Baysian Learning
732A91

Lab 1

Guilherme Barros
guiba484

Eric Herwin
erihe068

1. Bernoulli... Again

For this task we have the data $y_1, ..., y_n|\theta \sim Bern(\theta)$. For this we have obtained a sample where we have 14 successes for 20 trials, i.e. $s = 14,\ n = 20$. We will also assume a $Beta(\alpha_0, \beta_0)$ prior for θ. We will also let $\alpha_0 = \beta_0 = 2$.

a)

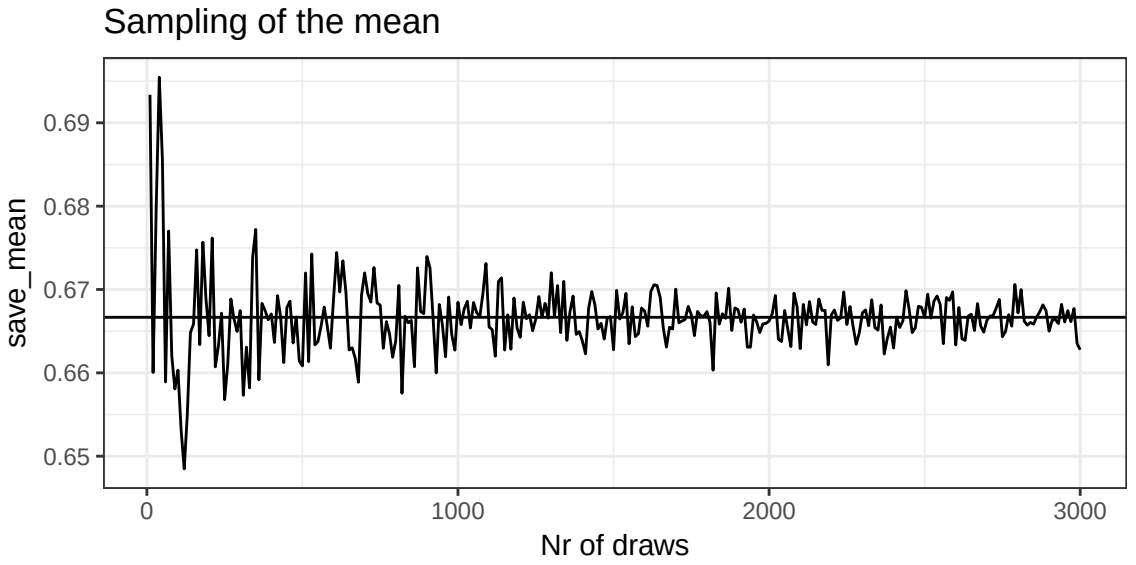
Here we will draw random rumbers from the posterior $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$ where $f = n - s$ and we will show graphically that the posterior mean and standard deviation converges to the true values. The true mean and standard devation of a Beta in out case is:

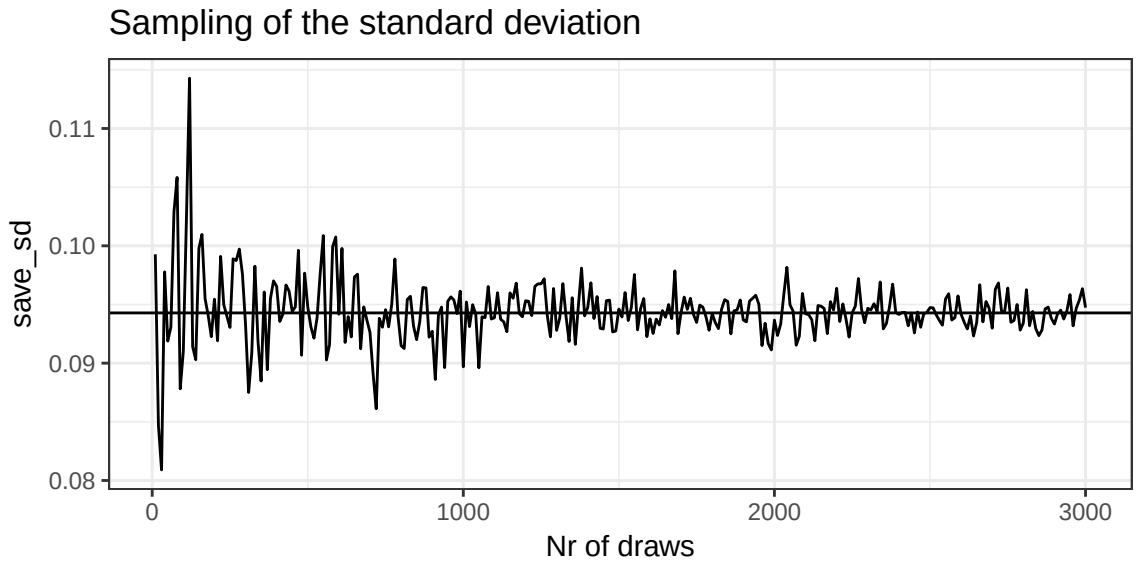
$$mean = \frac{\alpha_{post}}{\alpha_{post} + \beta_{post}}, \quad sd = \sqrt{\frac{\alpha_{post}\beta_{post}}{(\alpha_{post} + \beta_{post})^2(\alpha_{post} + \beta_{post} + 1)}}$$

