# Lab1

2024-04-05

## Bayesian Learning

## Computer Lab 1

```r
library(ggplot2)
```

### 1. Daniel Bernoulli

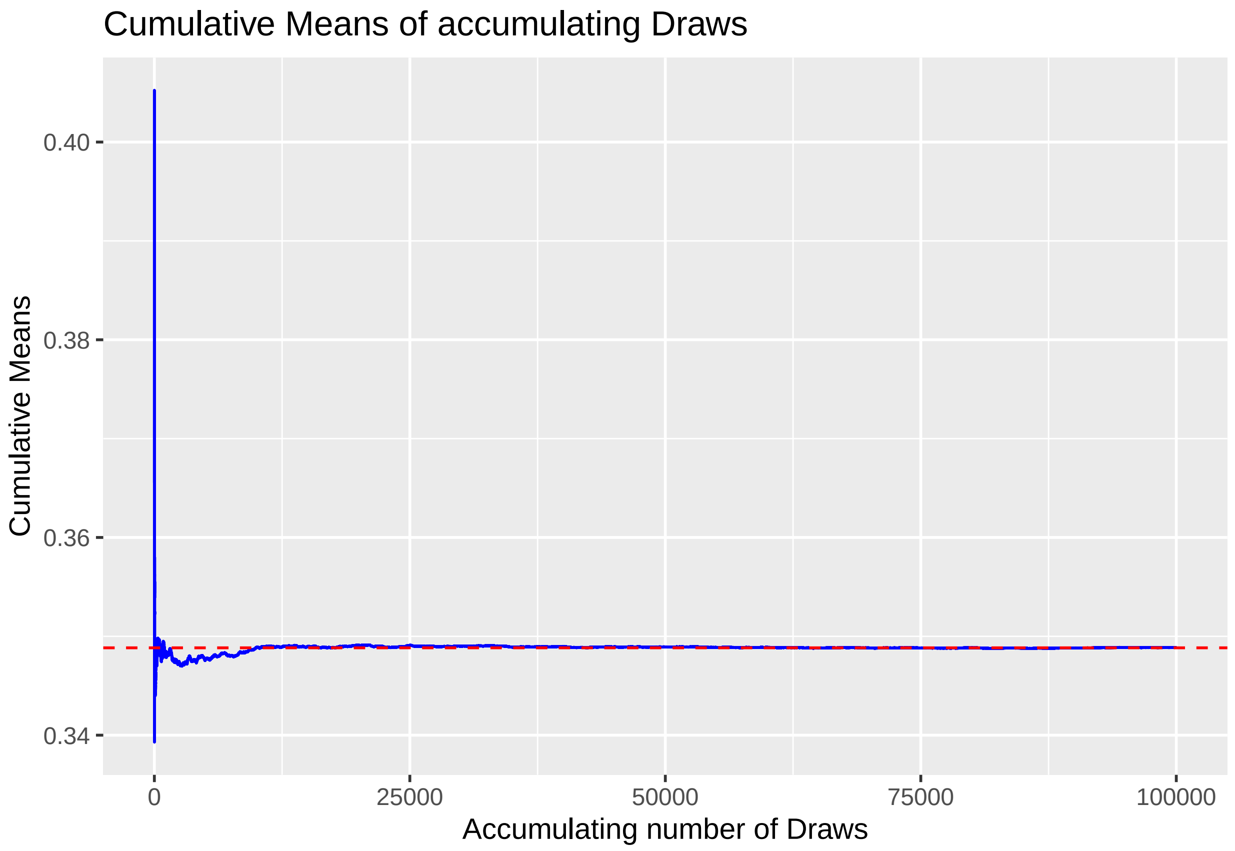
Let $y_1, ..., y_n|\theta \sim Bern(\theta)$, and assume that you have obtained a sample with s = 22 successes in n = 70 trials. Assume a Beta($\alpha_0$, $\beta_0$) prior for $\theta$ and let $\alpha_0 = \beta_0 = 8$.

