Bayesian Learning, Computer Lab 1

Arash Haratian (araha147) , Daniel Díaz-Roncero Gonzalez (dandi692)

2023-04-04

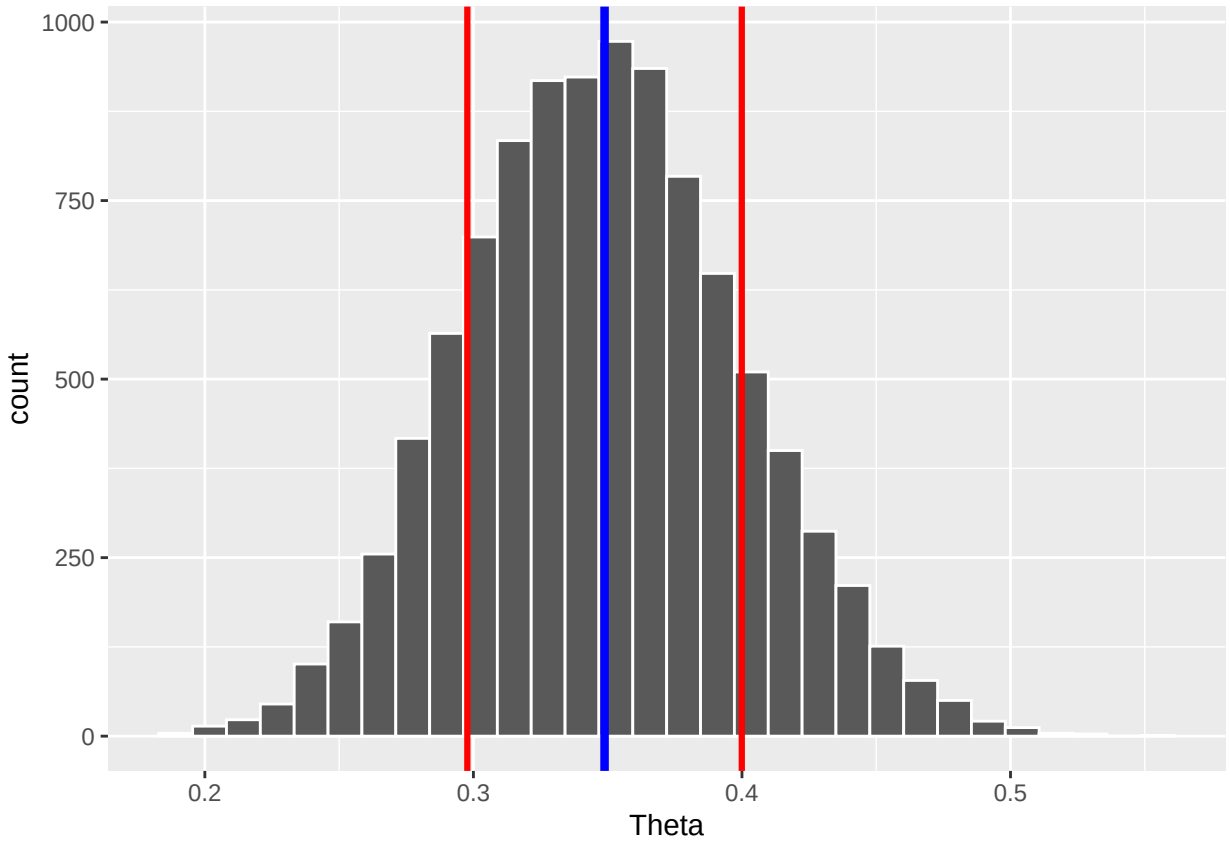
1. Daniel Bernoulli

1a

```
s <- 22
n <- 70
f <- n-s
alpha0 <- 8
beta0 <- 8

#a
nDraws <- 10000
theta <- rbeta(nDraws,alpha0+s,beta0+f)
df <- data.frame(x=1:nDraws,y=theta, csmean=cumsum(theta)/1:10000)
csstd = vector(length=nDraws)
for(i in 1:nDraws){
  csstd[i] = sd(theta[1:i])
}
df["csstd"] = csstd
sstd_expected <- sqrt((alpha0 + s)*(beta0 + f)/((alpha0 + s + beta0 + f)^2 * (alpha0 + s + beta0 + f + 1
smean_expected <- (alpha0 + s)/(alpha0 + s + beta0 + f)
ggplot(df) +
  geom_histogram(aes(y),color = "white") +
  geom_vline(xintercept = smean_expected,color = "blue",size=1.5) +
  geom_vline(xintercept = smean_expected - sstd_expected,color = "red",size=1.1) +
  geom_vline(xintercept = smean_expected + sstd_expected,color = "red",size=1.1) +
  xlab("Theta")
```