where $\alpha_{post} = \alpha_0 + s$ and $\beta_{post} = \beta_0 + f$.

Lets do some calculations:

```
s <- 14
n <- 20

f <- n-s

post_alpha <- 2 + s
post_beta <- 2 + f

tr_mean <- post_alpha / (post_alpha + post_beta) #true mean

vari_post <- post_alpha*post_beta / ((post_alpha + post_beta)^2 * (post_alpha + post_beta +1))
vari_post <- sqrt(vari_post) #true standard deviation

save_mean <- c()
save_sd <- c()
by_loop <- seq(10,3000, 10) # loop the number of draws
for(i in by_loop){
  dr <- rbeta(n = i, post_alpha, post_beta)
  save_mean <- c(save_mean, mean(dr)) #simmulated mean
  save_sd <- c(save_sd, sd(dr)) #simmulated standard deviation
}
```





From the plots we can see that with the increasing amount of draws, the mean and the standard deviation seem to be converging to the true value.

b)

Here we will use a summulation with 10000 draws to compute the posterior probability $Pr(\theta < 0.4|y)$ and compare it with the exact value.

```
nDraws <- 10000

beta_draw <- rbeta(n = nDraws, shape1 = post_alpha, shape2 = post_beta)
paste0("Simulated Value: ",sum(beta_draw<0.4)/length(beta_draw))
```

```
## [1] "Simulated Value: 0.0044"
```

```
paste0("Exact value: ", (pbeta(0.4,post_alpha, post_beta)))
```

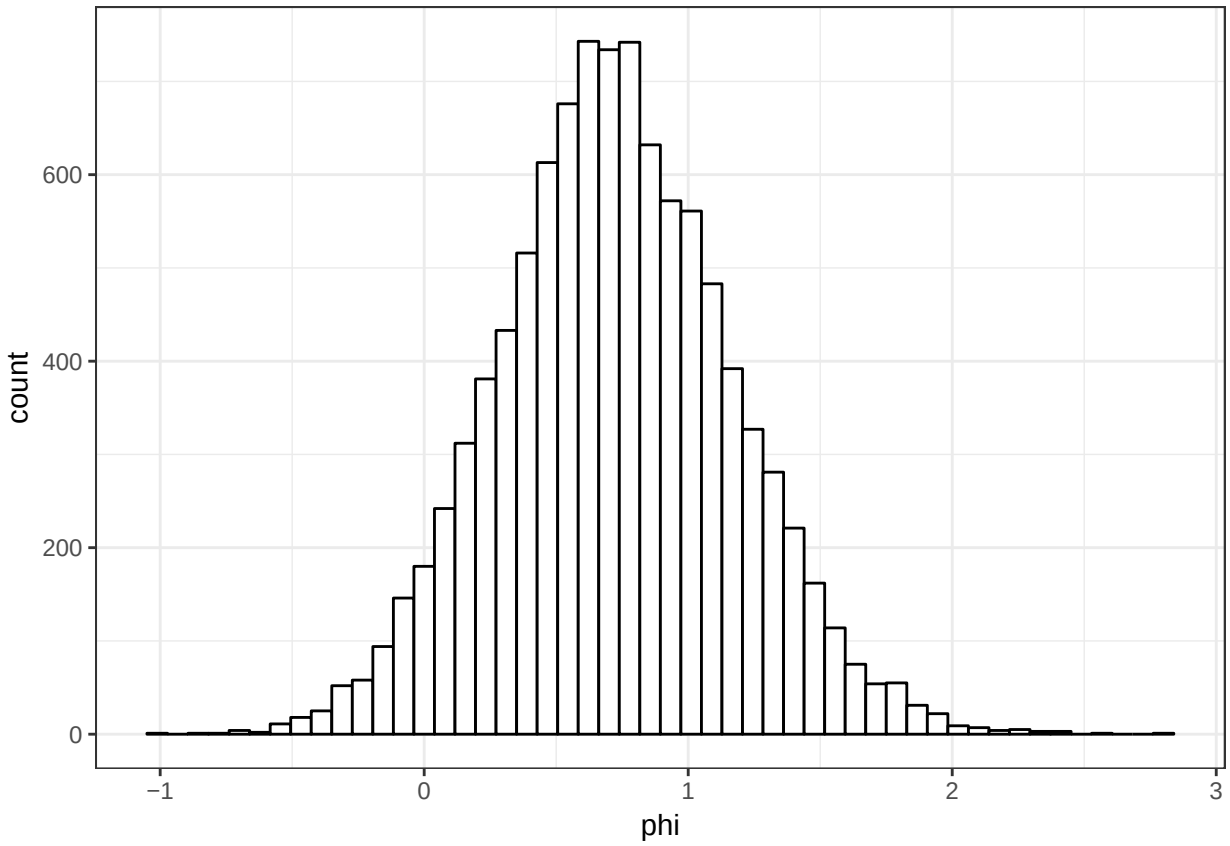
```
## [1] "Exact value: 0.00397268082810898"
```

