Lab 1

2024-03-29

```r
library(ggplot2)
library(patchwork)
```

1 - Daniel Bernoulli

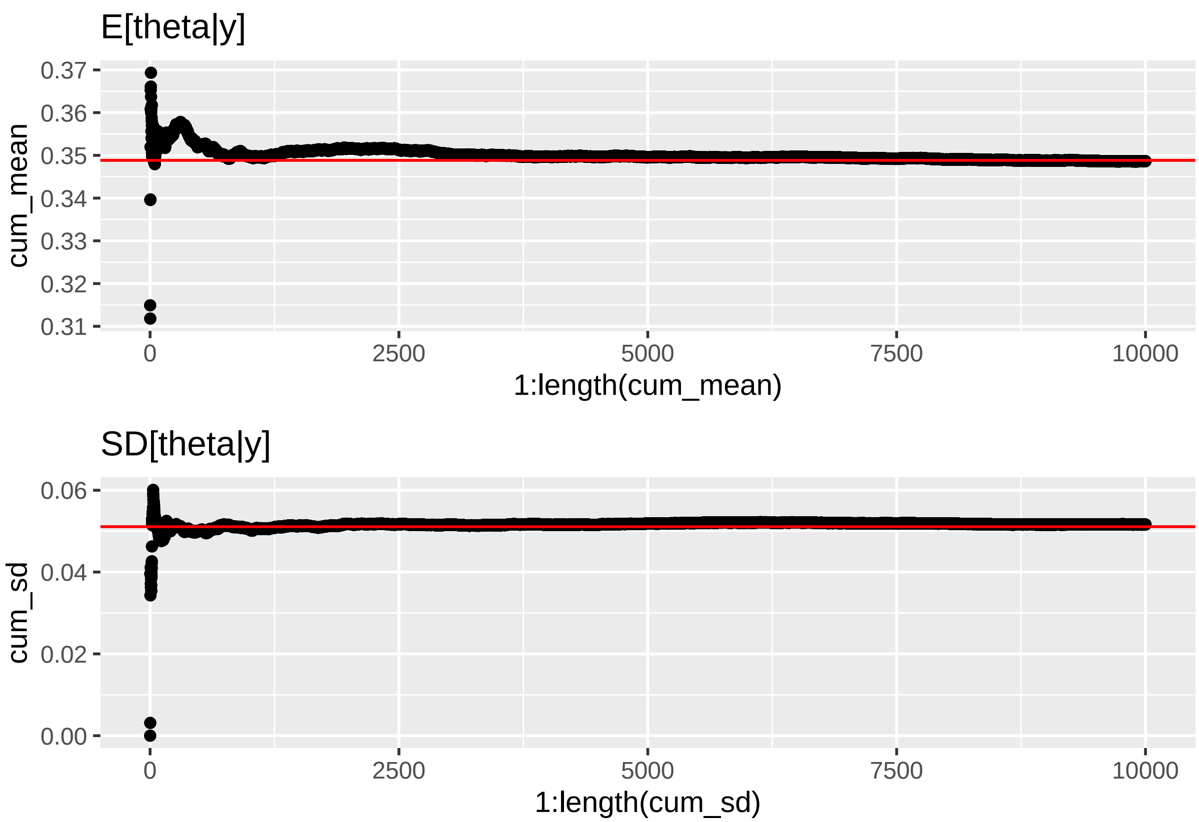
Let y1, …, yn|theta Bern(theta), and assume that you have obtained a sample with s = 22 successes in n = 70 trials. Assume a Beta(alpha0, beta0) prior for and let alpha0 = beta0 = 8.

a)