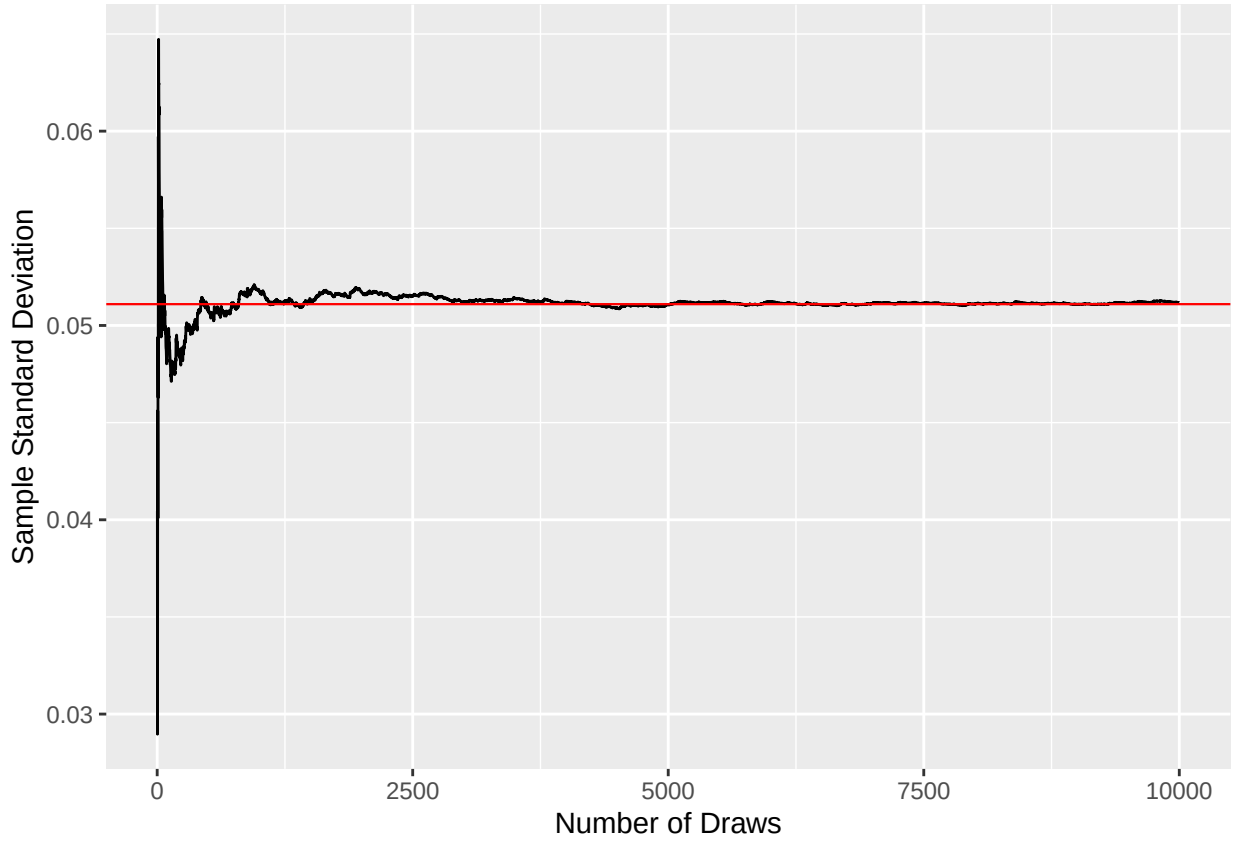
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



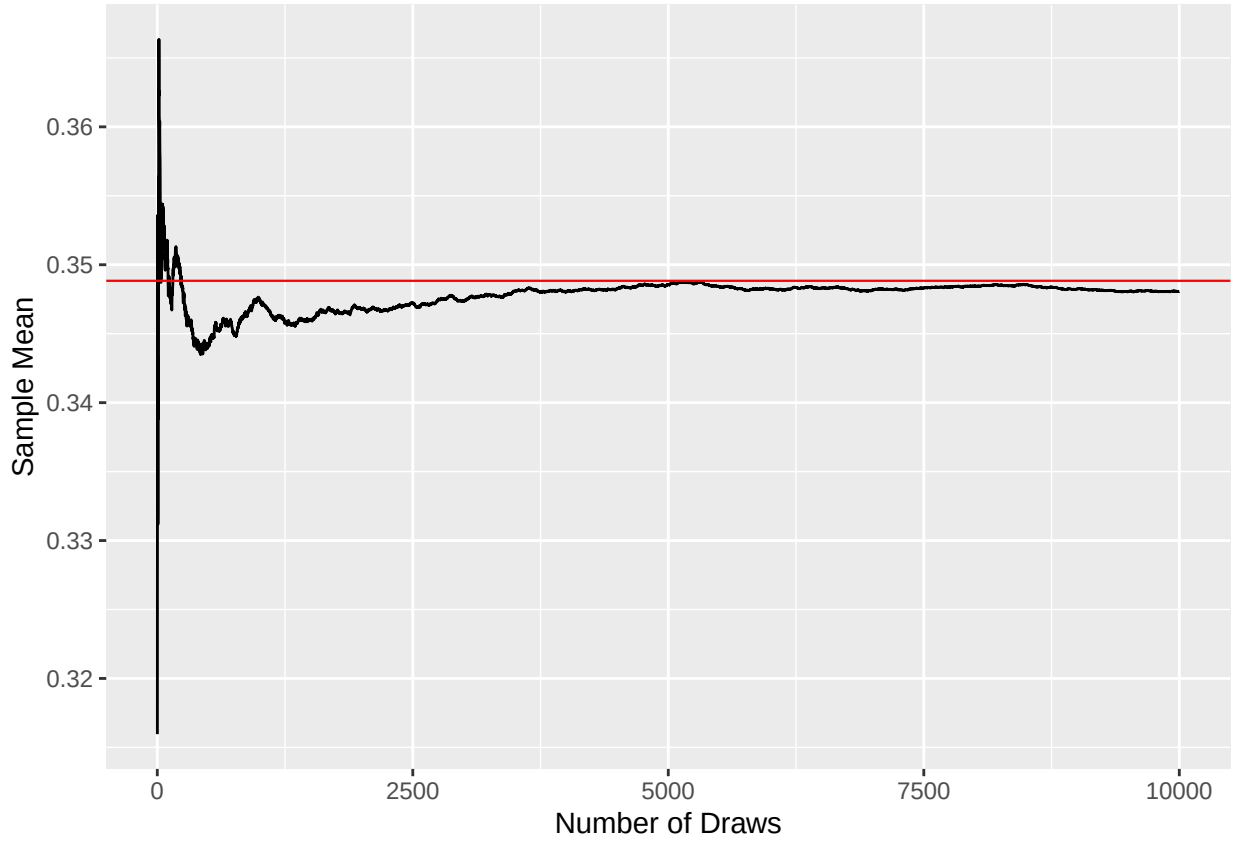
The above plot is the histogram of the sampled θ from the posterior density. Note that the blue line is the expected mean for the posterior distribution and the two red lines are one expected standard deviation to the right and left of the expected mean.

```
ggplot(df[2:nDraws,]) +
  geom_line(aes(x=x,y=csstd)) +
  geom_hline(yintercept = sstd_expected,color = "red",size=0.4) +
  ylab("Sample Standard Deviation") +
  xlab("Number of Draws")
```



