Bayesian Lab1

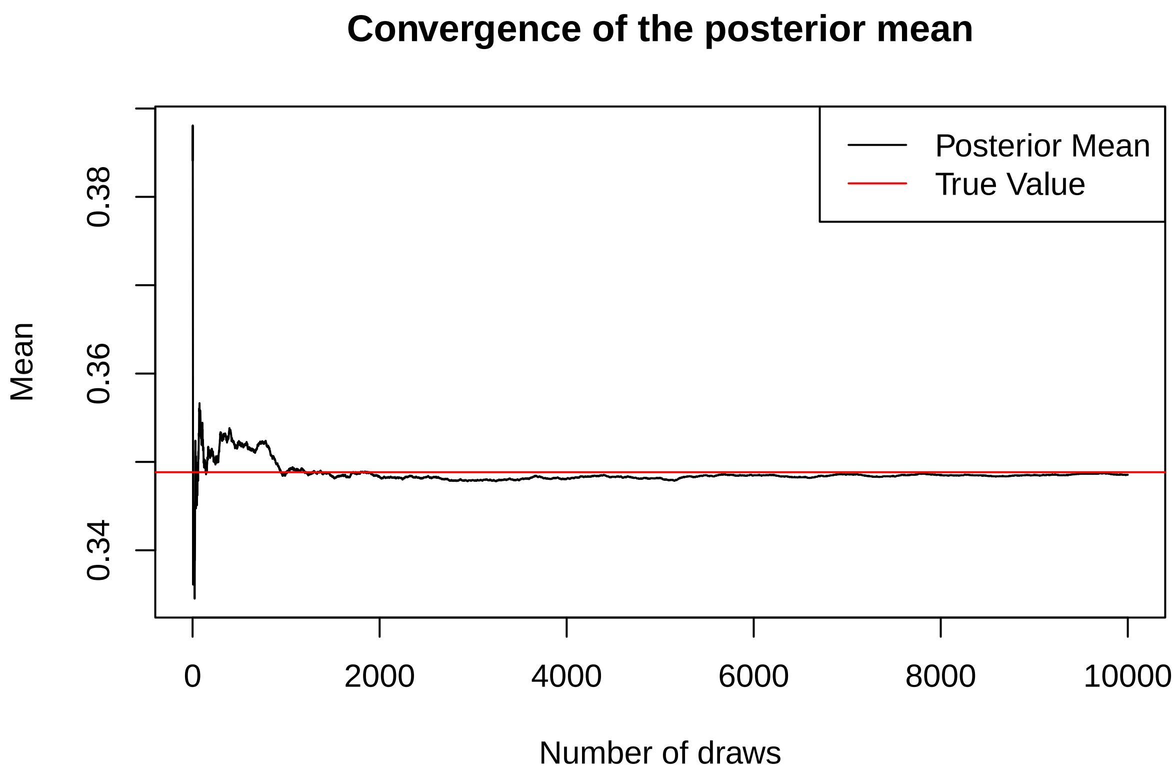
Xuan Wang & Lepeng Zhang

2024-04-03

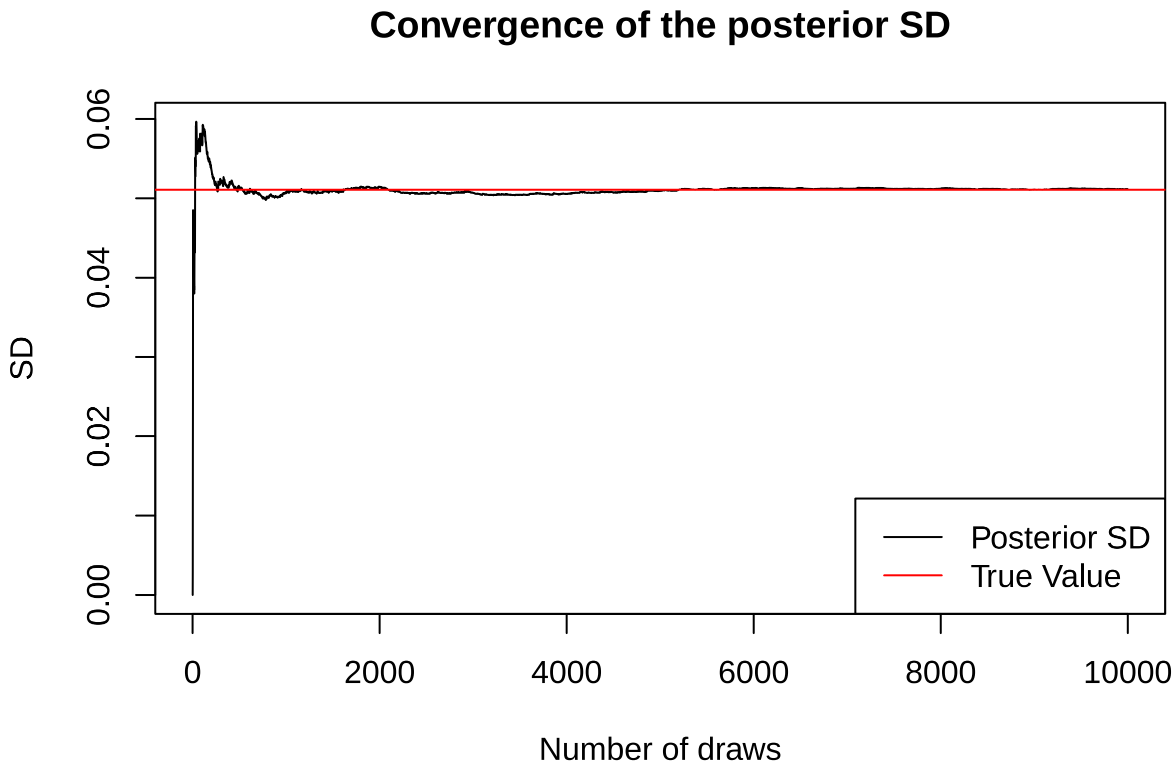
Question 1 Daniel Bbernoulli

1a)

```
# Set parameters
set.seed(12345)
nDraws <- 10000
s <- 22
n <- 70
alpha0 <- 8
beta0 <- 8
alpha <- alpha0 + s
beta <- beta0 + n - s

# True values
true_mean <- alpha / (alpha + beta)
true_sd <- sqrt((alpha * beta) / ((alpha + beta)^2 * (alpha + beta + 1)))

# (a) Draw random values from the posterior
posterior <- rbeta(nDraws, alpha, beta)
posterior_mean <- cumsum(posterior) / 1:nDraws
posterior_sd <- sqrt(cumsum((posterior - posterior_mean)^2) / 1:nDraws)

# Plot the convergence of the posterior mean and SD
plot(1:nDraws, posterior_mean, type = "l", xlab = "Number of draws", ylab = "Mean", main = "Convergence
abline(h = true_mean, col = 'red')
legend('topright', legend = c('Posterior Mean', 'True Value'), col = c('black', 'red'), lty = 1)
```



```
plot(1:nDraws, posterior_sd, type = "l", xlab = "Number of draws", ylab = "SD", main = "Convergence of 
abline(h = true_sd, col = 'red')
legend('bottomright', legend = c('Posterior SD', 'True Value'), col = c('black', 'red'), lty = 1)
```



1b)

```
# (b) Compute the posterior probability
posterior_prob <- sum(posterior > 0.3) / nDraws
exact_prob <- 1 - pbeta(0.3, alpha, beta)
cat("Posterior probability: ", posterior_prob, "\n")
```