Draw 10000 random values (nDraws = 10000) from the posterior theta|y Beta(alpha0+ s, beta0 + f), where y = (y1, . . . , yn), 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
n <- 70    # Sample trials
s <- 22    # Sample successes
f <- n - s # Sample failures

a0 <- b0 <- 8    # Prior parameters
nDraws <- 10000    # Number of random draws from posterior

an = a0 + s
bn = b0 + f
rdraw <- rbeta(nDraws, an, bn)

cum_n <- 1:nDraws
cum_mean <- cumsum(rdraw) / cum_n
cum_mean2 <- cumsum(rdraw ** 2) / cum_n
cum_var <- (cum_mean2 - cum_mean ** 2) * (n / (n - 1))
cum_sd <- sqrt(cum_var)

anlyt_mean <- an / (an + bn)
anlyt_sd <- sqrt(an * bn / ((an + bn) ** 2 * (an + bn + 1)))

p1 <- ggplot() +
  geom_point(aes(x=1:length(cum_mean), y=cum_mean)) +
  geom_hline(yintercept=anlyt_mean, color="red") +  #or abline()
  ggtitle("E[theta|y]")
```

```r
p2 <- ggplot() +
  geom_point(aes(x=1:length(cum_sd), y=cum_sd)) +
  geom_hline(yintercept=anlyt_sd, color="red") +
  ggtitle("SD[theta|y]")

p1 / p2
```



b)

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

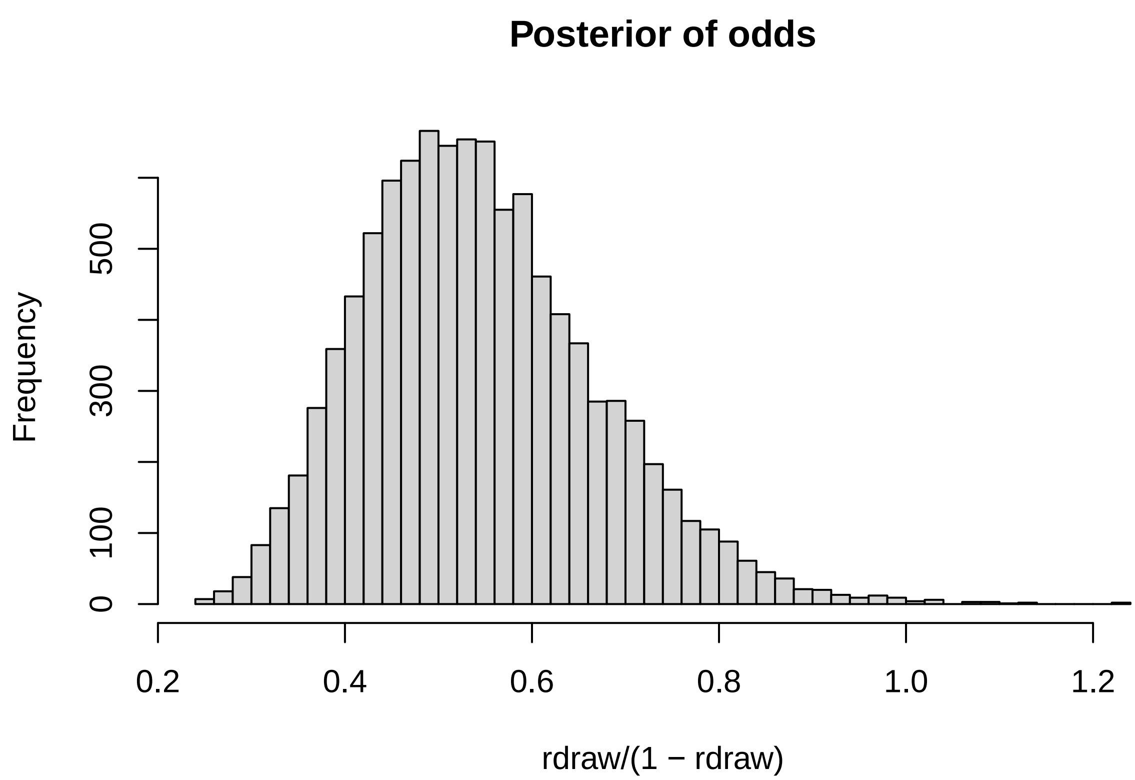
```r
p_draw = sum(rdraw > 0.3) / length(rdraw)
p_anlyt = pbeta(0.3, an, bn, lower.tail=FALSE)
```

```
## Random draw from posterior:
##
##     Pr(theta > 0.3 | y) =  0.8242
##
## Exact value from Beta distribution:
##
##     Pr(theta > 0.3 | y) =  0.8285936
```

c)

Draw 10000 random values from the posterior of the odds phi = 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].