The plot above shows that the sample standard deviation tends to the expected standard deviation when the number of draws grows.

```
ggplot(df) +
  geom_line(aes(x=x,y=csmean)) +
  geom_hline(yintercept = smean_expected,color = "red",size=0.4) +
  ylab("Sample Mean") +
  xlab("Number of Draws")
```



The plot above shows that the sample mean tends to the expected mean when the number of draws grows.

1b

```
#b

pr <- sum(theta>0.3)/length(theta)

pr_beta <- 1 - pbeta(0.3,alpha0+s,beta0+f)
```

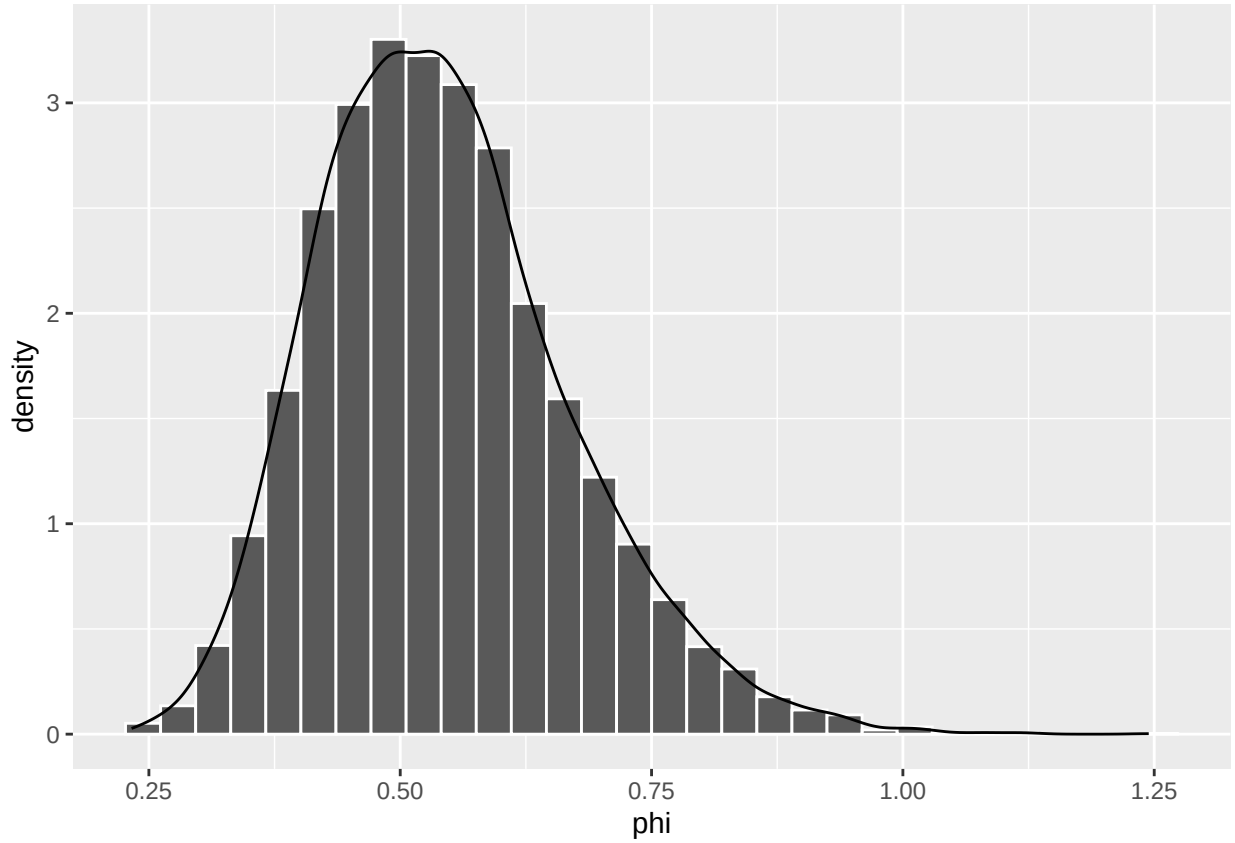
The posterior probability $P(\theta > 0.3|y)$ is `0.8221``` and the theoretical value from the Beta distribution is0.8285936“‘, both are pretty close.