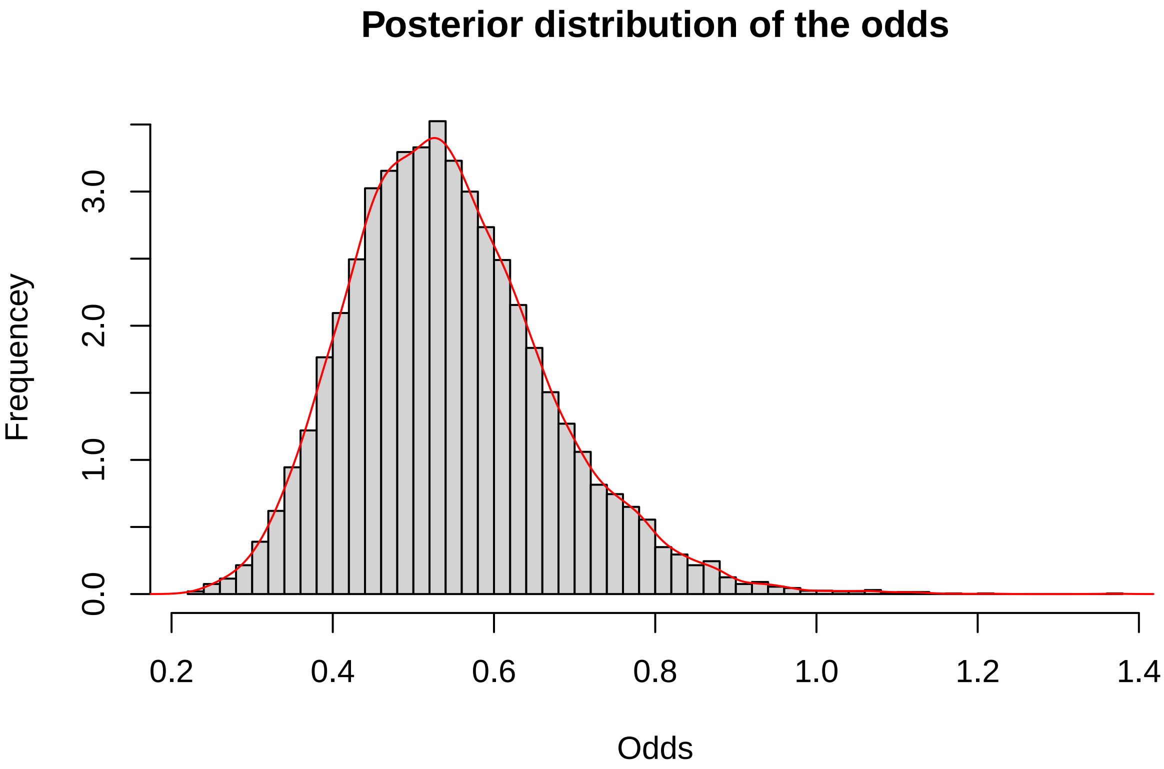
```
## Posterior probability:  0.8294
```

```
cat("Exact probability: ", exact_prob, "\n")
```

```
## Exact probability:  0.8285936
```

1c)

```
# (c) Draw random values from the posterior of the odds
odds <- posterior / (1 - posterior)
hist(odds, breaks=50, main = "Posterior distribution of the odds", xlab = "Odds", ylab = "Frequencey", 
lines(density(odds), col = "red")
```

