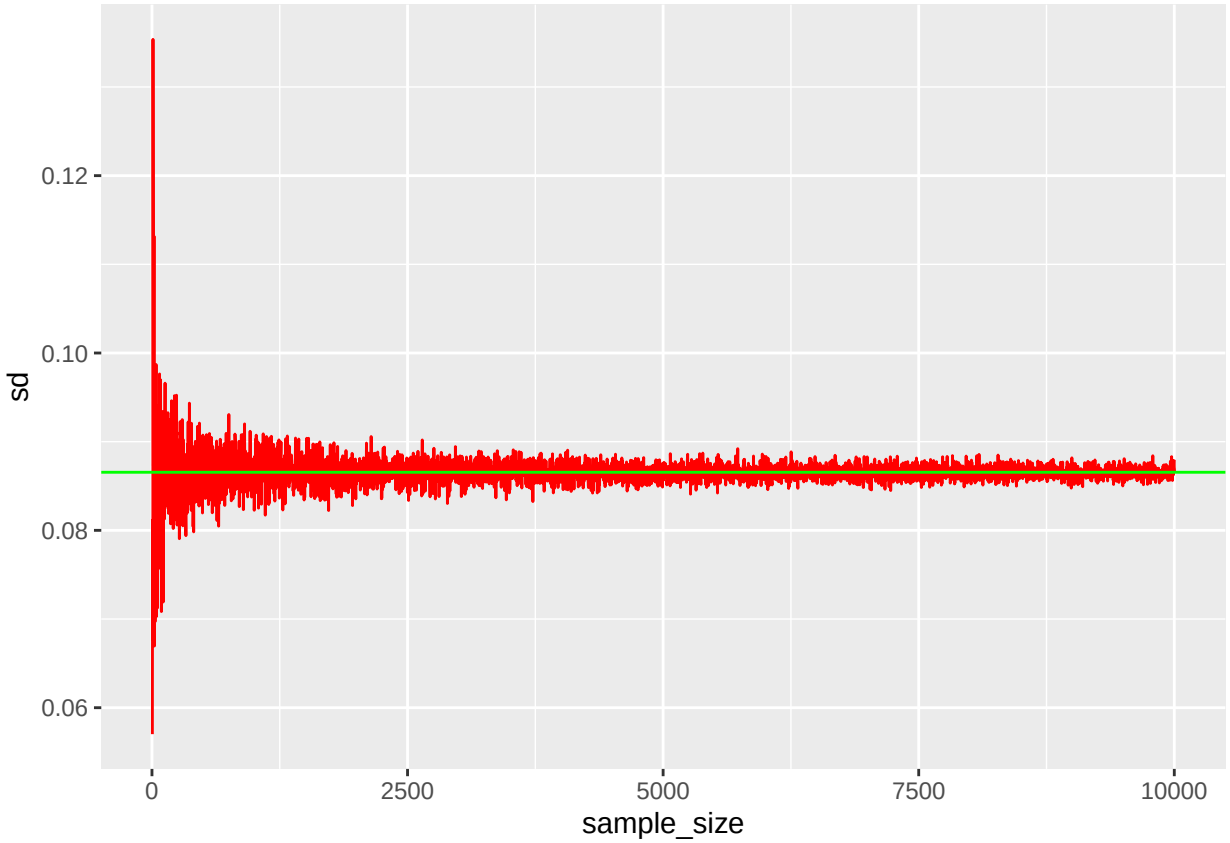
In this exercise, we draw random samples from the beta distribution given and verify graphically that the mean and standard deviation converge to their analytic values.

```
library(ggplot2)
set.seed(12345)

# Parameters given by the problem
s <- 8
n <- 24
f <- n - s
a_0 <- 3
b_0 <- a_0

# Derived values
a_pos <- a_0 + s
b_pos <- b_0 + f

# theoretical/ true values
t_mean <- a_pos/(a_pos + b_pos)
t_sd <- sqrt((a_pos * b_pos)/((a_pos + b_pos + 1) * (a_pos + b_pos)^2))

df <- data.frame("sample_size" = 1, "mean" = NA, "sd"  = NA)
draws <- seq(4, 10000, 2)
cnt <- 1

for(i in draws){
  rand_beta <- rbeta(i, a_pos, b_pos)
  df[cnt,] <- c(i, mean(rand_beta), sd(rand_beta))
  cnt <- cnt + 1
}


plot1 <- ggplot(df) +
  geom_line(mapping = aes(x = sample_size, y = mean), color = "green")+
  geom_hline(yintercept = t_mean, color = "red")

plot2 <- ggplot(df) +
  geom_line(mapping = aes(x = sample_size, y = sd), color = "red")+
  geom_hline(yintercept = t_sd, color = "green")
```

```
plot1
```