```
hist(rdraw / (1 - rdraw), breaks=50, main="Posterior of odds")
```



2 - Log-normal distribution and the Gini coefficient

A common model for non-negative continuous variables is the log-normal distribution.

$$y \sim logNormal(\mu, \sigma 2) => log(y) \sim Normal(\mu, \sigma 2)$$

Let y1,...,yn| , logN(mu,sigma^2), where mu = 3.6 is assumed to be known but sigma is unknown with non-informative prior p(sigma^2) 1/sigma^2. The posterior for sigma^2 is the Inv−chi2(n,to^2) distribution.

a)

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

Normal model with unknown variance:

model:

$$y1, ..., yn|\mu, \sigma \sim logN(\mu, \sigma^2)$$

Non-informative prior

$$p(\sigma 2) \propto 1/\sigma 2$$

Posterior:

mu is given here i.e 3.6 (Only in the below formula we take it as n no matter if it is "n" or "n-1")

$$\mu|\sigma 2, x \sim N(x, \sigma^2/n),$$

(Note, in the slides it was given "n-1" , but here we taken "n" since that is what is given in the question)

$$\sigma 2|x \sim Inv - \chi 2(n, \tau 2)$$

where (in the slides it was given "n-1" , but here we taken "n" since that is what is given in the question))

$$\tau 2 = (\sum(logyi - \mu)^2)/n$$

To simulate from the posterior:

(Note, in the slides it was given "n-1" , but here we taken "n" since that is what is given in the question)

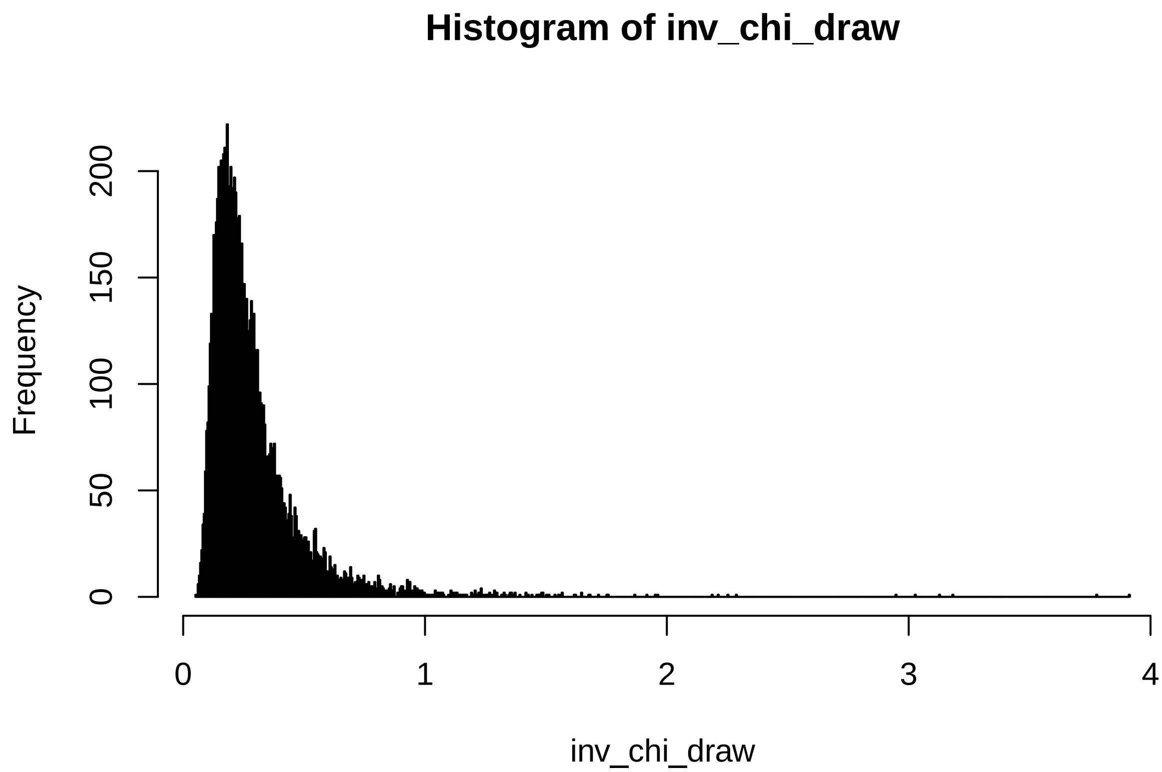
$$X \sim \chi 2(n)$$

A draw from Inv—chi2(n,to2) distribution((Note, in the slides it was given "n-1" , but here we taken "n" since that is what is given in the question))

$$\sigma 2 = (n * \tau^2)/X$$

Then draw mu from the above formula but in this case we assume it to be 3.6

