Bayesian Learning (732A91) Lab1 Report

Christoforos Spyretos (chrsp415) & Marketos Damigos (marda352)

2022-04-08

Assignment 1 *Daniel Bernoulli*

Let $y_1, y_2, ..., y_n \sim Bern(\theta)$, and the obtained sample has 13 successes out of 50 trails (37 failures). The $Beta(a_0, b_0)$ prior has $a_0 = b_0 = 5$.

Task 1a

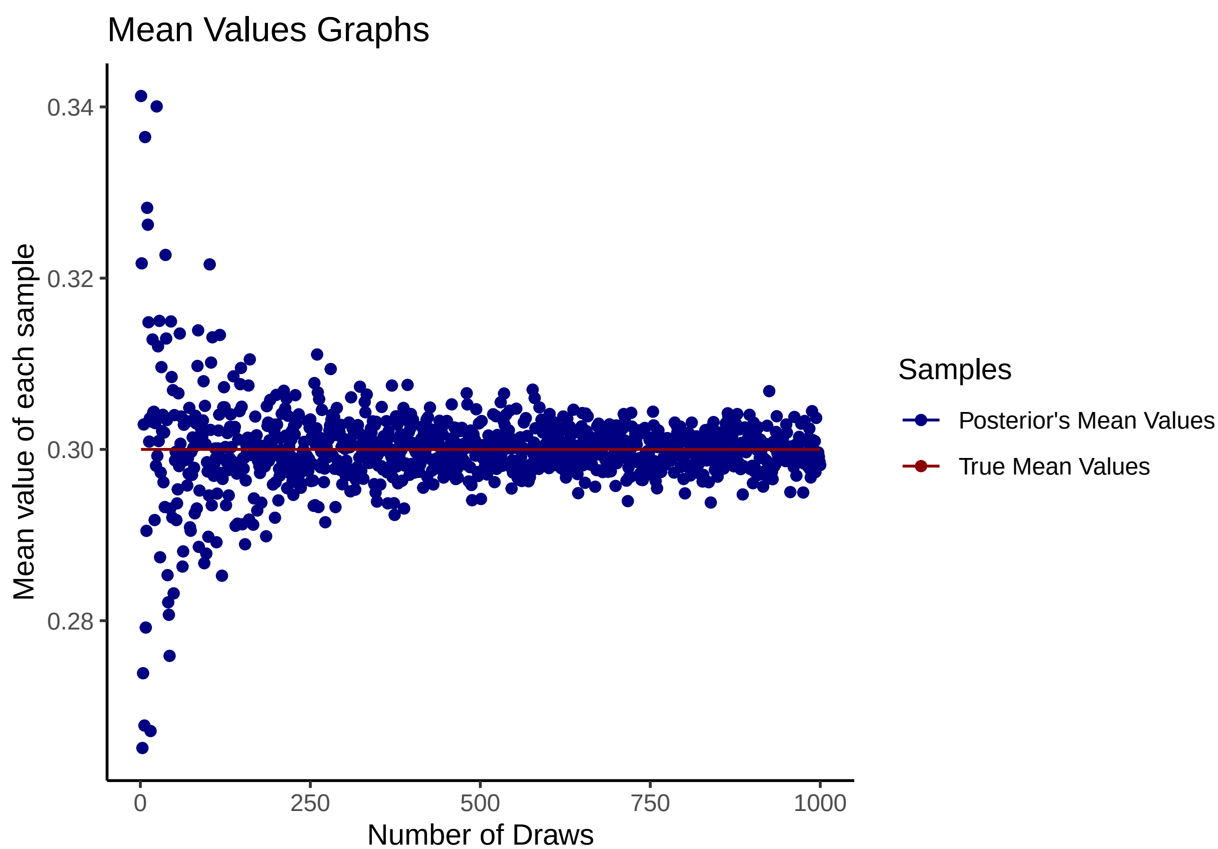
The mean value and the standard deviation of the Beta distribution for θ are calculated by the below formulas:

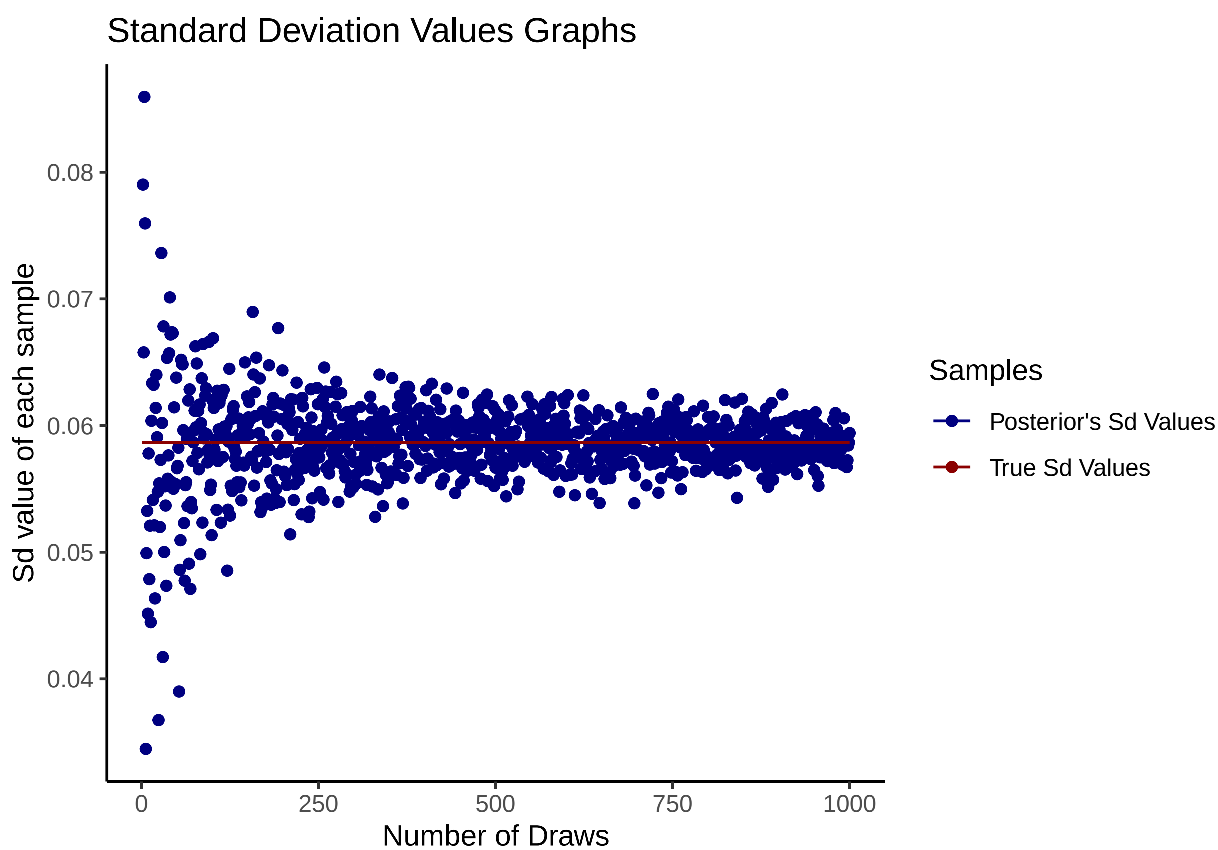
$$\begin{aligned} E[\theta] &= \frac{a_0 + s}{a_0 + s + b_0 + f} \\ &= \frac{18}{60} \\ &= 0.3 \end{aligned}$$

$$\begin{aligned} Var[\theta] &= \frac{(a_0 + s)(b_0 + f)}{\Big((a_0 + s) + (b_0 + f)\Big)^2 \Big((a_0 + s) + (b_0 + f) + 1\Big)} \\ &= \frac{18 \cdot 42}{(18 + 42)^2 (18 + 42 + 1)} \\ &= 0.003442623 \end{aligned}$$

$$SD[\theta] = \sqrt{Var[\theta]} = 0.05867387$$

Where a_0 and b_0 are the arguments of the Beta prior, s is the number of successes and f is the number of failures.


Mean Values Graphs
0.34
0.32
0.30
0.28
Mean value of each sample
0
250
500
750
1000
Number of Draws
Samples
Posterior's Mean Values
True Mean Values