#### 1.a

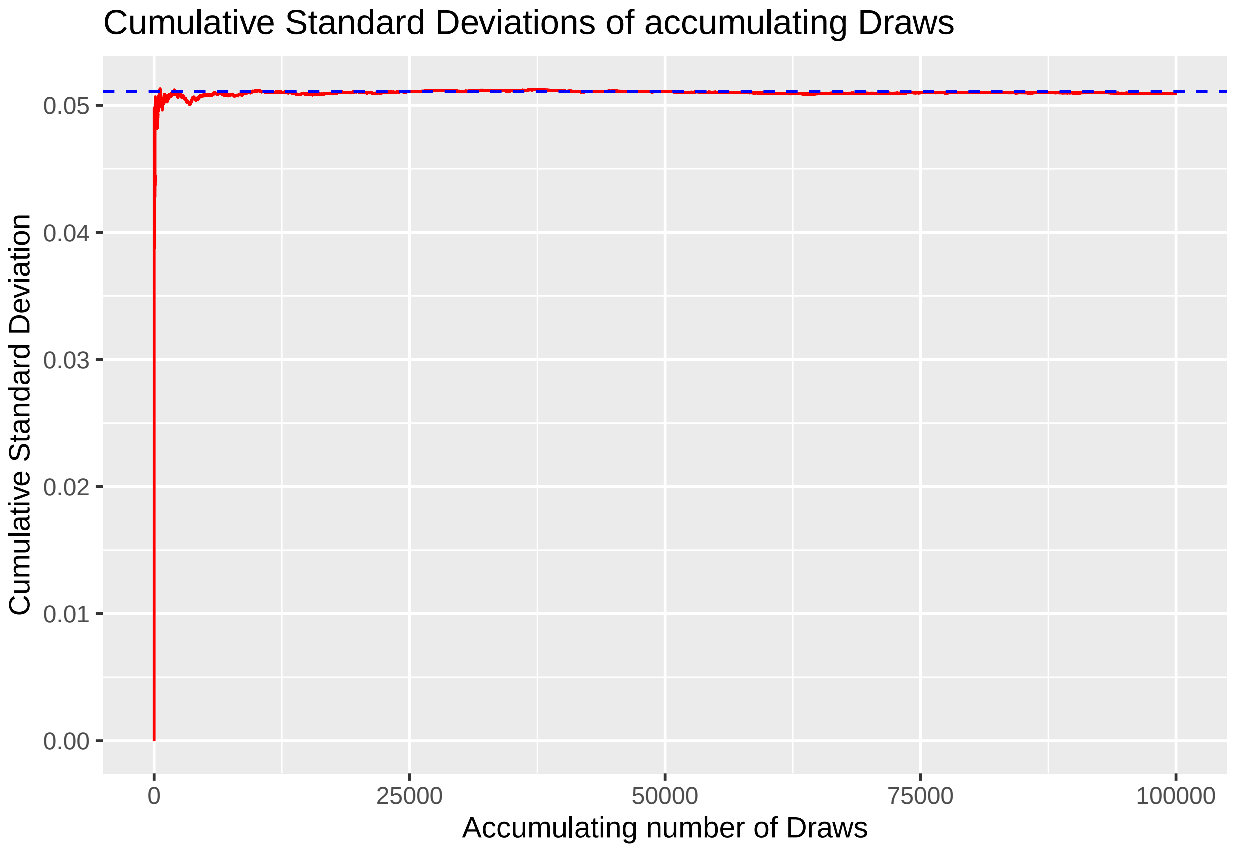
Draw 10000 random values (nDraws = 10000) from the posterior $\theta|y \sim Beta(\alpha_0 + s, \beta_0 + f)$, where $y = (y_1, ..., y_n)$, and verify graphically that the posterior mean $E[\theta|y]$ and standard deviation $SD[\theta|y]$ converges to the true values as the number of random draws grows large. [Hint: use rbeta() to draw random values and make graphs of the sample means and standard deviations of $\theta$ as a function of the accumulating number of drawn values].

```r
s <- 22
n <- 70
f <- n-s
nDraws <- 100000
alpha_0 <- 8
beta_0 <- 8

post_alpha <- alpha_0 + s
post_beta <- beta_0 + f

# rbeta drawing
delta <- rbeta(nDraws, post_alpha, post_beta)

# true values of mean and sd
mean_true <- post_alpha/(post_alpha + post_beta)
sd_true <- sqrt((post_alpha*post_beta)/((post_alpha+post_beta+1)*(post_alpha+post_beta)^2))

# sample means based on accumulating number of drawn values
cumu_mean <- cumsum(delta)/(1:nDraws)
cumu_sd <- sqrt(cumsum((delta - cumu_mean)^2) / (1:nDraws))
```

```r
plot_data <- data.frame(
  Draw = 1:nDraws,
  Mean = cumu_mean,
  SD = cumu_sd
)

# Mean plot
cumulativeMeans_plot <- ggplot(plot_data, aes(x = Draw)) +
  geom_line(aes(y = Mean), color = "blue") +
  # true mean line
  geom_hline(yintercept = mean_true, color = "red", linetype = "dashed") +
  ggtitle("Cumulative Means of accumulating Draws") +
  xlab("Accumulating number of Draws") +
  ylab("Cumulative Means")

# SD plot
cumulativeSD_plot <- ggplot(plot_data, aes(x = Draw)) +
  geom_line(aes(y = SD), color = "red") +
  # true standard deviation line
  geom_hline(yintercept = sd_true, color = "blue", linetype = "dashed") +
  ggtitle("Cumulative Standard Deviations of accumulating Draws") +
  xlab("Accumulating number of Draws") +
  ylab("Cumulative Standard Deviation")

cumulativeMeans_plot
```



```
cumulativeSD_plot
```