We can see that the true density value and a sample of beta values are almost the same.

c)

Here we will compute posterior distribution of the log-odds $\phi = log(\theta/(1-\theta))$ with 10000 draws. We will use the information from the last task.

```
phi <- log(beta_draw/(1-beta_draw)) #log odds

ggplot() +
  geom_histogram(aes(phi), bins = 50, color = "black", fill = "white") +
  theme_bw()
```



2. Log-normal distribution and the Gini coefficient.

For this task we have the data (y): 14, 25, 45, 25, 30, 33, 19, 50, 34 and 67. This is a $y_1, ..., y_n|\mu, \sigma^2 \sim log\mathcal{N}(\mu, \sigma^2)$, where $\mu = 3.5$ but our σ^2 is unknown. We have a non-informative prior $p(\sigma^2) \propto 1/\sigma^2$. The posterior of σ^2 is the $Inv - \chi^2(n, \tau^2)$ distibution, where

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

a)

For this task we want to simmulate 10000 draws from the posterior of σ^2 and compare it with the theoretical $Inv - \chi^2(n, \tau^2)$. The theoretical mean of σ^2 is formulated as:

$$mean = \frac{n\tau^2}{n-2}$$

We will do it with the code:

```
y <- c(14, 25, 45, 25, 30, 33, 19, 50, 34, 67) #data

mu <- 3.5
tau_sq <- sum((log(y) - mu)^2)/length(y)

n <- length(y)
X <- rchisq(10000, n)
inv_chi <- n * tau_sq/X #posterior simga^2
```

The sample mean:

```
mean(inv_chi) #sigma mean
```

```
## [1] 0.2485979
```

The true mean:

```
n*tau_sq / (n - 2) #true mean
```

```
## [1] 0.2473497
```

We can see that the posterior estimation of σ^2 from y is very close to the theoretical value. To visiualise this we need to sample from the true $Inv - \chi^2(n, \tau^2)$ distribution. To do that we need to do the following:

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

Then scaled distribution will look like:

$$Y \sim Scaled\ Inv\ \chi^2(n, \tau^2)$$

We have that:

$$\frac{Y}{\tau^2 n} \sim Inv\ \chi^2(n) \rightarrow \frac{Y}{\tau^2 n} = \frac{1}{X}$$