0.425
0.400
0.375
0.350
0.325
mean
0
2500
5000
7500
10000
sample_size

```
plot2
```




```
cat("The analytic value for the mean is: ", t_mean,"\nThe analytic value for the standard deviation is:
```

```
## The analytic value for the mean is:  0.3666667
## The analytic value for the standard deviation is:  0.08655079
```

In both graphs, the analytic solution for the mean and standard deviation are marked in horizontal lines colored red and green respectively. From these, we can see that as the sample size grows, we obtain results that are closer to the values to them.

1.b)

In this exercise, we are asked to to compute a posterior probability using a simulation of 10000 draws and compare it with R's inbuilt method.

```
# theoretical probability
set.seed(12345)
t_prob <- pbeta(0.4, a_pos, b_pos)

# Simulated probability
set.seed(12345)
sim_beta <- rbeta(10000, a_pos, b_pos)
sim_prob <- 1- sum((sim_beta <= 0.4))/ length(sim_beta)

cat("The theoretical probability is: ", t_prob,"\nThe calculated probability is: ", sim_prob)
```

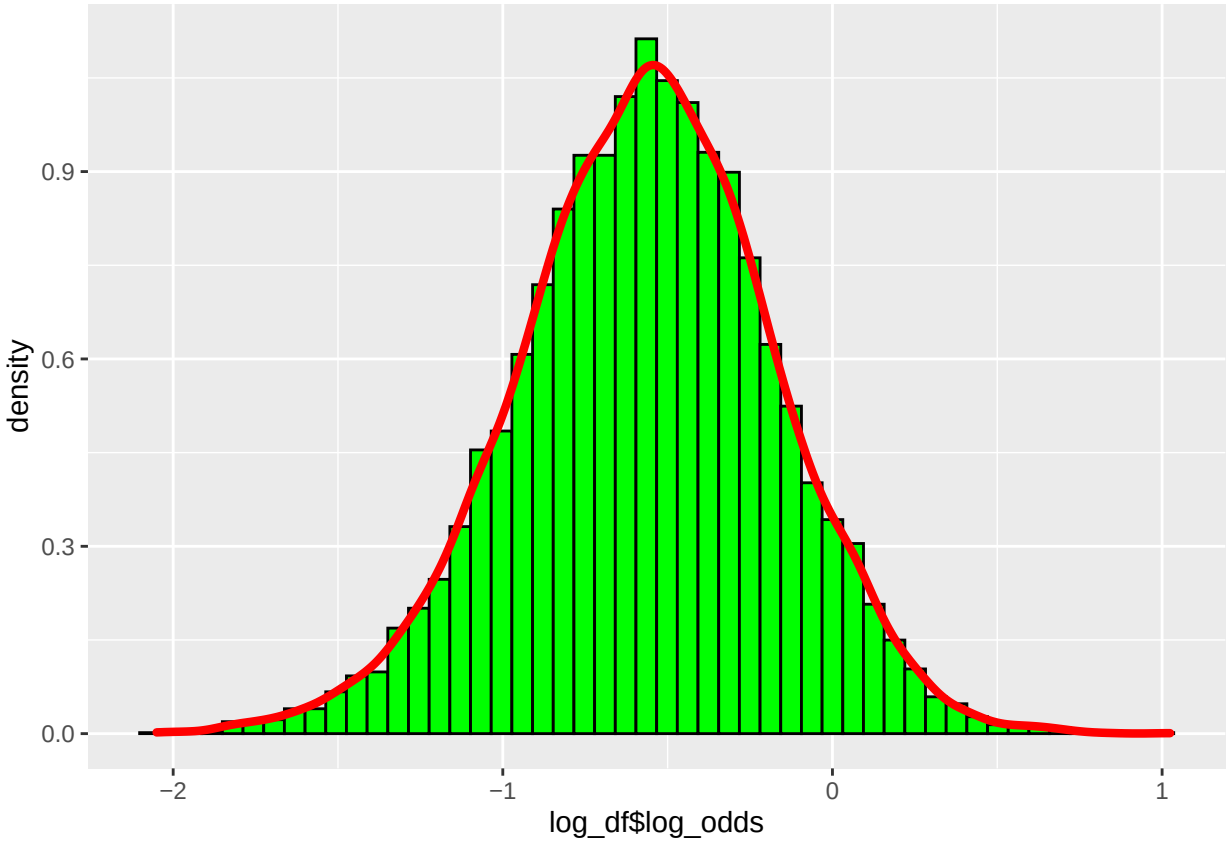
```
## The theoretical probability is:  0.6573346
## The calculated probability is:  0.3383
```