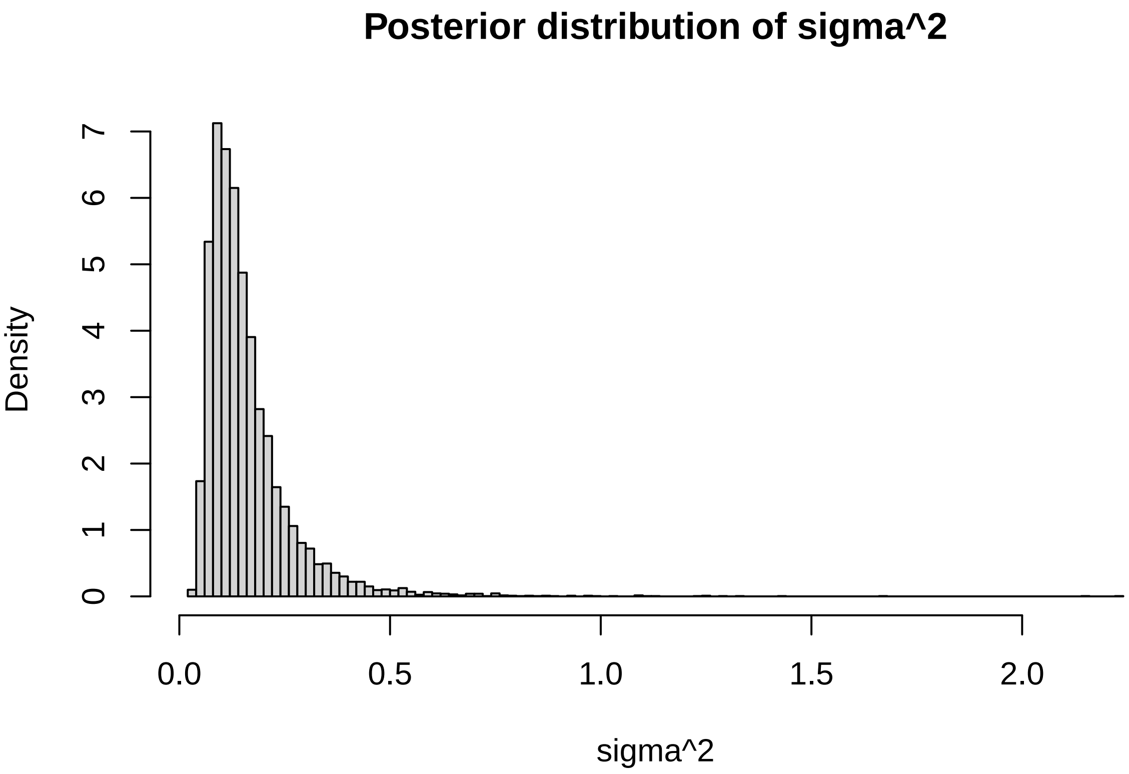


Question 2 Log-normal distribution and Gini coefficient

2a)