### 1.b

Draw 10000 random values from the posterior to compute the posterior probability $Pr(\theta > 0.3|y)$ and compare with the exact value from the Beta posterior. [Hint: use pbeta()]

```
# posterior probability:
p_post <- mean(delta > 0.3)

# beta posterior probability:
p_beta_post <- 1-(pbeta(0.3, post_alpha, post_beta))

p_post
```

```
## [1] 0.83041
```

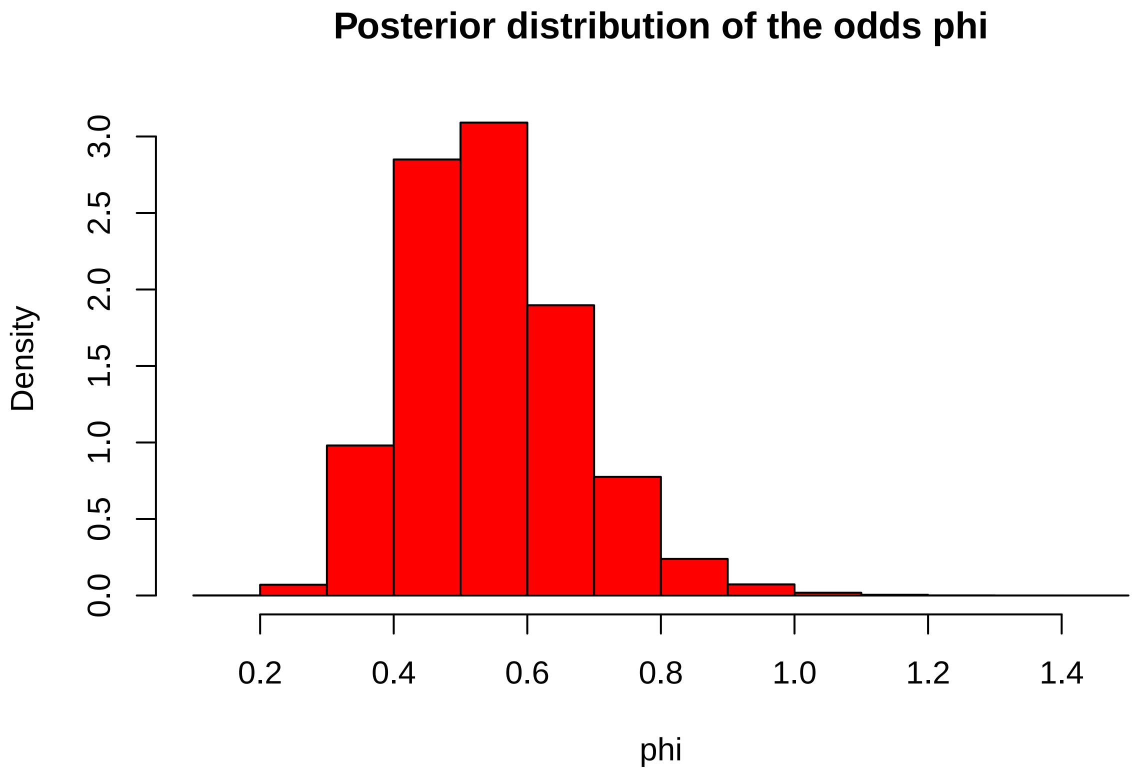
```
p_beta_post
```

```
## [1] 0.8285936
```