As we can see, the values are sensibly similar to each other

1.c)

In this exercise, we are asked to compute the log-odds posterior distribution using 10000 draws as previously.

```
log_odds <- log(sim_beta/ (1-sim_beta))

log_df <- as.data.frame(log_odds)

plot3 <- ggplot(log_df)+
  geom_histogram(mapping = aes(x = log_df$log_odds, y = ..density..), bins = 50, fill = "green", color =
  geom_density(mapping = aes(x = log_df$log_odds), color = "red", size = 1.5)

plot3
```



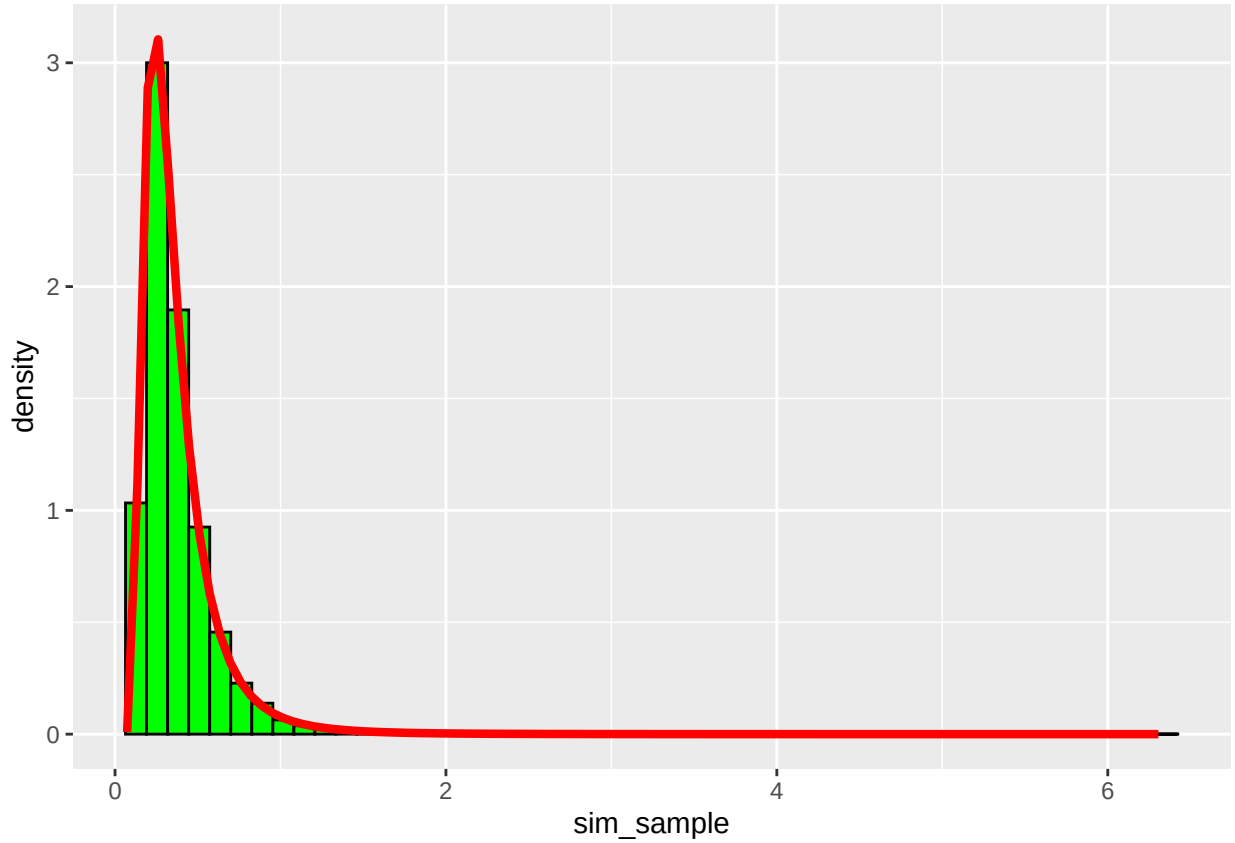
2. Log-normal distribution and the Gini coefficient

2.a)