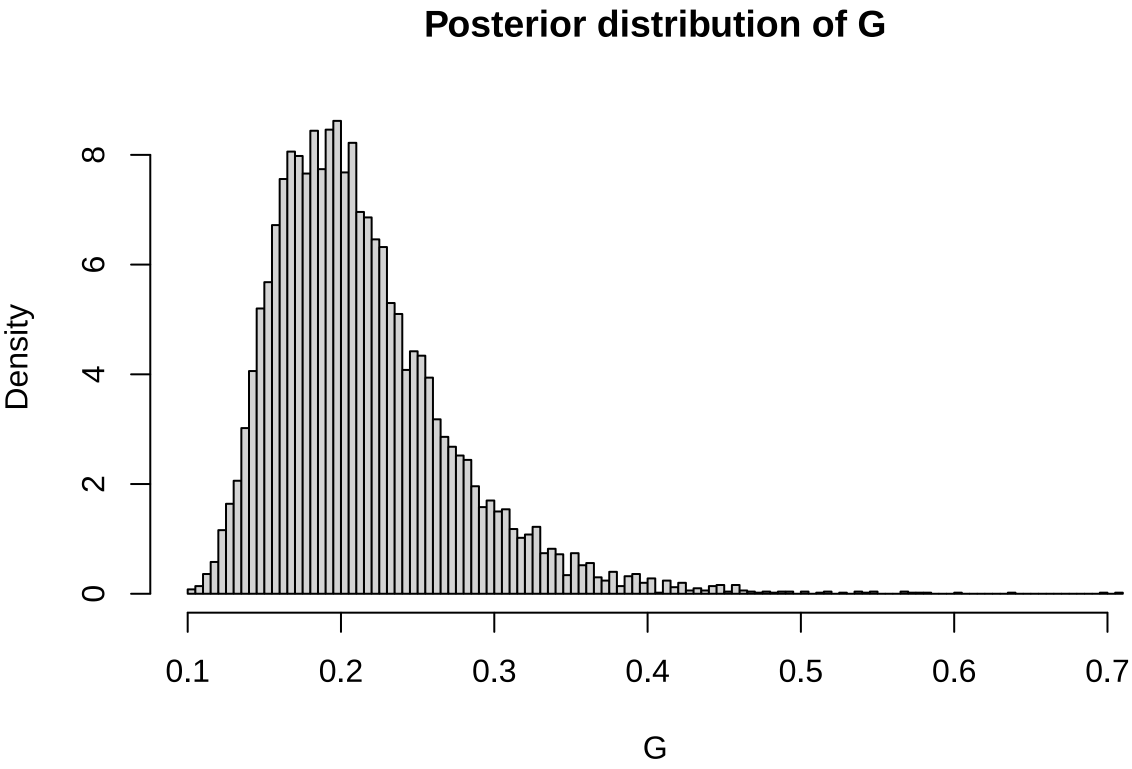
```
# Load necessary libraries
library(MASS)
set.seed(12345)
# Define the data
income <- c(33, 24, 48, 32, 55, 74, 23, 17)
```

```
log_income <- log(income)

# Known parameter
mu <- 3.6

# Calculate tau^2
tau_sq <- sum((log_income - mu)^2) / length(income)

# Draw 10000 random values from the posterior of sigma^2
posterior_sigma2 <- 1 / rchisq(10000, df = length(income), ncp = tau_sq)

# Plot the posterior distribution of sigma^2
hist(posterior_sigma2, breaks = 100, main = "Posterior distribution of sigma^2", xlab = "sigma^2", freq
```



2b)

```
# Compute the Gini coefficient for each sample
G_samples <- 2 * pnorm(sqrt(posterior_sigma2/2)) - 1

# Plot the posterior distribution of G
hist(G_samples,breaks = 100,  main = "Posterior distribution of G", xlab = "G", freq = FALSE)
```



2c)

```
credible_interval <- quantile(G_samples, c(0.025, 0.975))
credible_interval
```

```
##      2.5%     97.5%
## 0.1321501 0.3578281
```

2d)

```
#Compute a 95% Highest Posterior Density Interval (HPDI) for G
density_G <- density(G_samples)
sorted_density_G <- sort(density_G$y, decreasing = TRUE)
cumulative_density_G <- cumsum(sorted_density_G) / sum(sorted_density_G)
HPDI <- density_G$x[cumulative_density_G >= 0.025 & cumulative_density_G <= 0.975]

# Compare the two intervals
print(paste("95% equal tail credible interval for G: [", credible_interval[1], ", ", credible_interval[
```

```
## [1] "95% equal tail credible interval for G: [ 0.132150101093225 ,  0.357828061148171 ]"
```

```
print(paste("95% HPDI for G: [", min(HPDI), ", ", max(HPDI), "]"))
```

```
## [1] "95% HPDI for G: [ 0.0809303354333169 ,  0.328908462592719 ]"
```