```r
obs <- c(33, 24, 48, 32, 55, 74, 23,17)
mu <- 3.6

#Sample variance calculation
values <- c()
for (i in 1:length(obs)){
  values[i] <- (log(obs[i])-mu)^2
}

sample_var <- sum(values)/8

# Simulation from posterior:
inv_chi_draw <- c()
post_draw <- function(n){
  for(i in 1:n){
    chi_draw <- rchisq(1,8)
    inv_chi_draw[i]<<-((8)*sample_var)/(chi_draw)
  }
  hist(inv_chi_draw,breaks=1000)
}

post_draw(10000)
```



b)

We need to plot the posterior distribution of the gini coefficient which follows a log N (mu, sigma^2) distribution. (G|y)

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

and Φ is a cumulative distribution function (CDF) for the standard normal distribution with mean zero and unit variance. So we use pnorm() to calculate this since pnorm() gives the distribution function while dnorm() gives the density function.

```
gini_values <- c()
gini <- function(n){
  for(i in 1:n){
    gini_values[i]<<-2*pnorm(sqrt(inv_chi_draw[i])/sqrt(2))-1
  }
  hist(gini_values)
}

gini(10000)
```

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.

Approximate 95% credible interval for G

$$E(G|y) \pm 1.96 \cdot SD(G|y)$$

```
upper <- mean(gini_values) + 1.96*sd(gini_values)
lower <- mean(gini_values) - 1.96*sd(gini_values)
```