In this exercise we perform 10000 draws from the calculated posterior and compare it with the theoretical scaled inverse chi-square posterior distribution.

```
# Data provided
obs <- c(38, 20, 49, 58, 31, 70, 18, 56, 25, 78)
mu <- 3.8
n <- length(obs) - 1

# https://en.wikipedia.org/wiki/Inverse-chi-squared_distribution
# https://en.wikipedia.org/wiki/Scaled_inverse_chi-squared_distribution
```

```r
# Information about the PDF and how to sample from the scaled inverse chi-square distribution
# X~scale-inv-chisq(nu, t^2) then X/(t^2 * nu) ~inv-chisq and the inv-chisq is just 1/X if X is chisq d

set.seed(12345)
rInvChisq <- function(draws, n, tau_sq){
  X <- rchisq(draws, n)
  sample <- (tau_sq * n)/X
  return(sample)
}

n_draw <- 10000
tau_sq <- sum((log(obs) - mu)^2)/n


sim_sample <- rInvChisq(n_draw, n, tau_sq)

# PDF of scaled inv chi-sq
# https://en.wikipedia.org/wiki/Scaled_inverse_chi-squared_distribution

scl_inv_chisq <- function(n, tau_sq, x){
  frac1 <- ((tau_sq * n/2)^(n/2))/(gamma(n/2))
  frac2 <- (exp((-n * tau_sq)/(2*x)))/(x^(1 + (n/2)))
  res <- frac1 * frac2
  return(res)
}


# Plotting to compare
df_sim_sample <- as.data.frame(sim_sample)

plot_chi_sq <- ggplot(df_sim_sample)+
  geom_histogram(mapping = aes(x = sim_sample, y = ..density..), bins = 50, fill = "green", color = "bl
  stat_function(mapping = aes(x = sim_sample), fun = scl_inv_chisq, args = list(n = n, tau_sq = tau_sq)

plot_chi_sq
```