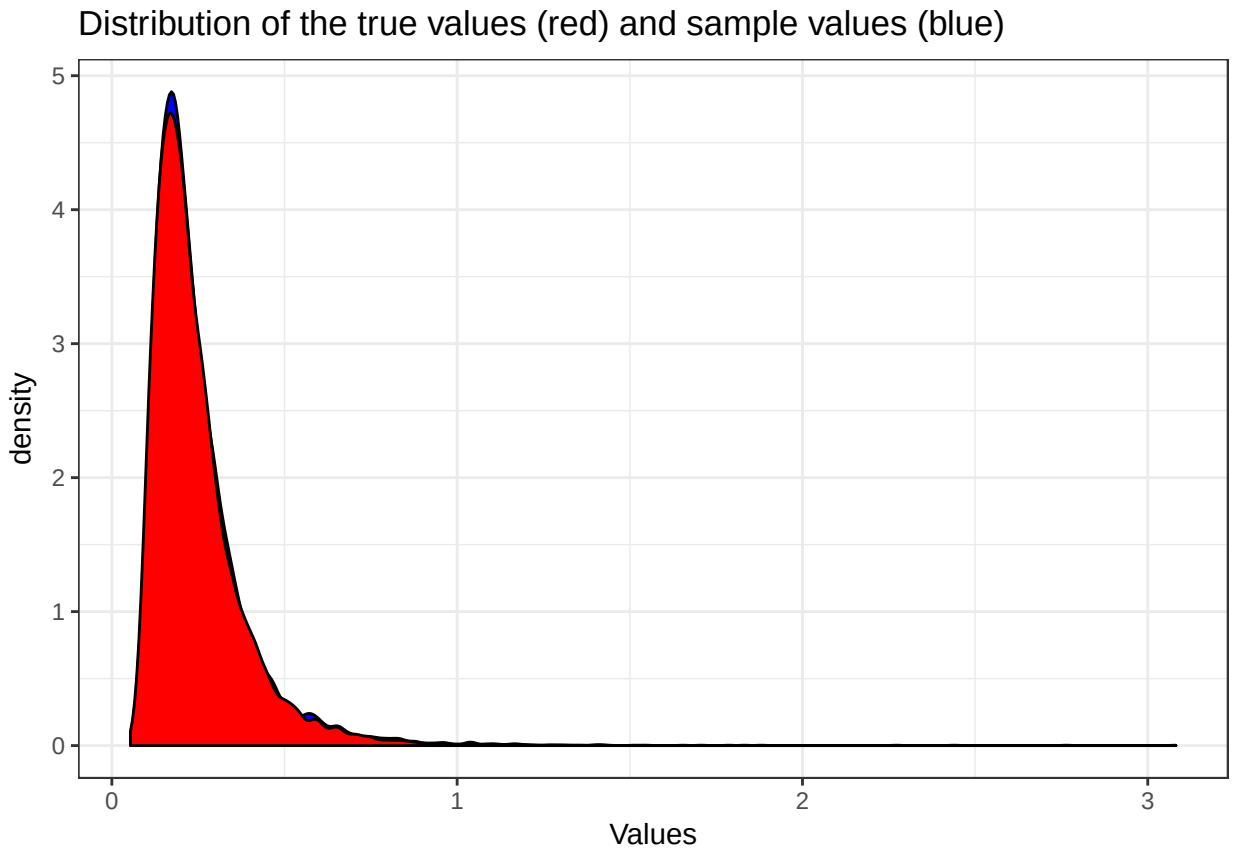
This will result in:

$$Y = \frac{\tau^2 n}{X}$$

Applying this will result in:

```
true <- (tau_sq*n)/rchisq(10000, n)
ggplot() +
  geom_density(aes(sort(inv_chi)), color = "black", fill = "blue") +
  geom_density(aes(sort(true)), fill = "red") +
```

```
  labs(title = "Distribution of the true values (red) and sample values (blue)", x = "Values")+
  theme_bw()
```



We can see that the sample and the true distibution are almost the same for out given μ.

b)

For this task we will compute the Gini-coefficient, G, where G is calculated as follows:

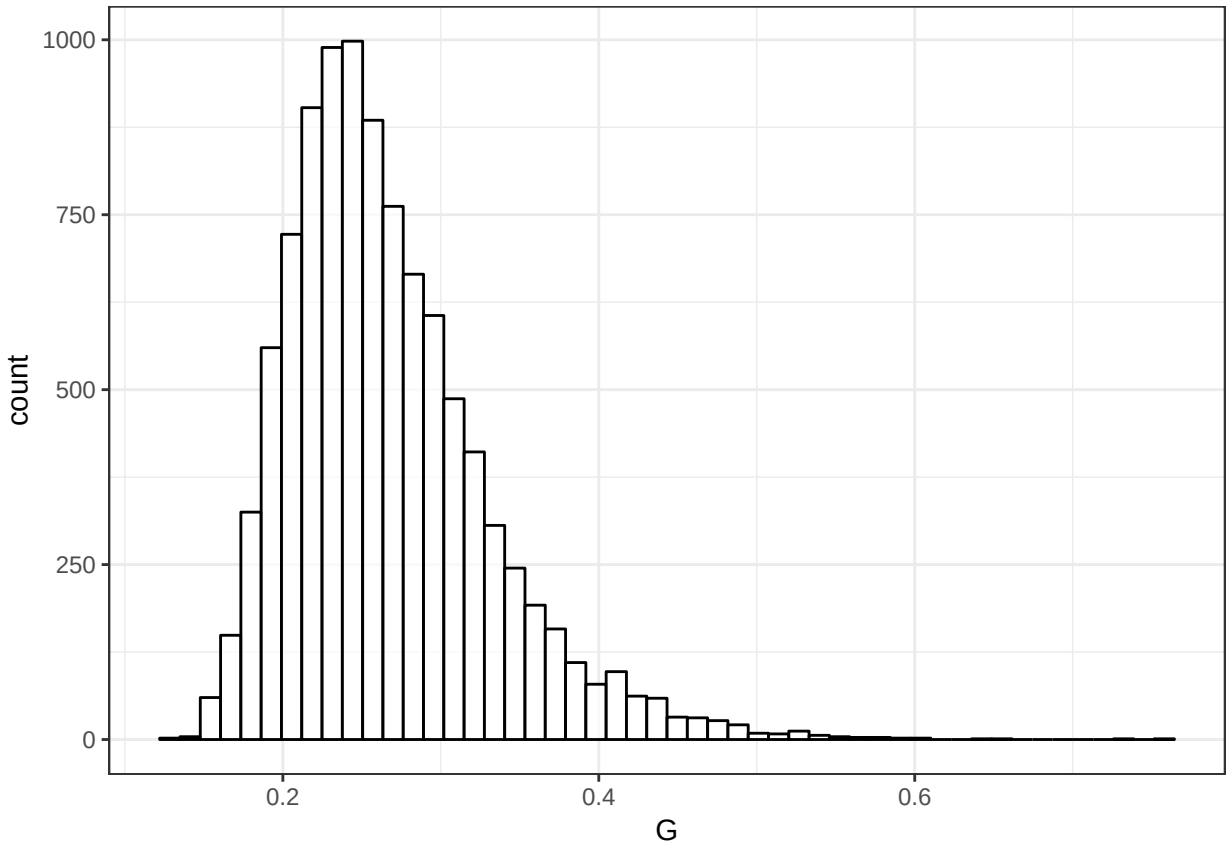
$$G = 2\Phi(\sigma/\sqrt{2}) - 1$$

where the $\Phi(z)$ is the CDF for the standard normal distribution: $N(0, 1)$. The CDF will be drawn with the function `pnorm(q = sqrt(inv_chi/2), mean = 0, sd = 1)`. we will use σ^2 draws (`inv_chi`) from the previous task as:

```
G <- 2*pnorm(q = sqrt(inv_chi/2), mean = 0, sd = 1) -1 #gini coef
```

The Gini-coefficient, G, follows a *log* $\mathcal{N}(\mu, \sigma^2)$ distribution.

The distribution for G is:

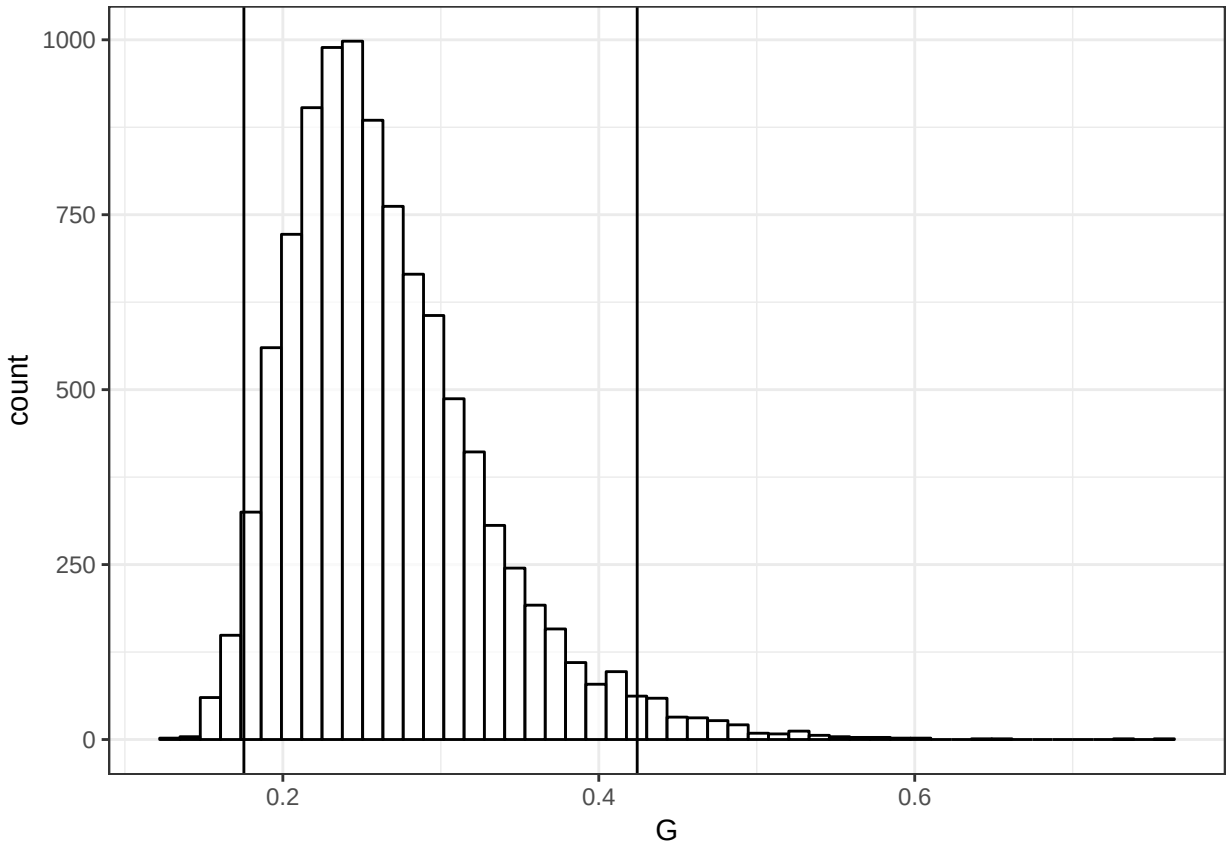


We can see that from the distribution that the most dense region is around ≈ 0.25, with low probability that the G is higher than 0.4. This implies that it is very likely that the income distribution in this case is reasonably equal, especially when compared to the world average (which is estimated to be close to 0.4).

c)