```
## [1] "Upper limit:"
```

```
## [1] 0.4340629
```

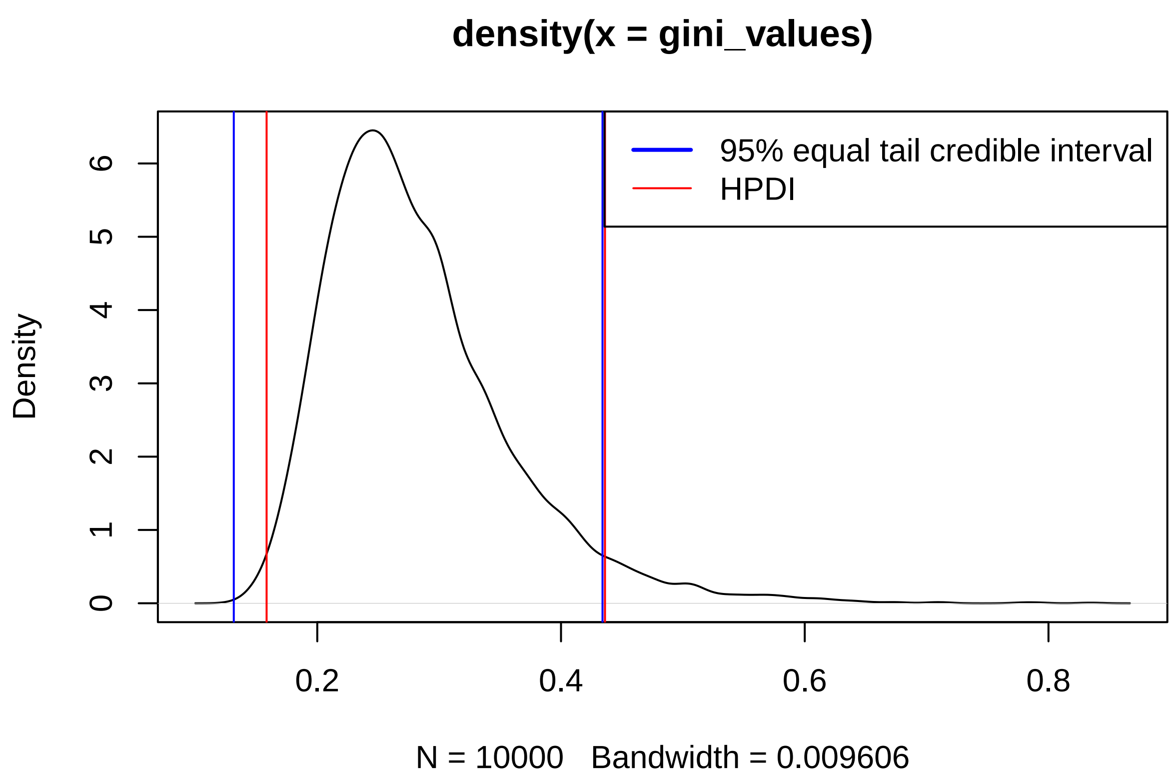
```
## [1] "lower limit:"
```

```
## [1] 0.1315154
```

d)

Use the posterior draws from b) to compute a 95% Highest Posterior Density Interval (HPDI) for G.

```
gini_den<-density(gini_values)
gini_den_df <- data.frame(x = gini_den$x, y = gini_den$y)

#x gives the corrdinates of the points where the density is estimated and y gives the estimated
#density values.

gini_den_df <- gini_den_df[order(gini_den_df$y),]
gini_den_df<- data.frame(x = gini_den_df$x, y = gini_den_df$y)
gini_den_cum = cumsum(gini_den_df$y)/sum(gini_den_df$y) # we divde so that it sums to 1.
value = which(gini_den_cum  >= 0.05)[1]
gini_den_df = gini_den_df[(value+1):length(gini_den$y),]
gini_den_interval <- c(min(gini_den_df$x),max(gini_den_df$x))

plot(gini_den)
abline(v=gini_den_interval[1],col ="red")
abline(v=gini_den_interval[2], col = "red")

abline(v=upper,col ="blue")
abline(v=lower, col = "blue")