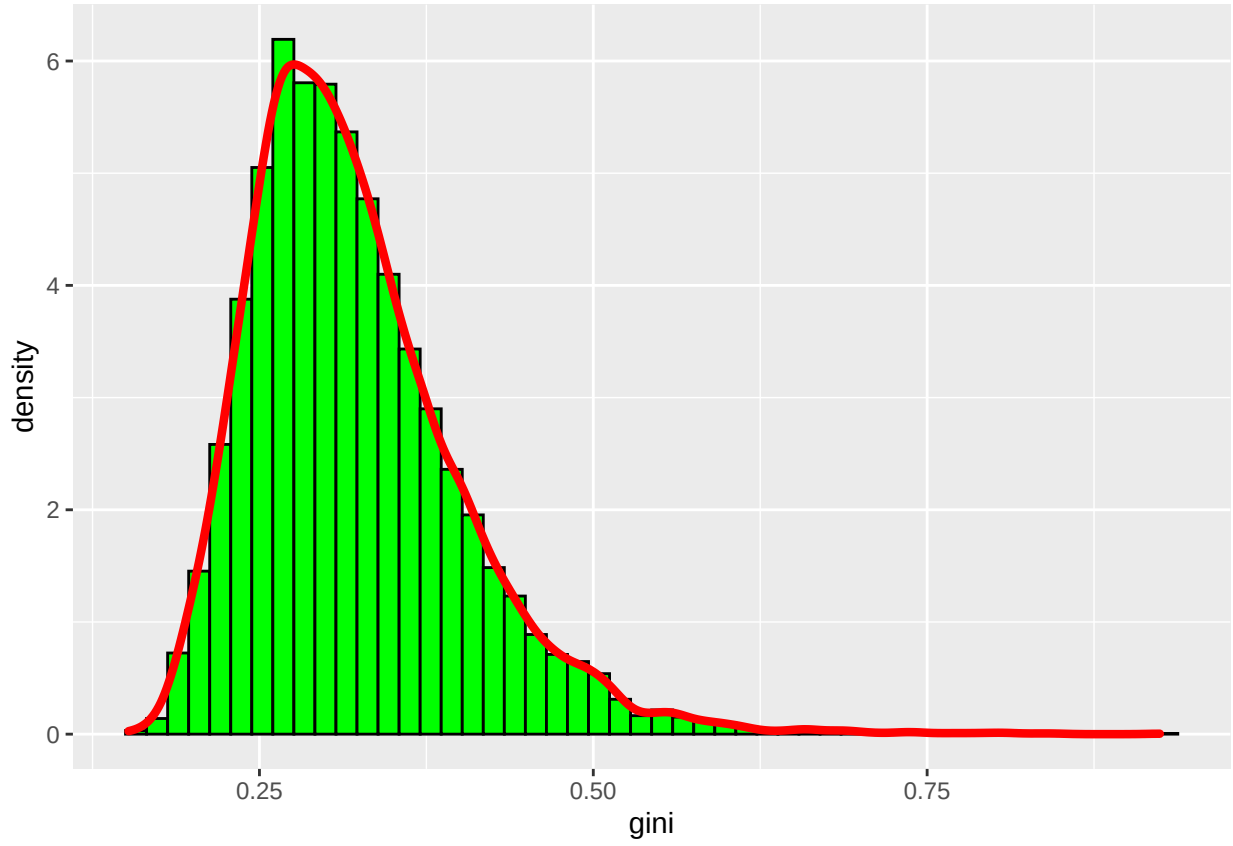


Our sample is marked as the green historgram while the theoretical scaled inverse chi-square posterior distribution is marked by the red line. They agree closely.

2.b)

Now we are asked to use the previous posterior draws to compute the the posterior distribution of the Gini coefficient.

```
set.seed(12345)
gini <- (2 * pnorm(sqrt(sim_sample)/sqrt(2))) - 1
df_gini <- as.data.frame(gini)

plot_gini <- ggplot(df_gini)+
  geom_histogram(mapping = aes(x = gini, y = ..density..), bins = 50, fill = "green", color = "black")+
  geom_density(mapping = aes(x = gini, y = ..density..), color = "red", size = 1.5)

plot_gini
```



2.c)

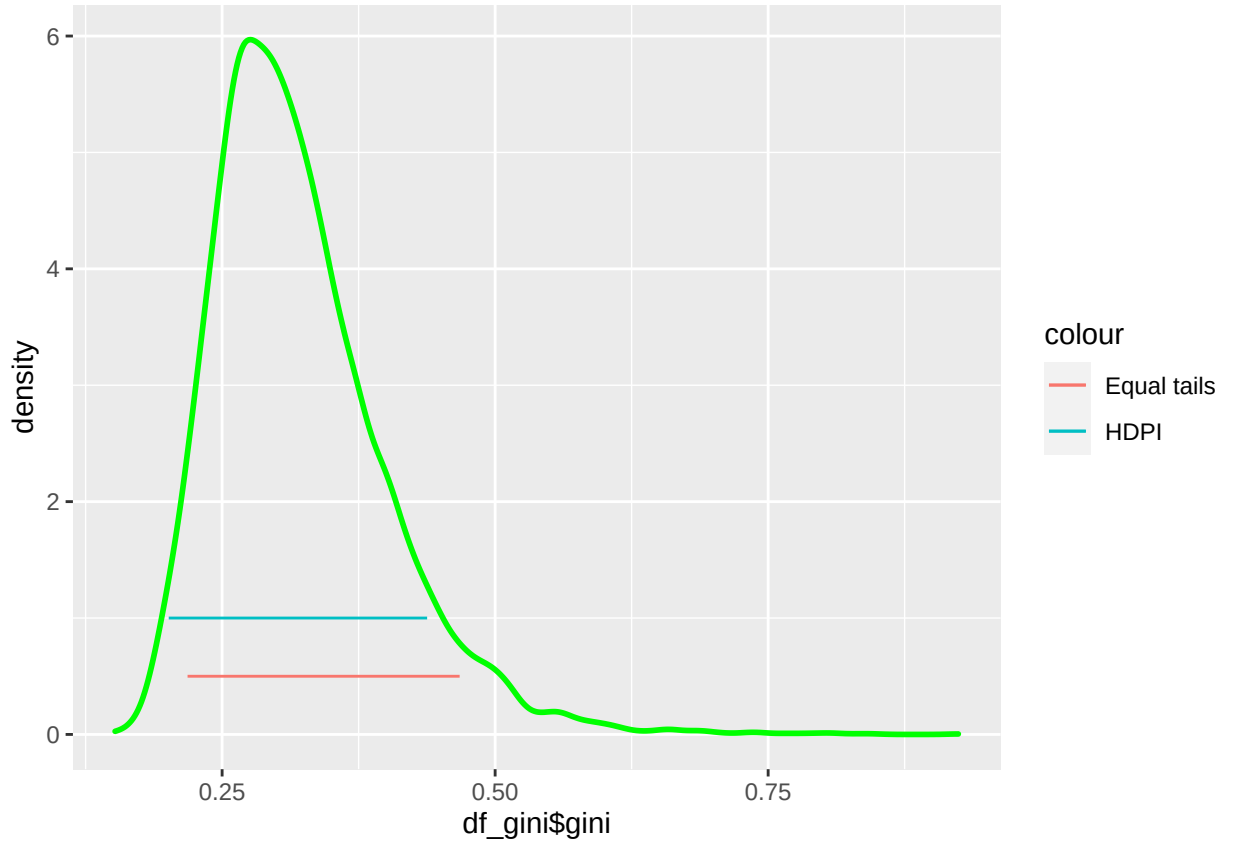
Here we are asked to compute the 90% equal tail interval as well as the 90% Highest Posterior Density Interval (HDPI) for the previously obtained distribution.

```
# Equal tail interval
equal_tail <- c(quantile(gini, 0.05), quantile(gini, 0.95))

#HPDI
kernel_density <- density(gini)
densities_df <- data.frame(x = kernel_density$x, density = kernel_density$y)
densities_df <- densities_df[order(densities_df$density, decreasing = TRUE), ]
# Must get the densities mass through cumulative sum
densities_df$cum_mass <- cumsum(densities_df$density)
cutoff <- 0.9 * tail(densities_df$cum_mass, 1)
densities_df <- densities_df[which(densities_df$cum_mass <= cutoff), ]

HDPI <- c(min(densities_df$x), max(densities_df$x))

cat("The equal tail interval is: ", equal_tail, "\nThe HDPI interval is: ", HDPI)
```

```
## The equal tail interval is:  0.2183606 0.4675031
## The HDPI interval is:  0.201116 0.4377371
```

```
plot_density <- ggplot()+
  geom_density(mapping = aes(x = df_gini$gini, y = ..density..), color = "green", size = 1)+
  geom_segment(aes(x = equal_tail[1], y = 0.5, xend = equal_tail[2], yend = 0.5, colour = "Equal tails"
```

```
  geom_segment(aes(x = HDPI[1], y = 1, xend = HDPI[2], yend = 1, colour = "HDPI"))

plot_density
```