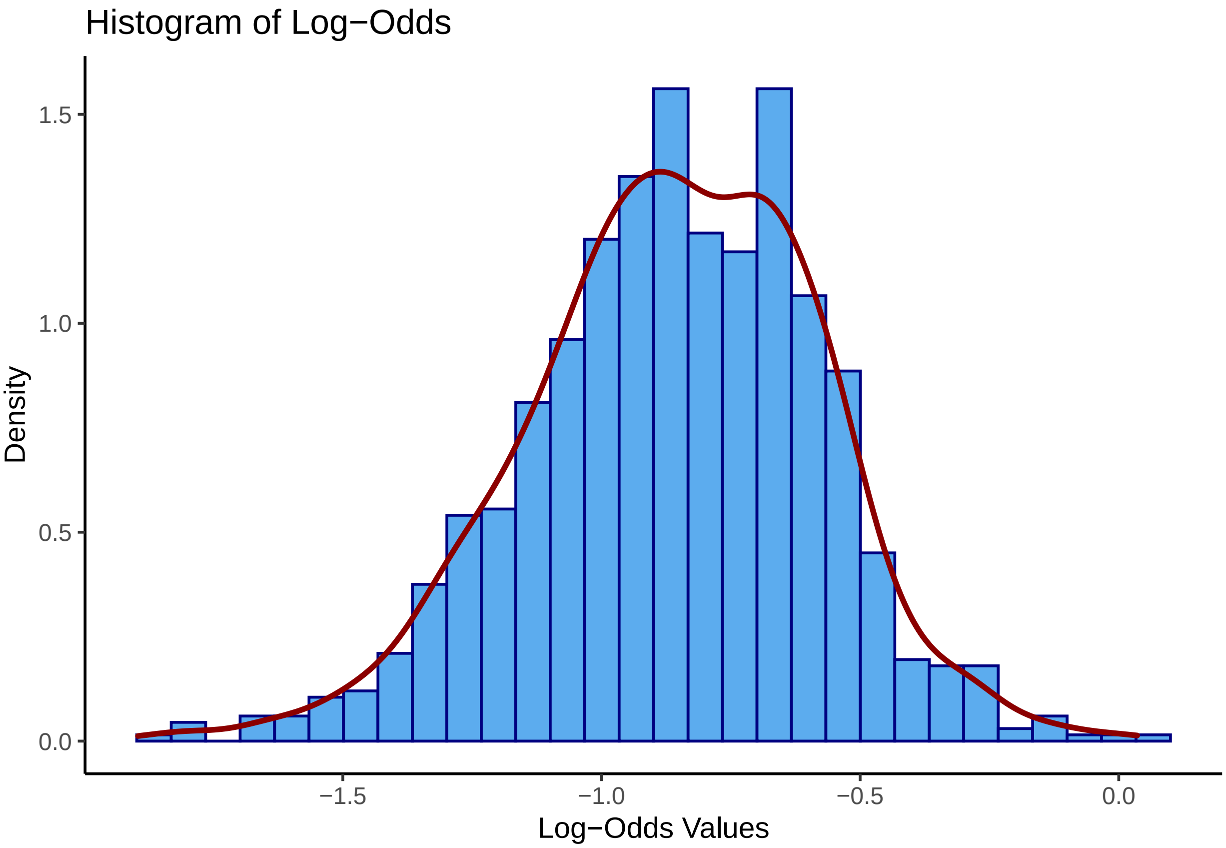

Standard Deviation Values Graphs
0.08
0.07
0.06
0.05
0.04
Sd value of each sample
0
250
500
750
1000
Number of Draws
Samples
Posterior's Sd Values
True Sd Values

From the above plots, it could be seen that both posterior's mean and standard deviation values converge to the actual mean and standard deviation values, respectively. More specifically, between 0 and approximately 250 draws in both graphs, some of the posterior's values abstain from true values. However, after the 250 draws, the posterior's values start to converge to the true ones in both graphs.

Task 1b

The posterior probability $P(\theta < 0.3|y)$ equals 0.506, and the exact probability value from the Beta posterior is 0.5150226; thus, it could be assumed that both values are pretty similar.