1c

The plot below shows the posterior for the odds $\phi = \frac{\theta}{1-\theta}$

```
#c

phi <- theta/(1-theta)
df["phi"] <- phi
ggplot(df) +
  geom_histogram(aes(phi, after_stat(density)),color = "white") +
  geom_density(aes(phi))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

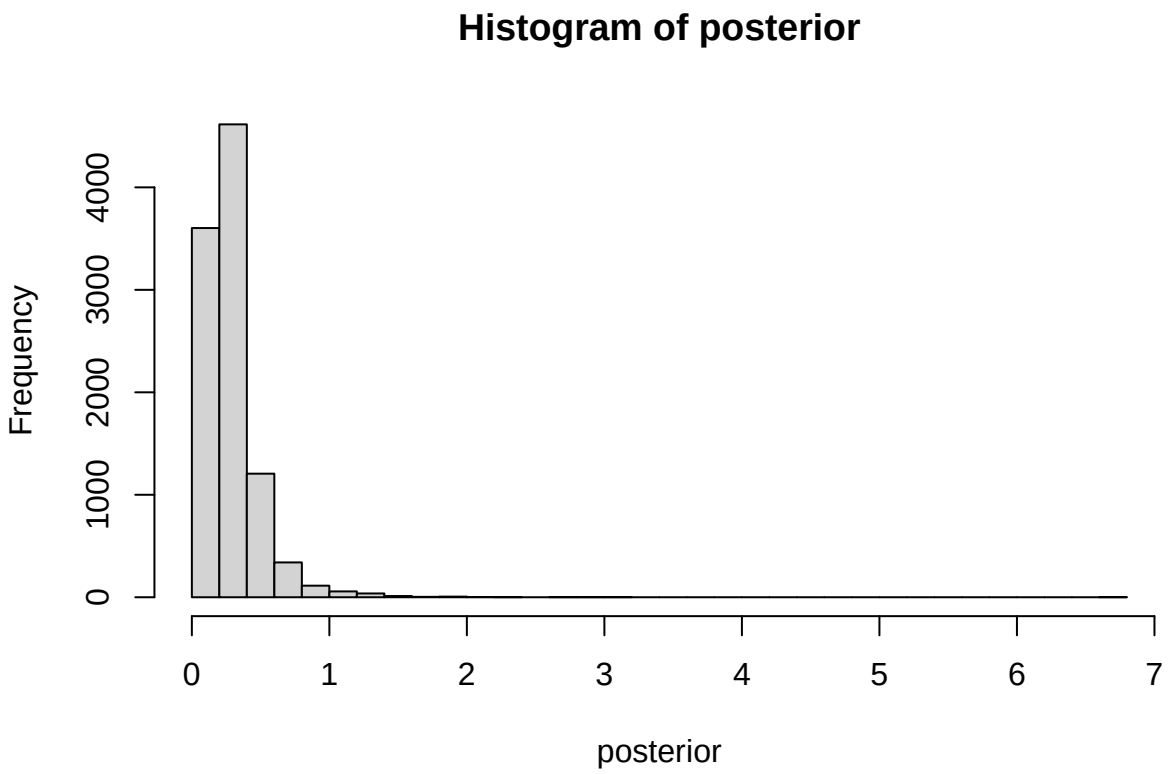


2. Log-normal distribution and the Gini coefficient

2a

The plot below is the representation of 10000 random values from the posterior.

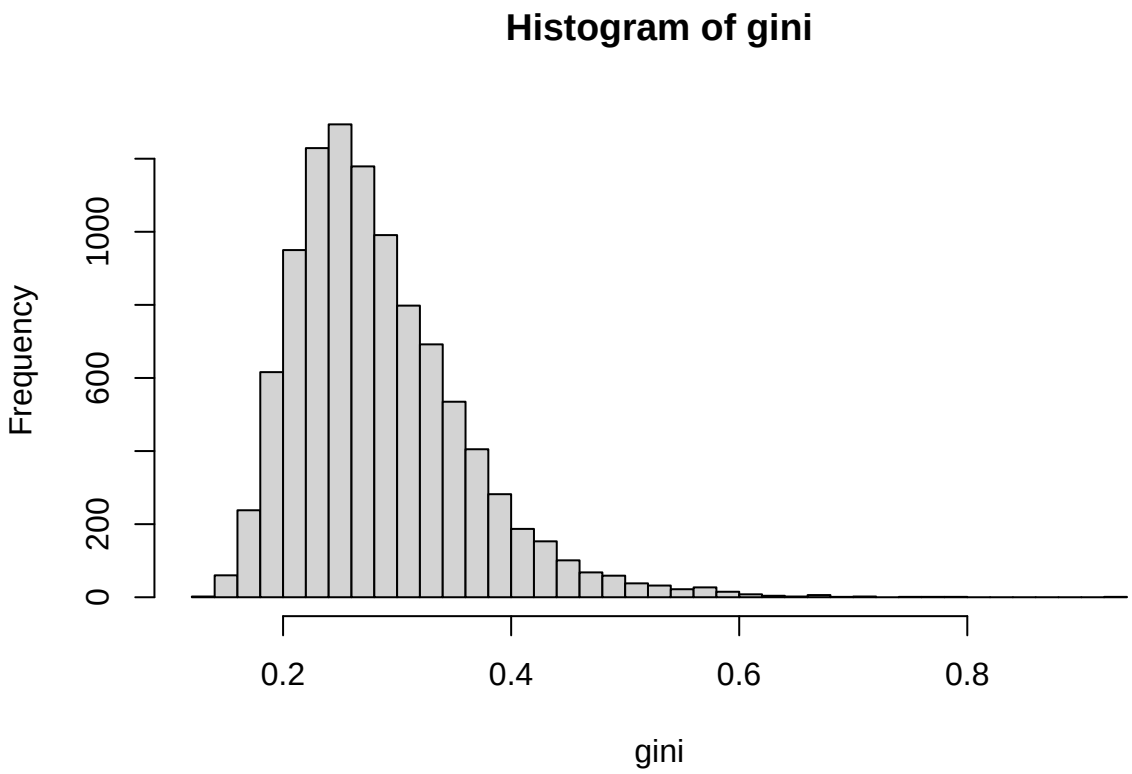
```
#a
y <- c(33, 24, 48, 32, 55, 74, 23, 17)
mu <- 3.6
n <- 8
nDwaws <- 10000