### 1.c

Draw 10000 random values from the posterior of the odds $\phi = \frac{\theta}{1-\theta}$ by using the previous random draws from the Beta posterior for $\theta$ and plot the posterior distribution of $\phi$. [Hint: hist() and density() can be utilized].

```
phi <- delta/(1-delta)

postDistribution_plot <- hist(phi, breaks=10, col="red", prob=TRUE, xlab="phi",
   main="Posterior distribution of the odds phi")
```



```
postDistribution_plot
```

```
## $breaks
##  [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5
##
## $counts
##  [1]     7   703  9805 28498 30908 18972  7753  2391   727   180    46     7
## [13]     2     1
##
## $density
##  [1] 0.0007 0.0703 0.9805 2.8498 3.0908 1.8972 0.7753 0.2391 0.0727 0.0180
## [11] 0.0046 0.0007 0.0002 0.0001
##
## $mids
##  [1] 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95 1.05 1.15 1.25 1.35 1.45
```

```
##
## $xname
## [1] "phi"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

## 2. Log-normal distribution and Gini coefficient

Assume that you have asked 8 randomly selected persons about their monthly income (in thousands Swedish Krona) and obtained the following eight observations: 33, 24, 48, 32, 55, 74, 23, and 17. A common model for non-negative continuous variables is the log-normal distribution. The log-normal distribution $logN(\mu, \sigma^2)$ has density function

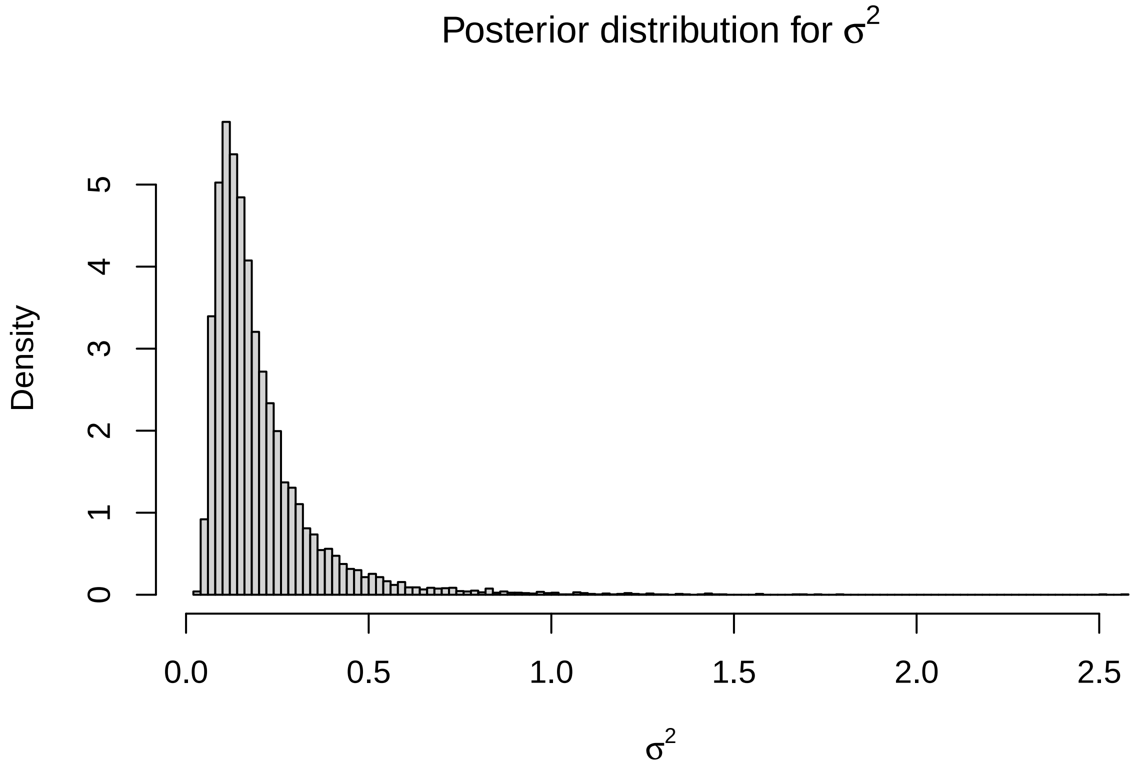
$$p(y|\mu, \sigma^2) = \frac{1}{y * \sqrt{2\pi\sigma^2}} exp[-\frac{1}{2\sigma^2}(\log y - \mu)^2]$$

, where y $> 0$, $-\infty < \mu < \infty$ and $\sigma^2 > 0$. The log-normal distribution is related to the normal distribution as follows: if $y \sim logN(\mu, \sigma^2)$ then $logy \sim N(\mu, \sigma^2)$. Let $y_1, ..., y_n|\mu,\ \sigma^2 iid \sim logN(\mu, \sigma^2)$, where $\mu = 3.6$ is assumed to be known but $\sigma^2$ is unknown with non-informative prior $p(\sigma^2) \propto 1/\sigma^2$ . The posterior for $\sigma^2$ is the $Inv - \chi^2(n, \tau^2)$ distribution, where $\tau^2 = \frac{\sum_{i=1}^{n}(\log y_i - \mu)^2}{n}$ .

### 2.a

Draw 10000 random values from the posterior of $\sigma^2$ by assuming $\mu = 3.6$ and plot the posterior distribution.

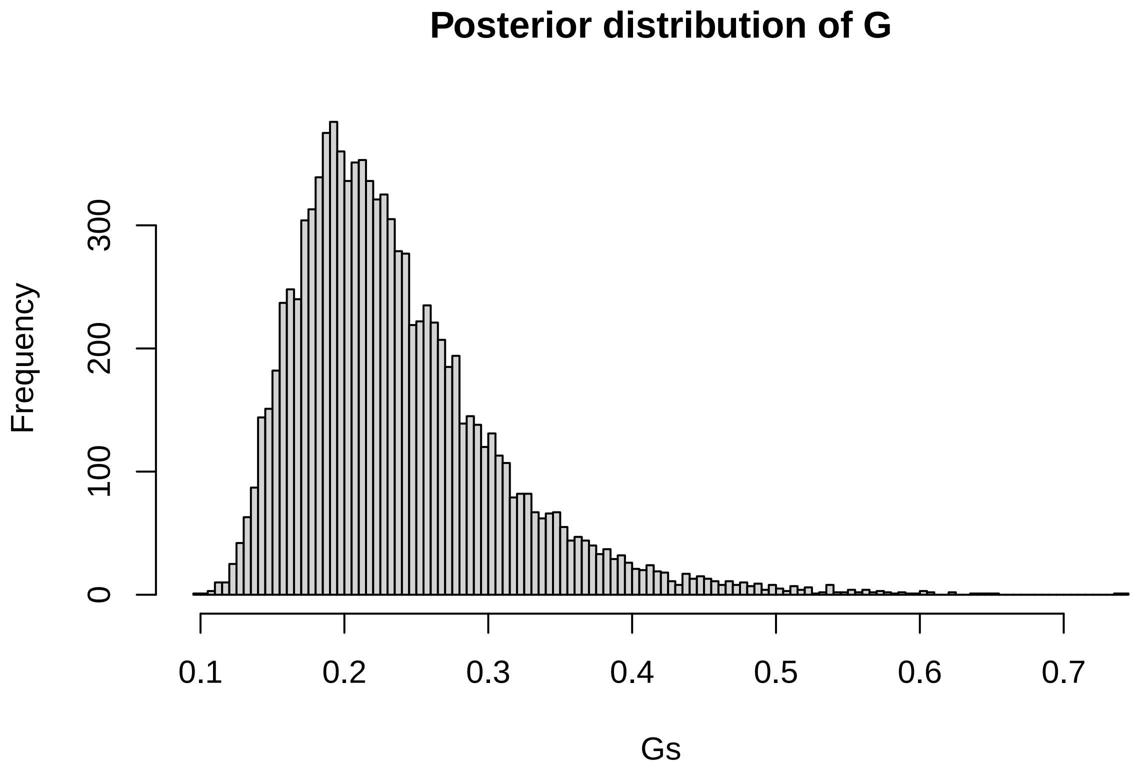
```
Y = c(33, 24, 48, 32, 55, 74, 23, 17)
n = length(Y)
df = n-1