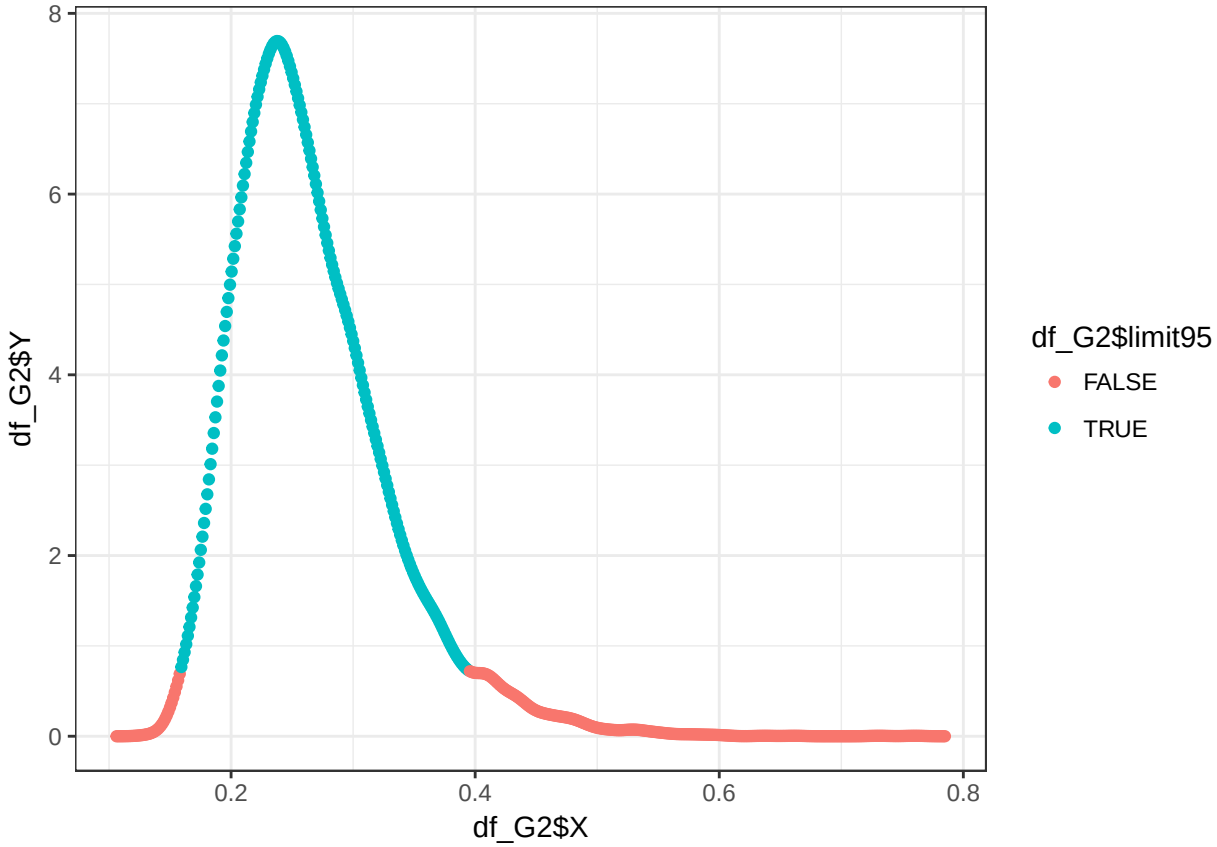
For this task we will use the posterior draws from b) to compute a 95% equal tail credible interval for G.

```
interval <- quantile(G, probs = c(0.025, 0.975)) #equal tail credible interval


ggplot() +
  geom_histogram(aes(G), bins = 50, fill = "white", color = "black", alpha = 0.7) +
  geom_vline(xintercept = interval[1]) +
  geom_vline(xintercept = interval[2]) +
  theme_bw()
```



Here we will compute a 95% Highest Posterior Density interval for G.

```
G_dens = density(G)

maximum_local = which.max(G_dens$y)
maximum_y = G_dens$y[maximum_local]
maximum_x = G_dens$x[maximum_local]

G_dens_sort = sort(G_dens$y, decreasing=TRUE)
G_dens_sum = cumsum(G_dens_sort)
G_dens_sum = G_dens_sum/G_dens_sum[length(G_dens_sum)]
G_dens95 = max(which(G_dens_sum<0.95))
y_cut = G_dens_sort[G_dens95]

df_G2 = data.frame(Y = G_dens$y,X = G_dens$x)
df_G2$limit95 = df_G2$Y>y_cut

qplot(df_G2$X,df_G2$Y,col=df_G2$limit95, geom="point")+ theme_bw()
```



Comparing the two plots, we can see that the HPD interval which is expected. It is worth noting that in this case, the HDP is not that different from the quantile interval, but bigger differences may arise in other types of distributions, especially in multimodal cases (in these, the HDP can yield non-interval set estimators). Any point in the HDP has a higher density than any point outside it, so it is the collection of most likely values of the parameter.

3 Wind direction.

Here we have a dataset of 10 observations that is expressed from degrees, into radians y, where $-\pi < y < \pi$. The data-points are following the von Mises distribution

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

where $I_0(\kappa)$ is the Bessel function for order zero (used with `besselI()`).