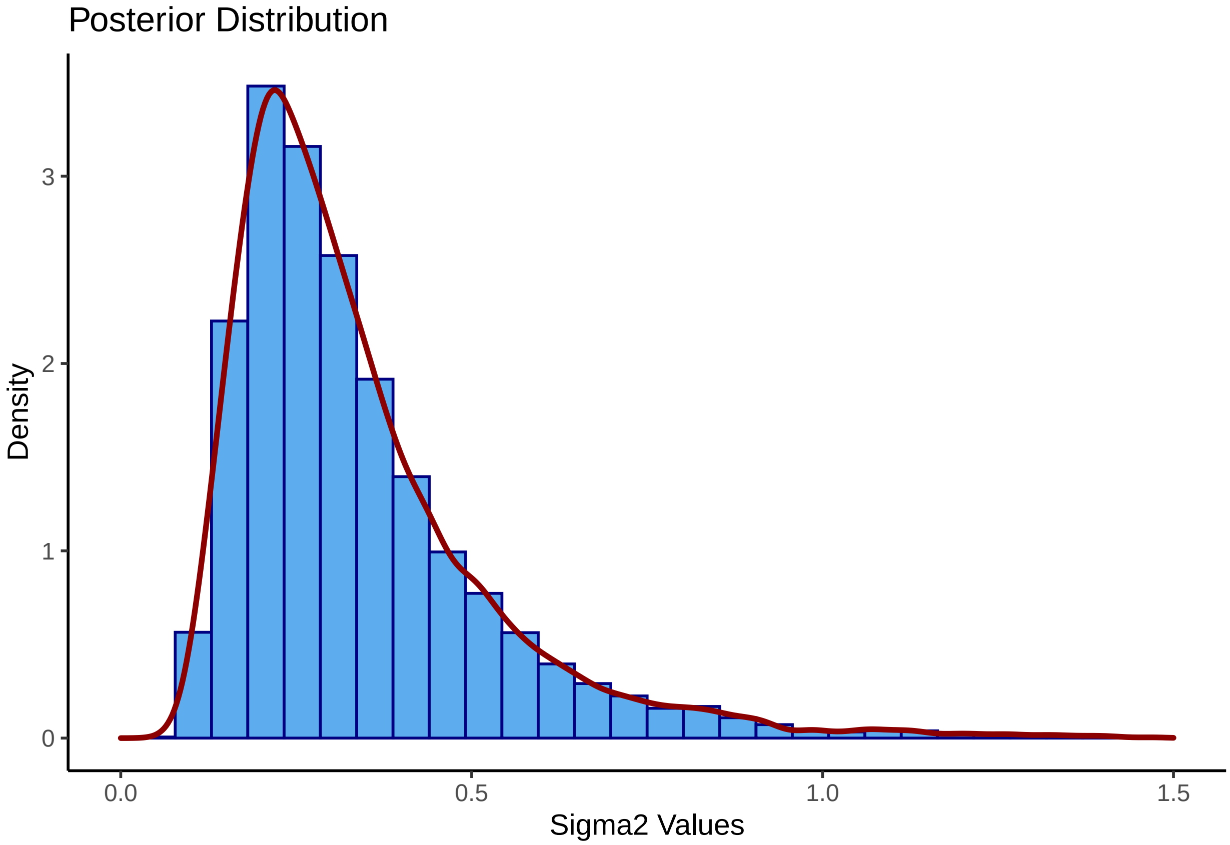
Task 1c



The above plot illustrates the density of the log-odds values $\phi = log\frac{\theta}{1-\theta}$, where θ takes values from simulated draws from the Beta posterior.

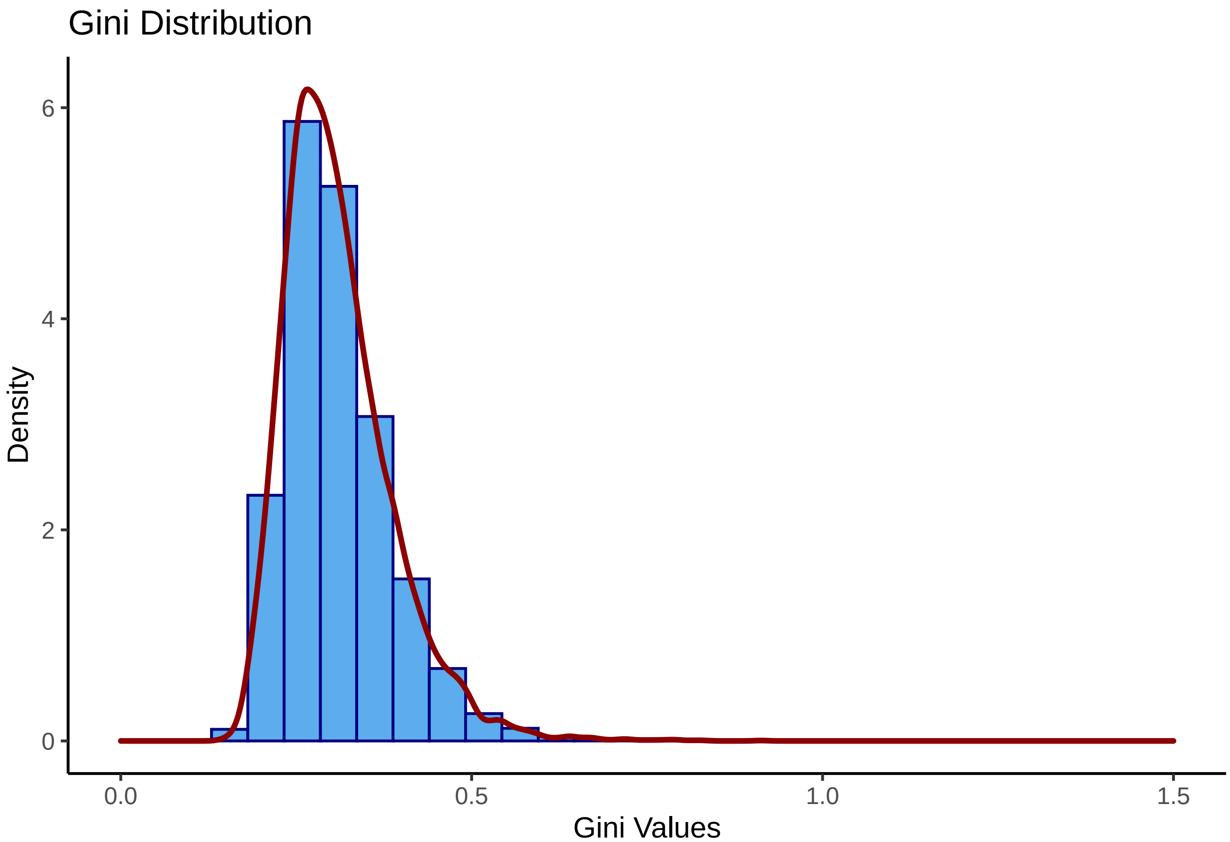
Assignment 2 *Log-normal distribution and the Gini coefficient.*