mu = 3.6
tau_sq = (sum((log(Y)-mu)^2))/n

sigsq_draws = 1 / rchisq(10000, df)

hist(sigsq_draws, prob=TRUE, xlab = expression(sigma^2), main=expression(paste("Posterior distribution
```



### 2.b

The most common measure of income inequality is the Gini coefficient, G, where $0 \leq G \leq 1$. G $=$ 0 means a completely equal income distribution, whereas G $=$ 1 means complete income inequality (see e.g. Wikipedia for more information about the Gini coefficient). It can be shown that $G = 2\Phi(\sigma/\sqrt{2}) - 1$ when incomes follow a $logN(\mu, \sigma^2)$ distribution. $\Phi(z)$ is the cumulative distribution function function (CDF) for the standard normal distribution with mean zero and unit variance. Use the posterior draws in a) to compute the posterior distributiuon of the Gini coefficient G for the current data set.

```
Gs = 2*pnorm((sqrt(sigsq_draws)/sqrt(2))) - 1

hist(Gs, main = "Posterior distribution of G", breaks = 100 )
```



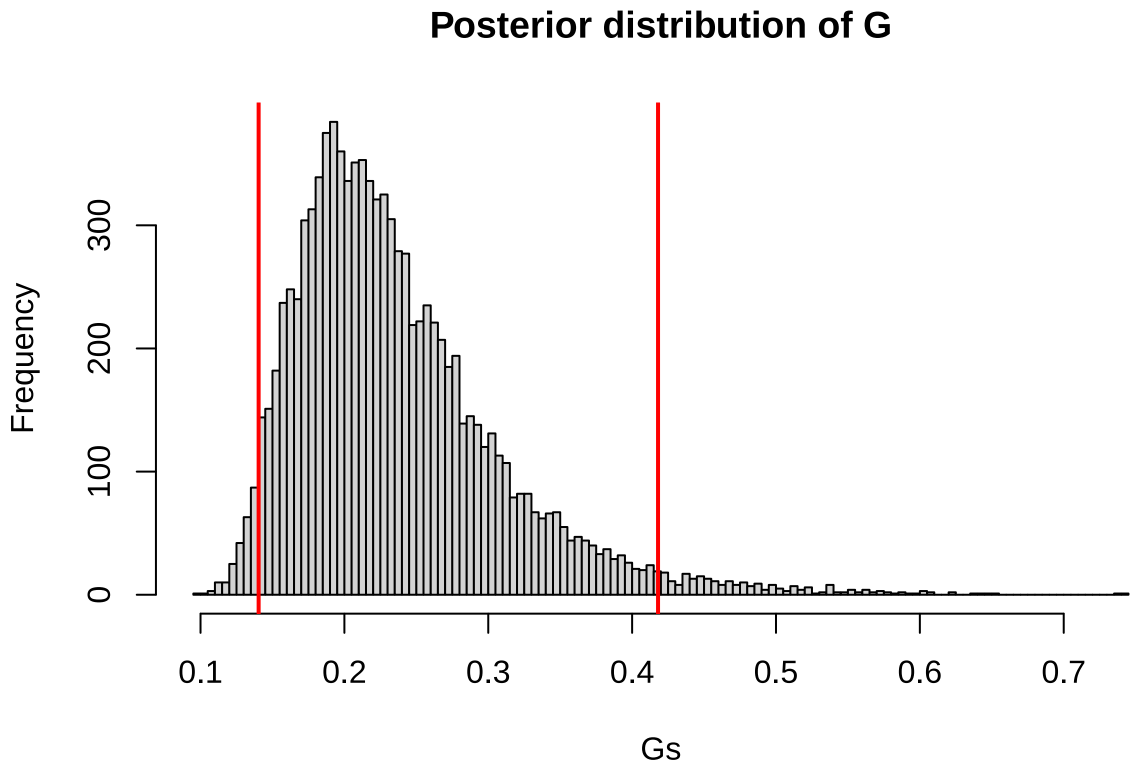
**2.c**

Use the posterior draws from b) to compute a 95% equal tail credible interval for G. A 95% equal tail credible interval (a, b) cuts off 2.5% percent of the posterior probability mass to the left of a, and 2.5% to the right of b.