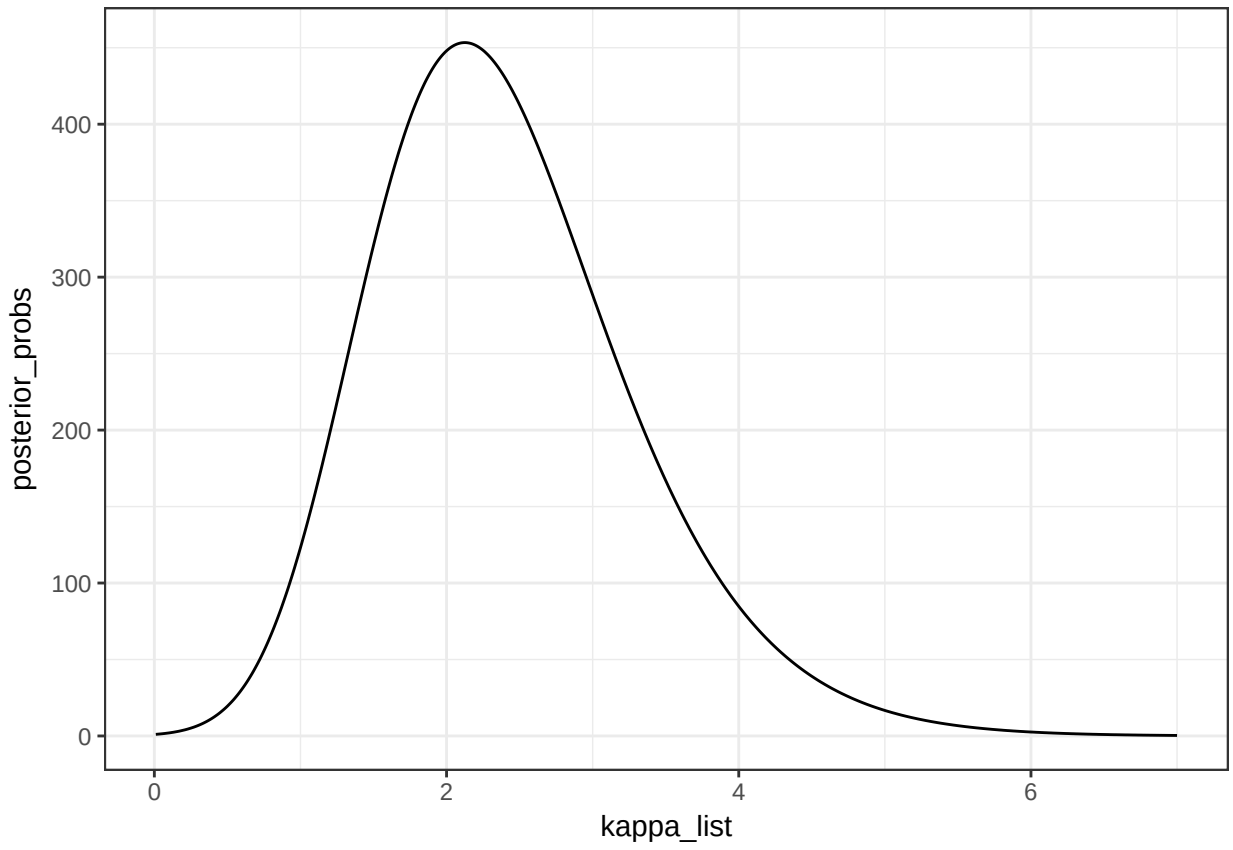
We will assume in this task that $\mu = 2.39$

a)

Here we plot the posterior distribution for different κ values for the wind direction data.

```
kappa_list = seq(from=0.01, to=7, by=0.01)

y = c(-2.44,2.14,2.54,1.83,2.02,2.33,-2.79,2.23,2.07,2.02)
```

```r
mu = 2.39


posterior = function(y, kappa, mu){
  n = length(y)
  b = besselI(kappa, 0)
  p_y = exp(kappa*(sum(cos(y - mu)) - 1))/((b)^n)
  return((p_y))
}


n = length(kappa_list)
posterior_probs = c()
i = 1

for (kappa in kappa_list){
  posterior_probs[i] = posterior(y, kappa, mu)
  i <- i + 1
}

ggplot() +
  geom_line(aes(x = kappa_list, y = posterior_probs)) +
  theme_bw()
```



b)

Here we will find the (approximate) posterior mode of κ from the results in a).

```
kappa_list[which.max(posterior_probs)] #posterior mode
```

```
## [1] 2.12
```

And we can see that the posterior mode corresponds to the highest point in the distribution plotted above.