tau_squared <- sum((log(y)-mu)^2)/n

posterior <- n*tau_squared / rchisq(nDwaws,n)

hist(posterior,breaks=30)
```



2b

The posterior distribution of the Gini coefficient is represented in the following histogram.

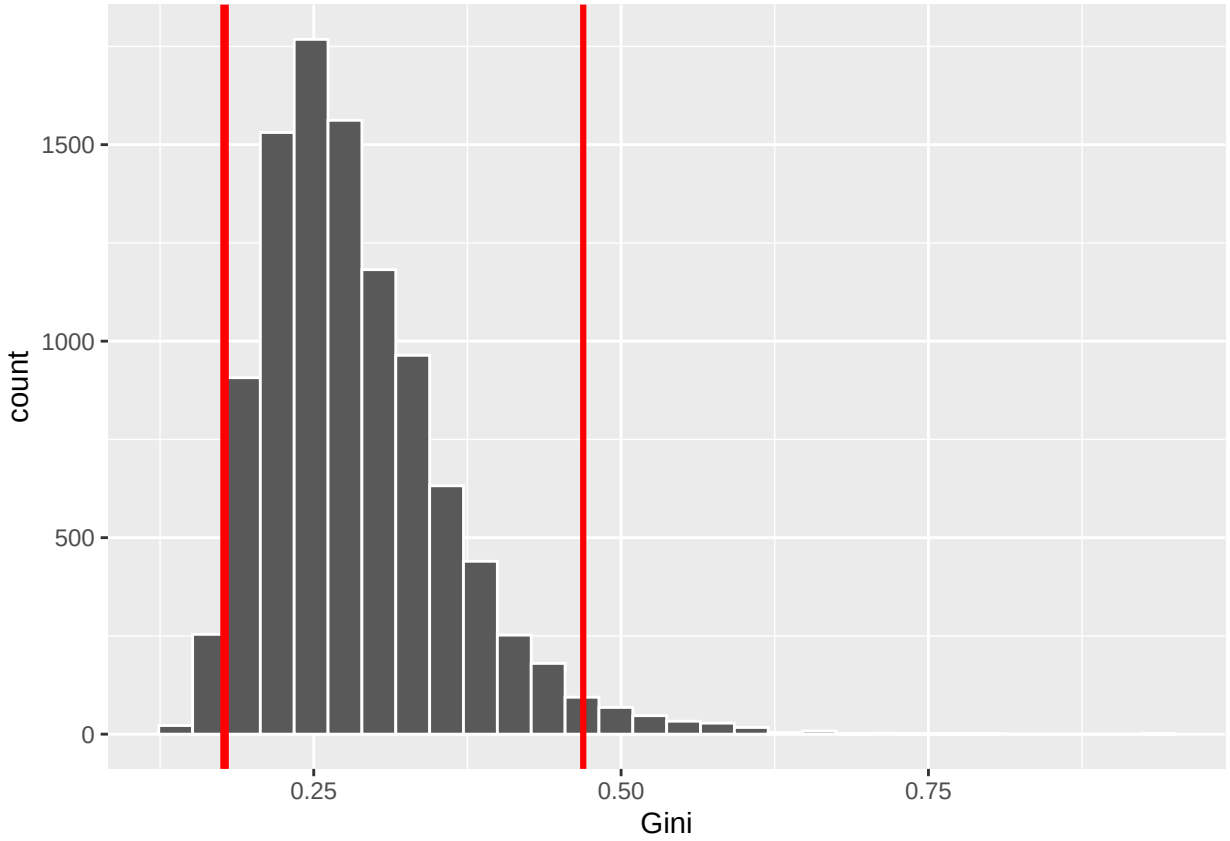
```
#b