```
print(quantile(Gs, probs = c(0.025, 0.975)))
```

```
##      2.5%     97.5% 
## 0.1403804 0.4180314
```

```
sortedGs = sort(Gs)
G250 = sortedGs[250]
G9750 = sortedGs[9750]

hist(Gs, main = "Posterior distribution of G", breaks = 100 )
abline(v = G250, col = "red", lwd=2)
abline(v = G9750, col= "red", lwd=2)
```



### 2.d

Use the posterior draws from b) to compute a 95% Highest Posterior Density Interval (HPDI) for G. Compare the two intervals in (c) and (d). [Hint: do a kernel density estimate of the posterior of G using the density function in R with default settings, and use that kernel density estimate to compute the HPDI. Note that you need to order/sort the estimated density values to obtain the HPDI.].

```r
kernel_estimate = density(Gs)
x = kernel_estimate$x
y = kernel_estimate$y

#sort based on y

y_sorted = sort(y)

plot(kernel_estimate)

y_val = 7.0
box_width = (max(kernel_estimate$x)-min(kernel_estimate$x))/512
delta = 0.001
while(sum(kernel_estimate$y[kernel_estimate$y >= y_val]*box_width) < .95){
  #cat("y=", y_val, "\r")
  y_val = y_val - delta
}
print(y_val)
```

```
## [1] 0.674
```

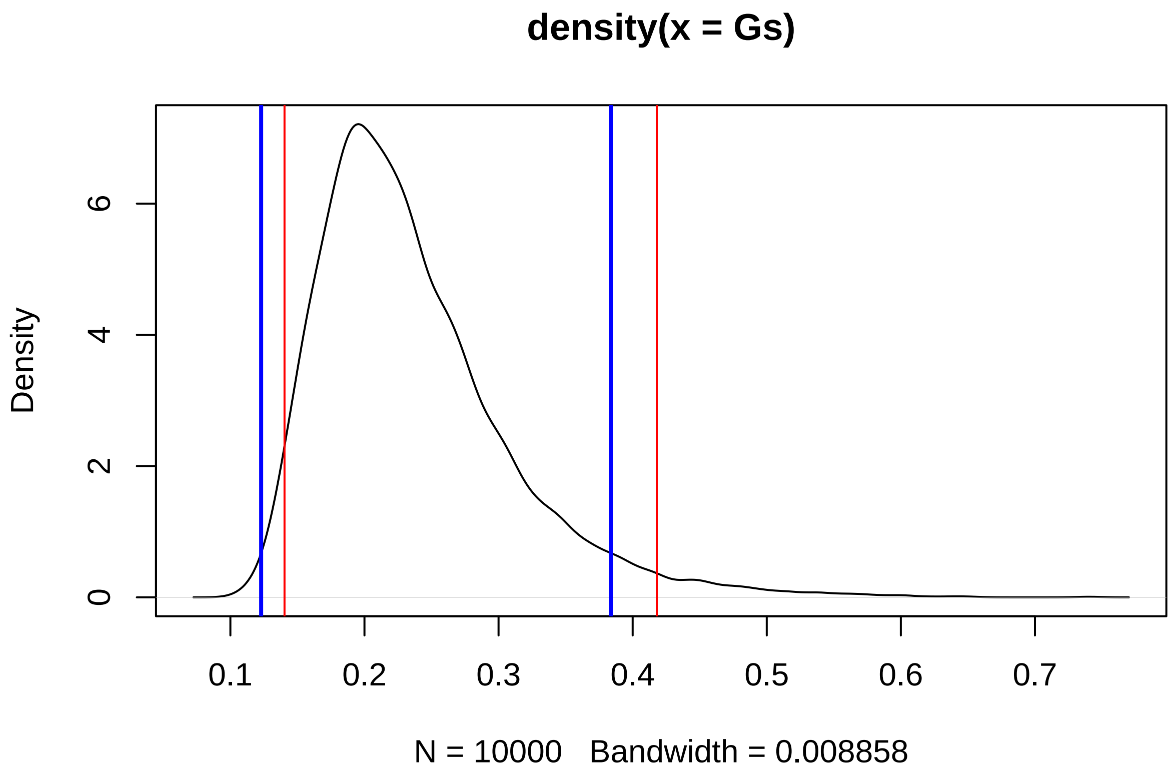
```
x_left_bound = min(kernel_estimate$x[kernel_estimate$y >= y_val])
print(x_left_bound)
```