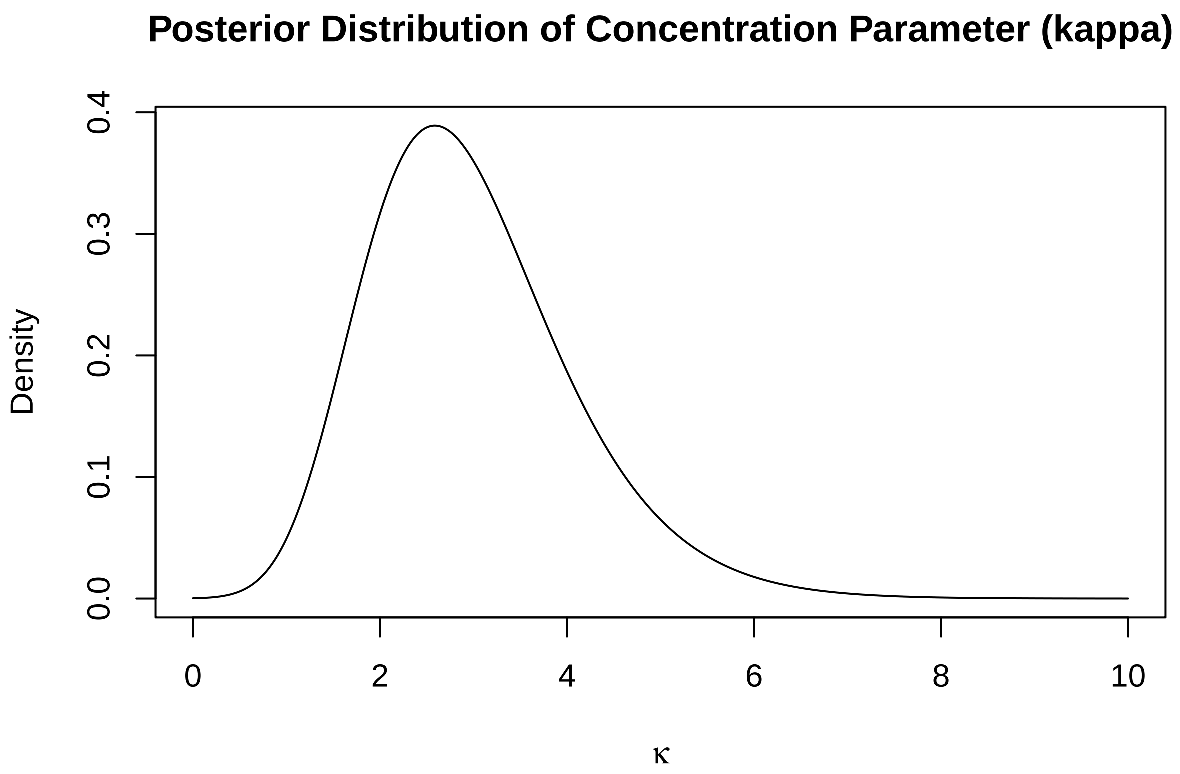
Question 3 Bayesian iinference for the concentration parameter in the von Mises distribution

3a)

```r
# Required Libraries
library(REdaS)
```

```
## Loading required package: grid
```

```r
# Data
radians <- c(-2.79, 2.33, 1.83, -2.44,2.23,2.33, 2.07, 2.02, 2.14, 2.54)
# Known Parameters
mu <- 2.4
lambda <- 0.5

# Grid of kappa values
kappa_values <- seq(0, 10, length.out = 1000)

# Prior
prior <- dexp(kappa_values, rate = lambda)

# Likelihood
likelihood <- sapply(kappa_values, function(kappa) {
  density_values <- exp(kappa * cos(radians - mu)) / (2 * pi * besselI(kappa, 0))
  prod(density_values)
})

# Posterior
posterior <- likelihood * prior
posterior <- posterior / (sum(posterior) * 10/1000)  # normalize to make it a proper distribution

# Plotting the Posterior
plot(kappa_values, posterior, type = "l", xlab = expression(kappa), ylab = "Density",
     main = "Posterior Distribution of Concentration Parameter (kappa)")
```



3b)

```r
# Posterior Mode
posterior_mode <- kappa_values[which.max(posterior)]
print(paste("Posterior mode of kappa: ", round(posterior_mode, 2)))
```

```
## [1] "Posterior mode of kappa:  2.58"
```