gini <- 2*pnorm(sqrt(posterior/2),0,1)-1
hist(gini,breaks=30)
```



2c

```
#c

interval <- quantile(gini,c(0.025,0.975))
df <- data.frame("y" = gini)
ggplot(df) +
  geom_histogram(aes(y),color = "white") +
  geom_vline(xintercept = interval[["2.5%"]],color = "red",size=1.5) +
  geom_vline(xintercept = interval[["97.5%"]],color = "red",size=1.1) +
  xlab("Gini")
```

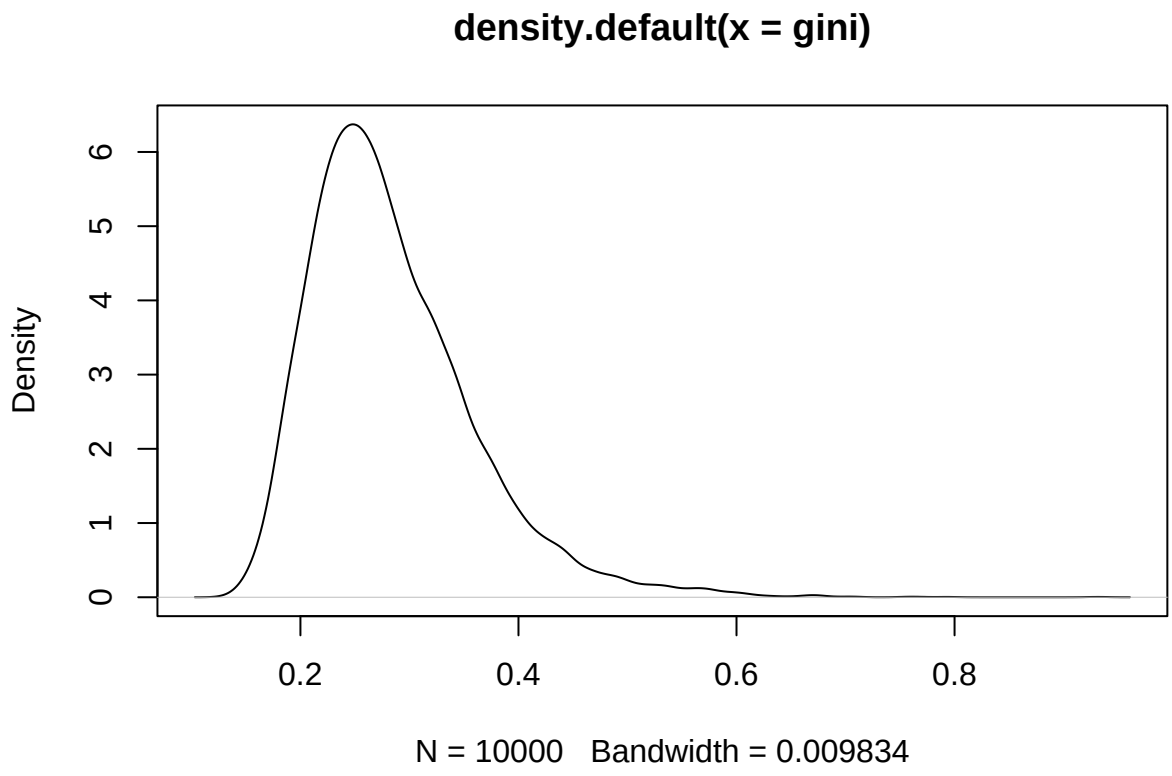
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The 95% equal tail credible interval is represented with the red lines in the plot above.

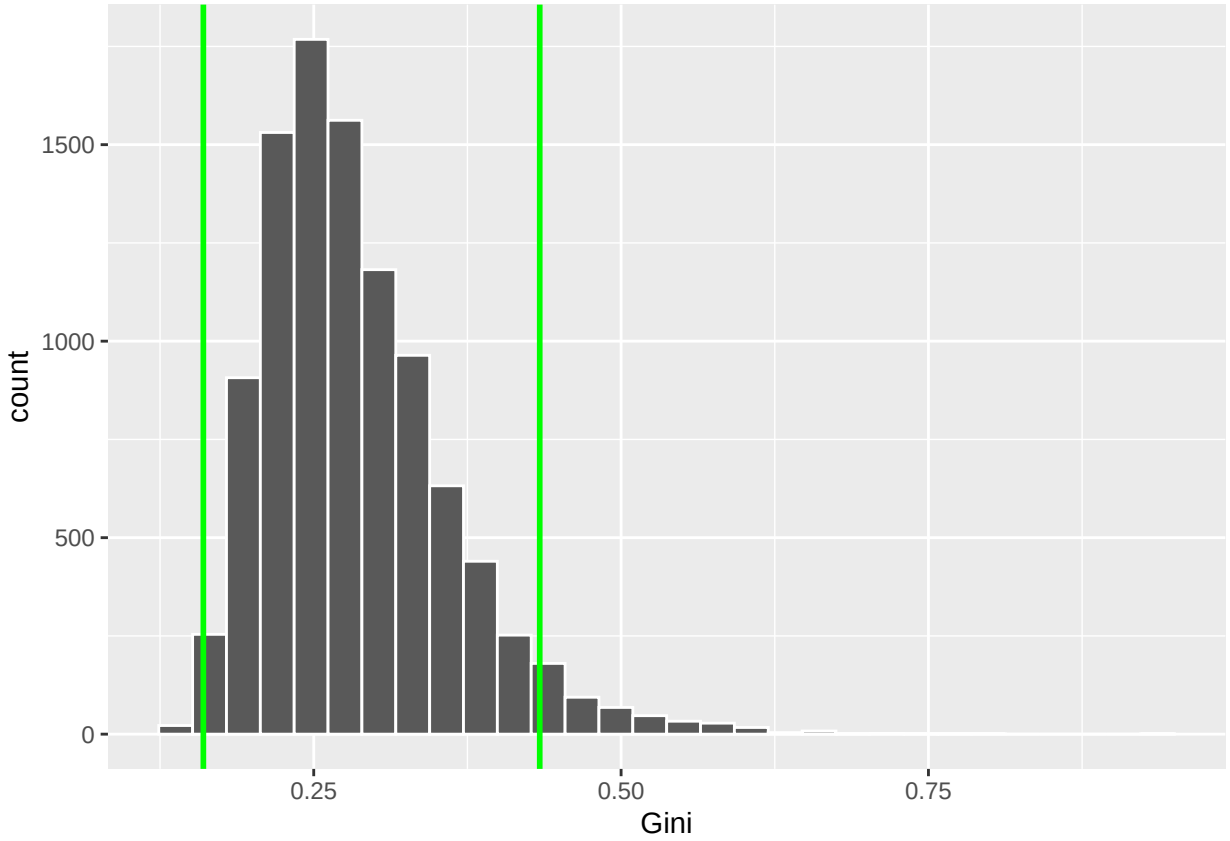
2d