We observe that the equal tails measure doesn't take well into account the skewness of the distribution whereas the HDPI more accurately takes into account the probability mass present at lower values of the Gini coefficient.

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

3.a)

In this exercise, we must first compute the formula for the posterior distribution of the κ term (concentration parameter) of the von Mises distribution. In order to do so, we first compute the likelihood term:

Likelihood for the von Mises distribution:

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

We then explicit the prior probability distribution as an exponential distribution with parameter λ:

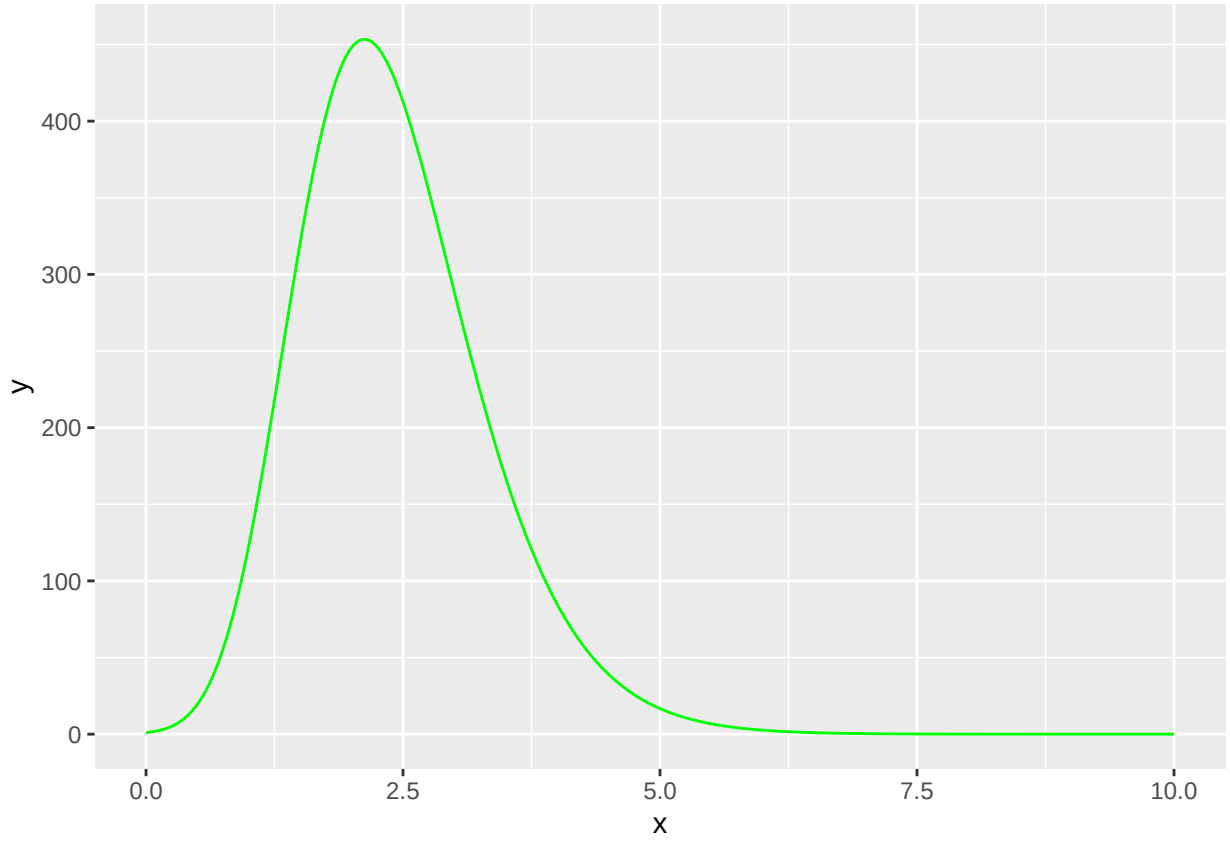
$$p(\kappa) = \lambda e^{-\lambda\kappa}$$