```
## [1] 0.1229636
```

```
x_right_bound = max(kernel_estimate$x[kernel_estimate$y >= y_val])
print(x_right_bound)
```

```
## [1] 0.3837321
```

```
plot(kernel_estimate)
abline(v=x_left_bound, col = "blue", lwd=2)
abline(v=x_right_bound, col = "blue", lwd=2)
abline(v = G250, col = "red", lwd=1)
abline(v = G9750, col= "red", lwd=1)
```



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

This exercise is concerned with directional data. The point is to show you that the posterior distribution for somewhat weird models can be obtained by plotting it over a grid of values. The data points are observed wind directions at a given location on ten different days. The data are recorded in degrees:

$$(20, 314, 285, 40, 308, 314, 299, 296, 303, 326)$$

, where North is located at zero degrees (see Figure 1 on the next page, where the angles are measured clockwise). To fit with Wikipedia's description of probability distributions for circular data we convert the data into radians $-\pi \leq y \leq \pi$. The 10 observations in radians are

$$(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)$$

. Assume that these data points conditional on $(\mu, \kappa)$ are independent observations from the following von Mises distribution:

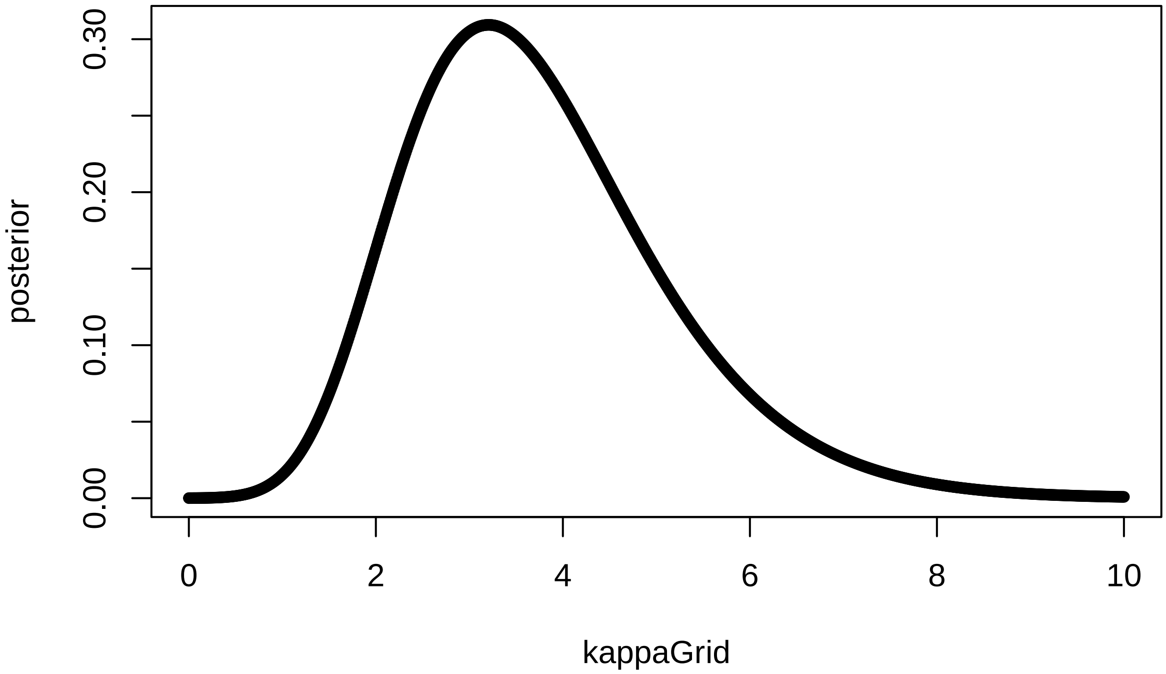
$$p(y|\mu, \kappa) = \frac{exp[\kappa \mathring{u} cos(y - \mu)]}{2\pi I_0(\kappa)}, -\pi \leq y \leq \pi,$$

where $I0(\kappa)$ is the modified Bessel function of the first kind of order zero [see ?besselI in R]. The parameter $\mu(-\pi \leq \mu \leq \pi)$ is the mean direction and $\kappa > 0$ is called the concentration parameter. Large $\kappa$ gives a small variance around $\mu$, and vice versa. Assume that $\mu$ is known to be 2.4. Let $\kappa \sim Exponential(\lambda = 0.5)$ a priori, where $\lambda$ is the rate parameter of the exponential distribution (so that the mean is $1/\lambda$).

### 3.a

Derive the expression for what the posterior $p(\kappa|y, \mu)$ is proportional to. Hence, derive the function $f(\kappa)$ such that $p(\kappa|y, \mu) \propto f(\kappa)$. Then, plot the posterior distribution of $\kappa$ for the wind direction data over a fine grid of $\kappa$ values. [Hint: you need to normalize the posterior distribution of $\kappa$ so that it integrates to one.]