```
#d
gini_density <- density(gini)
plot(gini_density)
```




```
gini_density <- data.frame("x" = gini_density$x, "y" = gini_density$y)
gini_density_sorted <- gini_density[order(gini_density$y, decreasing = T), ]
gini_density_sorted[["cumsum"]] <- cumsum(gini_density_sorted$y)

threshold <- 0.95 * sum(gini_density_sorted$y)
HDPI <- range(gini_density_sorted[gini_density_sorted[["cumsum"]] <= threshold, "x"])


ggplot(df) +
  geom_histogram(aes(y),color = "white") +
  geom_vline(xintercept = HDPI[1],color = "green",size=1) +
  geom_vline(xintercept = HDPI[2],color = "green",size=1) +
  xlab("Gini")
```

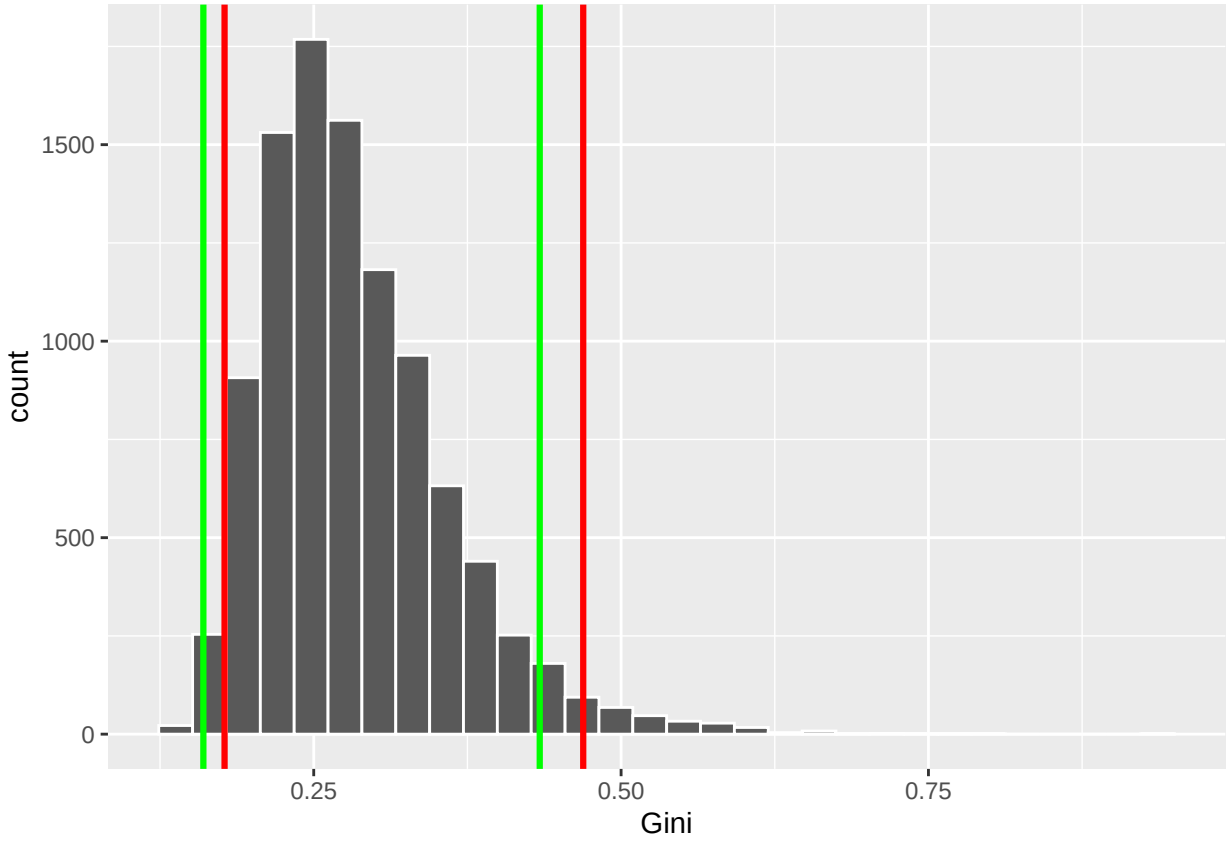
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The 95% HDPI is represented with the green lines in the plot above.

```
ggplot(df) +
  geom_histogram(aes(y),color = "white") +
  geom_vline(xintercept = interval[["2.5%"]],color = "red",size=1.1) +
  geom_vline(xintercept = interval[["97.5%"]],color = "red",size=1.1) +
  geom_vline(xintercept = HDPI[1],color = "green",size=1.1) +
  geom_vline(xintercept = HDPI[2],color = "green",size=1.1) +
  xlab("Gini")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The 95% equal tail credible interval is represented with the red lines and the 95% HDPI is represented with the green lines in the plot above. As we can see, the HDPI is more centered compared to CI. Therefore HDPI is a better choice for skewed distributions.