Finally, we compute the posterior probability distribution by eliminating constant terms that do not depend on κ. We obtain the following:

$$p(\kappa|\mu, y) = p(y|\mu, \kappa)p(\kappa) \propto \left(\frac{1}{I_0(\kappa)}\right)^n exp\left[\kappa\left(\sum_{i=1}^{n} cos(y_i - \mu) - \lambda\right)\right]$$

Now we can plot this distribution by varying κ from 0 to 10 in steps of 0.01:

```
# data given
data <- 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

# a)
posterior <- function(kappa, data){
  n <- length(data)
  fac1 <- (1/(besselI(kappa, 0)))^n
  fac2 <- exp(kappa * (sum(cos(data - mu)) - lambda))
  res <- fac1 * fac2
  return(res)
}

kappa_seq <- seq(0, 10, 0.01)

dist <- posterior(kappa_seq, data = data)
dist_df <- data.frame(x = kappa_seq ,y = dist)


plot_von_mises <- ggplot(dist_df)+
  geom_line(mapping = aes(x = x, y = y), color = "green")

plot_von_mises
```



3.b)

We now find the mode of this distribution, which is the value that the parameter takes when the distribution is maximal.

```
mode <- dist_df[which.max(dist_df$y),]$x

cat("The approximate posterior mode of kappa is: ", mode)
```

```
## The approximate posterior mode of kappa is:  2.12
```

Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
set.seed(12345)

# Parameters given by the problem
s <- 8
n <- 24
f <- n - s
a_0 <- 3
b_0 <- a_0

# Derived values
a_pos <- a_0 + s
```

```r
b_pos <- b_0 + f

# theoretical/ true values
t_mean <- a_pos/(a_pos + b_pos)
t_sd <- sqrt((a_pos * b_pos)/((a_pos + b_pos + 1) * (a_pos + b_pos)^2))

df <- data.frame("sample_size" = 1, "mean" = NA, "sd"  = NA)
draws <- seq(4, 10000, 2)
cnt <- 1

for(i in draws){
  rand_beta <- rbeta(i, a_pos, b_pos)
  df[cnt,] <- c(i, mean(rand_beta), sd(rand_beta))
  cnt <- cnt + 1
}


plot1 <- ggplot(df) +
  geom_line(mapping = aes(x = sample_size, y = mean), color = "green")+
  geom_hline(yintercept = t_mean, color = "red")

plot2 <- ggplot(df) +
  geom_line(mapping = aes(x = sample_size, y = sd), color = "red")+
  geom_hline(yintercept = t_sd, color = "green")

plot1
plot2

cat("The analytic value for the mean is: ", t_mean,"\nThe analytic value for the standard deviation is:

# theoretical probability
set.seed(12345)
t_prob <- pbeta(0.4, a_pos, b_pos)

# Simulated probability
set.seed(12345)
sim_beta <- rbeta(10000, a_pos, b_pos)
sim_prob <- 1- sum((sim_beta <= 0.4))/ length(sim_beta)

cat("The theoretical probability is: ", t_prob,"\nThe calculated probability is: ", sim_prob)

log_odds <- log(sim_beta/ (1-sim_beta))

log_df <- as.data.frame(log_odds)

plot3 <- ggplot(log_df)+
  geom_histogram(mapping = aes(x = log_df$log_odds, y = ..density..), bins = 50, fill = "green", color =
  geom_density(mapping = aes(x = log_df$log_odds), color = "red", size = 1.5)

plot3

# Data provided
obs <- c(38, 20, 49, 58, 31, 70, 18, 56, 25, 78)
```

```r
mu <- 3.8
n <- length(obs) - 1

# https://en.wikipedia.org/wiki/Inverse-chi-squared_distribution
# https://en.wikipedia.org/wiki/Scaled_inverse_chi-squared_distribution
# Information about the PDF and how to sample from the scaled inverse chi-square distribution
# X~scale-inv-chisq(nu, t^2) then X/(t^2 * nu) ~inv-chisq and the inv-chisq is just 1/X if X is chisq d

set.seed(12345)
rInvChisq <- function(draws, n, tau_sq){
  X <- rchisq(draws, n)
  sample <- (tau_sq * n)/X
  return(sample)
}

n_draw <- 10000
tau_sq <- sum((log(obs) - mu)^2)/n


sim_sample <- rInvChisq(n_draw, n, tau_sq)

# PDF of scaled inv chi-sq
# https://en.wikipedia.org/wiki/Scaled_inverse_chi-squared_distribution

scl_inv_chisq <- function(n, tau_sq, x){
  frac1 <- ((tau_sq * n/2)^(n/2))/(gamma(n/2))
  frac2 <- (exp((-n * tau_sq)/(2*x)))/(x^(1 + (n/2)))
  res <- frac1 * frac2
  return(res)
}


# Plotting to compare
df_sim_sample <- as.data.frame(sim_sample)

plot_chi_sq <- ggplot(df_sim_sample)+
  geom_histogram(mapping = aes(x = sim_sample, y = ..density..), bins = 50, fill = "green", color = "bl
  stat_function(mapping = aes(x = sim_sample), fun = scl_inv_chisq, args = list(n = n, tau_sq = tau_sq)

plot_chi_sq
set.seed(12345)
gini <- (2 * pnorm(sqrt(sim_sample)/sqrt(2))) - 1
df_gini <- as.data.frame(gini)

plot_gini <- ggplot(df_gini)+
  geom_histogram(mapping = aes(x = gini, y = ..density..), bins = 50, fill = "green", color = "black")+
  geom_density(mapping = aes(x = gini, y = ..density..), color = "red", size = 1.5)

plot_gini

# Equal tail interval
equal_tail <- c(quantile(gini, 0.05), quantile(gini, 0.95))
```

```r
#HPDI
kernel_density <- density(gini)
densities_df <- data.frame(x = kernel_density$x, density = kernel_density$y)
densities_df <- densities_df[order(densities_df$density, decreasing = TRUE), ]
# Must get the densities mass through cumulative sum
densities_df$cum_mass <- cumsum(densities_df$density)
cutoff <- 0.9 * tail(densities_df$cum_mass, 1)
densities_df <- densities_df[which(densities_df$cum_mass <= cutoff), ]

HDPI <- c(min(densities_df$x), max(densities_df$x))

cat("The equal tail interval is: ", equal_tail, "\nThe HDPI interval is: ", HDPI)

plot_density <- ggplot()+
  geom_density(mapping = aes(x = df_gini$gini, y = ..density..), color = "green", size = 1)+
  geom_segment(aes(x = equal_tail[1], y = 0.5, xend = equal_tail[2], yend = 0.5, colour = "Equal tails"))
  geom_segment(aes(x = HDPI[1], y = 1, xend = HDPI[2], yend = 1, colour = "HDPI"))

plot_density
# data given
data <- 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

# a)
posterior <- function(kappa, data){
  n <- length(data)
  fac1 <- (1/(besselI(kappa, 0)))^n
  fac2 <- exp(kappa * (sum(cos(data - mu)) - lambda))
  res <- fac1 * fac2
  return(res)
}

kappa_seq <- seq(0, 10, 0.01)

dist <- posterior(kappa_seq, data = data)
dist_df <- data.frame(x = kappa_seq ,y = dist)


plot_von_mises <- ggplot(dist_df)+
  geom_line(mapping = aes(x = x, y = y), color = "green")

plot_von_mises
mode <- dist_df[which.max(dist_df$y),]$x

cat("The approximate posterior mode of kappa is: ", mode)
```