```r
obs = c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54)
n = length(obs)

mu = 2.4

minimum = 0 #adjust later
maximum = 10 #adjust later

delta = 0.01
kappaGrid <- seq(minimum, maximum, by = delta)


f <- function(kappa, mu, y){
  n = length(y)
  (exp(kappa * sum(cos(y - mu))) - kappa/2)/(besselI(kappa, nu = 0))^n
}
posterior = f(kappaGrid, mu, obs)
#normalizing:
posterior = posterior/(sum(posterior)*delta)

plot(posterior ~ kappaGrid, pch = 20)
```



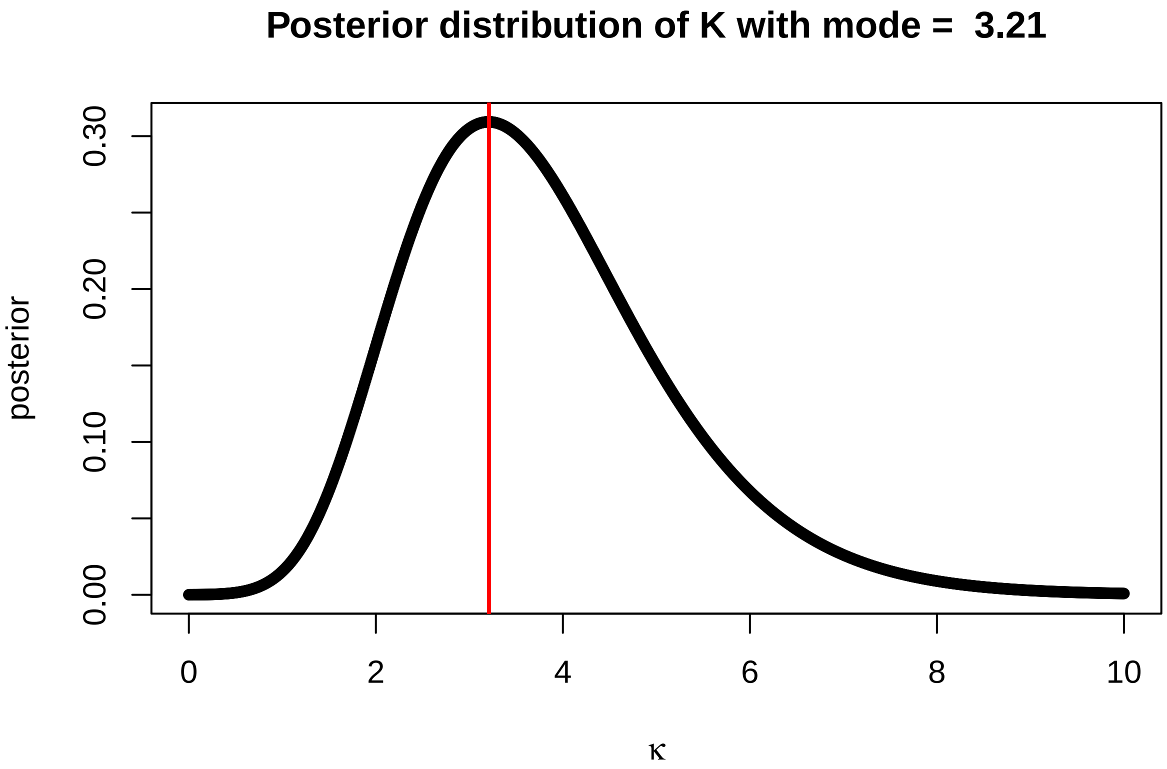
### 3.b

Find the (approximate) posterior mode of $\kappa$ from the information in a).

```
idx_mode = which.max(posterior)
post_mode = kappaGrid[idx_mode]
print(post_mode)
```

```
## [1] 3.21
```

```
plot(posterior ~ kappaGrid, pch = 20, xlab = expression(kappa), main = paste("Posterior distribution of
abline(v = post_mode, col = "red", lwd=2)
```


Posterior distribution of K with mode =  3.21
posterior
0.00
0.10
0.20
0.30
0
2
4
6
8
10
κ