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

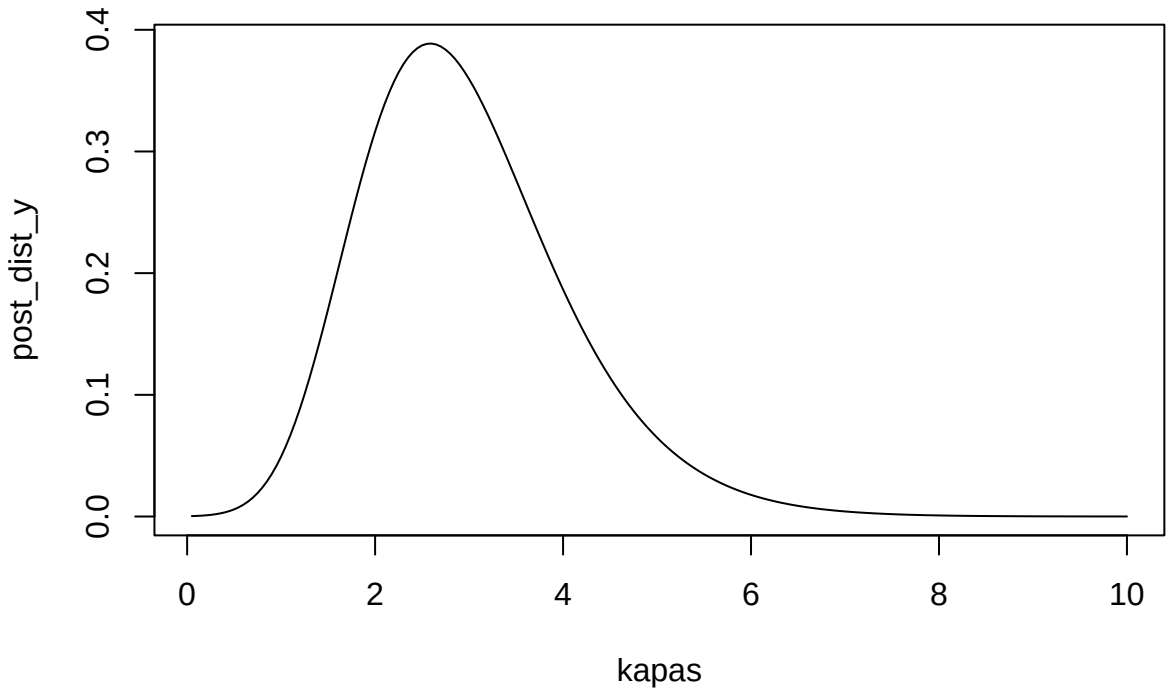
3a

The posterior is proportional to the following expression:

$$p(k|y,\mu) \propto \frac{\exp(k\sum_i^n cos(y_i-\mu)-0.5)}{I_0(k)^n}$$

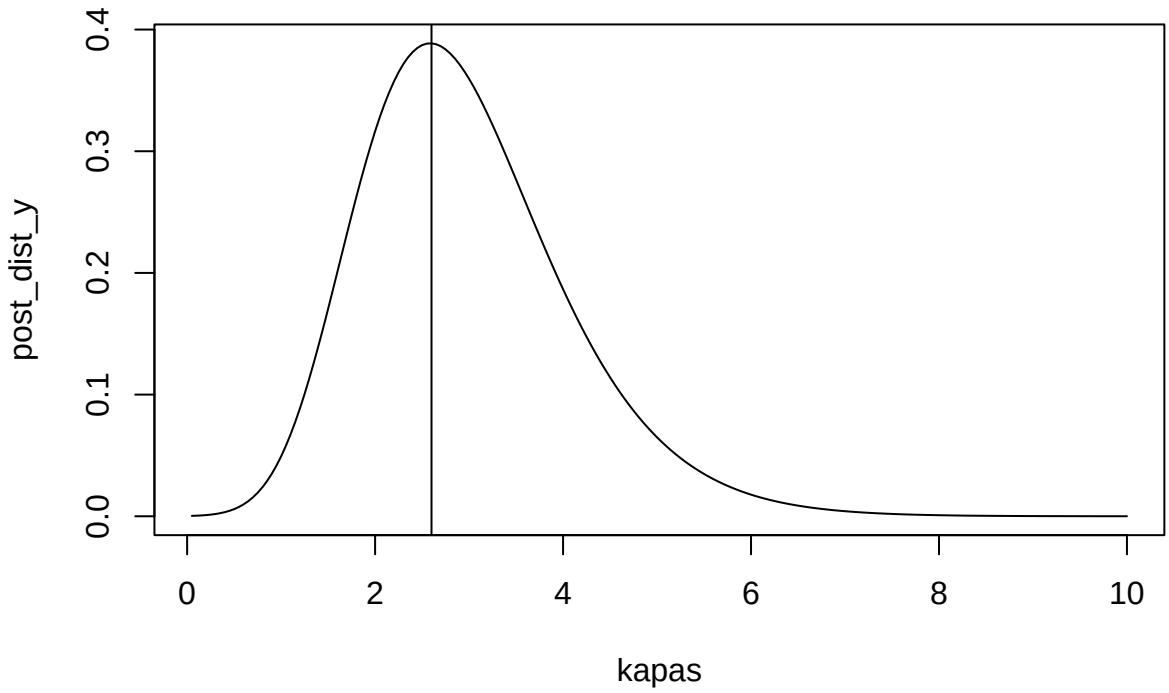
```
wind_dir <- c(20, 314, 285, 40, 308, 314, 299, 296, 303, 326)
wind_dir_rad <- c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)


lambda <- 0.5
mu <- 2.4
n <- length(wind_dir_rad)
# 3.a

posterior <- function(kapa, y = wind_dir_rad){
exp(kapa * (sum(cos(y - mu)) - lambda))/ (besselI(kapa, 0))^n
}
```

```r
kapas <- seq(0.05, 10, by = 0.05)

post_dist_y <- posterior(kapas)

post_dist_y <- post_dist_y/integrate(posterior, 0,10 )$value

plot(kapas, post_dist_y, type = "l")
```



3b

```r
# 3.b
mode_post <- kapas[which.max(post_dist_y)]
plot(kapas, post_dist_y, type = "l")
abline(v = mode_post)
```



As we can see, even if the models are weird, the posterior distribution can be plotted using a grid of values for the κ. Also, in this way we can approximate the mode of the posterior.
Also we used `integrate()` function to make the density integrate to one.