legend("topright", legend = c("95% equal tail credible interval", "HPDI"),
       col = c("blue", "red"), lty = 1, lwd = c(2, 1))
```



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

a)

Derive the expression for what the posterior p(x|y, k) is proportional to. Hence, derive the function f (k) such that p(x|y, mu)   f (k). Then, plot the posterior distribution of k for the wind direction data over a fine grid of k values.

Model:

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

Likelihood:

$$p(y|\mu, \kappa) = \prod_{i=1}^{10} \frac{\exp(\kappa cos(y_i - \mu))}{2\pi I_0(\kappa)}$$

$$= \frac{\exp(\kappa \sum_{i=1}^{10} cos(y_i - \mu))}{(2\pi I_0(\kappa))^{10}}$$

Prior: (exponential distribution)

$$p(\kappa) = 0.5e^{-0.5\kappa}$$

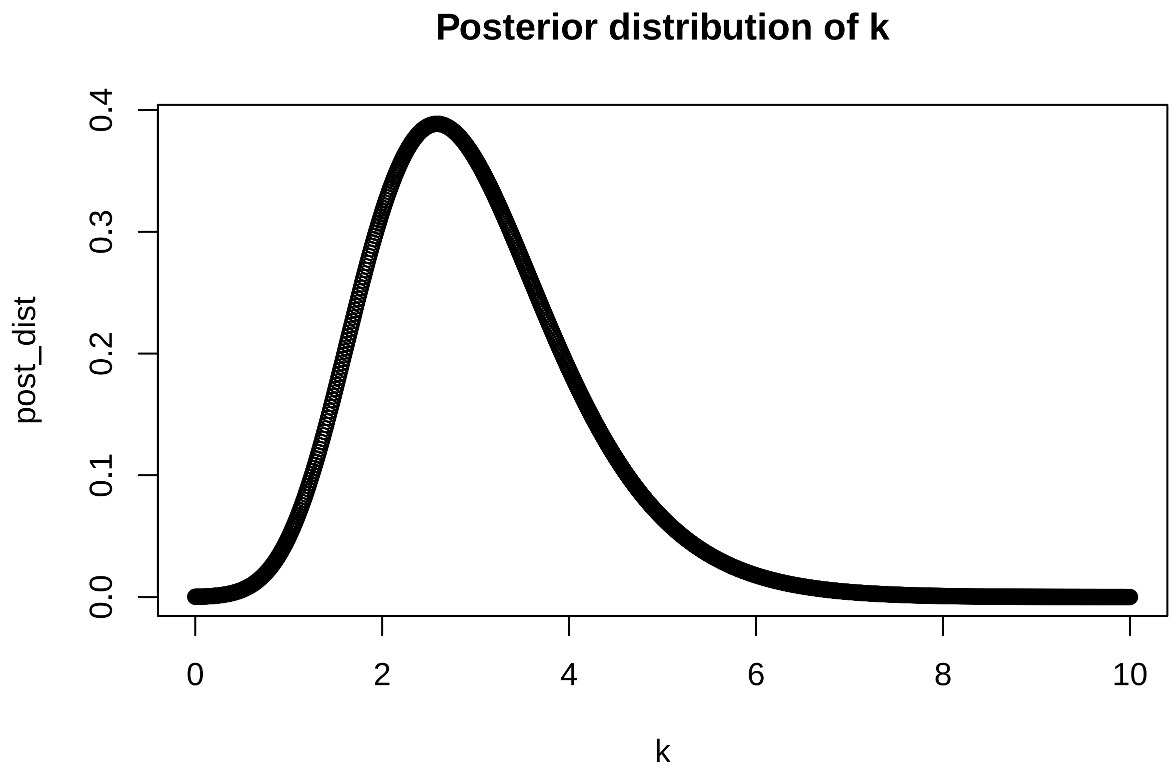
Posterior:

$$p(\kappa|y, \mu) \propto \frac{\exp(\kappa \sum_{i=1}^{10} cos(y_i - \mu))}{2\pi I_0(\kappa)} * \exp(-0.5\kappa)$$

$$\propto \frac{\exp[\kappa(-0.5 + \sum_{i=1}^{10} cos(y_i - \mu))]}{I_0(\kappa)^{10}}$$

```r
y <- c(-2.79, 2.33, 1.83, -2.44, 2.23, 2.33, 2.07, 2.02, 2.14, 2.54) # Data points
mu <- 2.4 # Given parameter mu

post <- function(k, y, mu) {
  num <- exp(k * (-0.5 + sum(cos(y - mu))))
  denom <- besselI(k, 0) ** 10

  res = num / denom
  res[is.na(res)] <- 0

  return (res)
}

post_norm <- function(k, y, mu, c_norm) {
  res = post(k, y, mu) / c_norm

  return (res)
}

#[Hint: you need to normalize the posterior distribution of   so that it integrates to one.]
c_norm <- integrate(post, 0, Inf, y=y, mu=mu)$value
```

```r
k <- seq(from=0, to=10, length.out=1000)
post_dist <- post_norm(k, y, mu, c_norm)

plot(k, post_dist, main="Posterior distribution of k")
```



Find the (approximate) posterior mode of from the information in a). ## b)

```r
paste0("Approx. mode of distribution of k: ", round(k[which.max(post_dist)], 3))
```

```
## [1] "Approx. mode of distribution of k: 2.583"
```