Task 2a



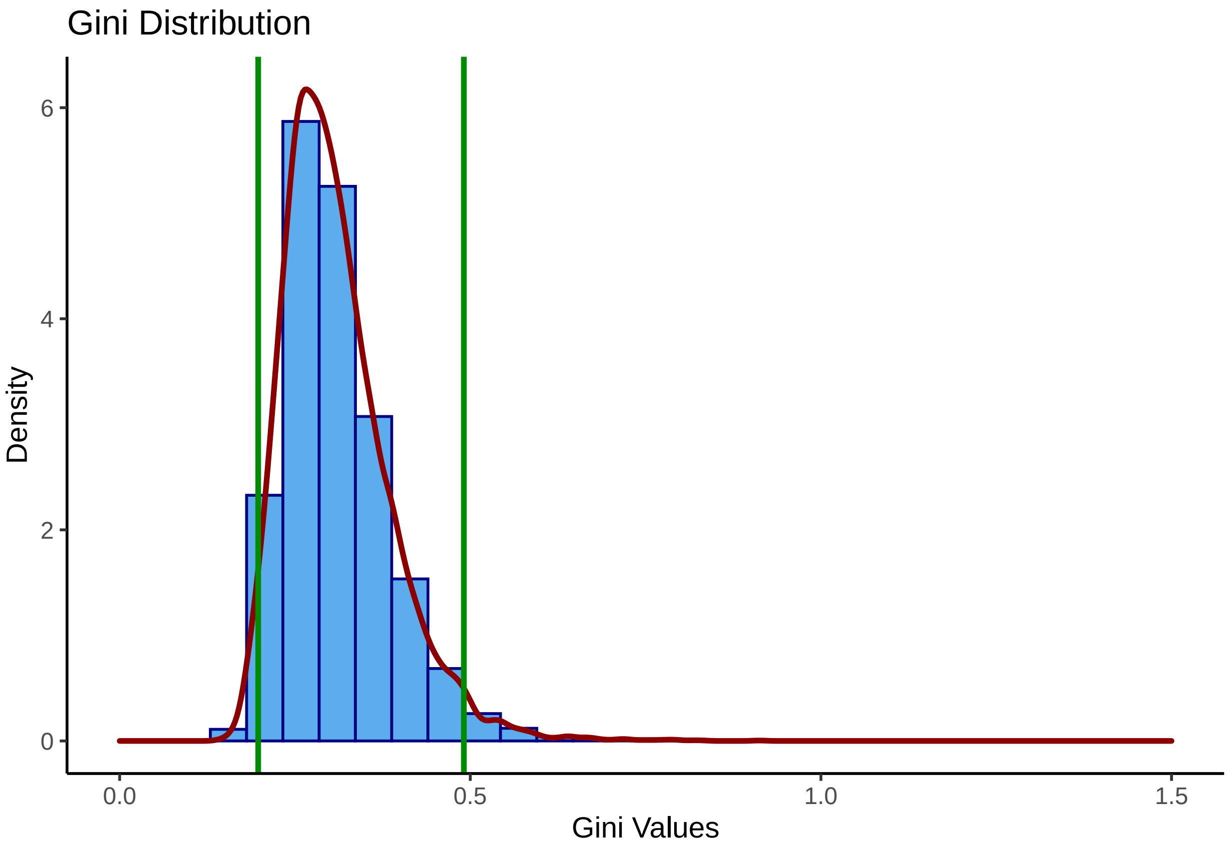
The above plot illustrates the posterior distribution of σ^2.

Task 2b



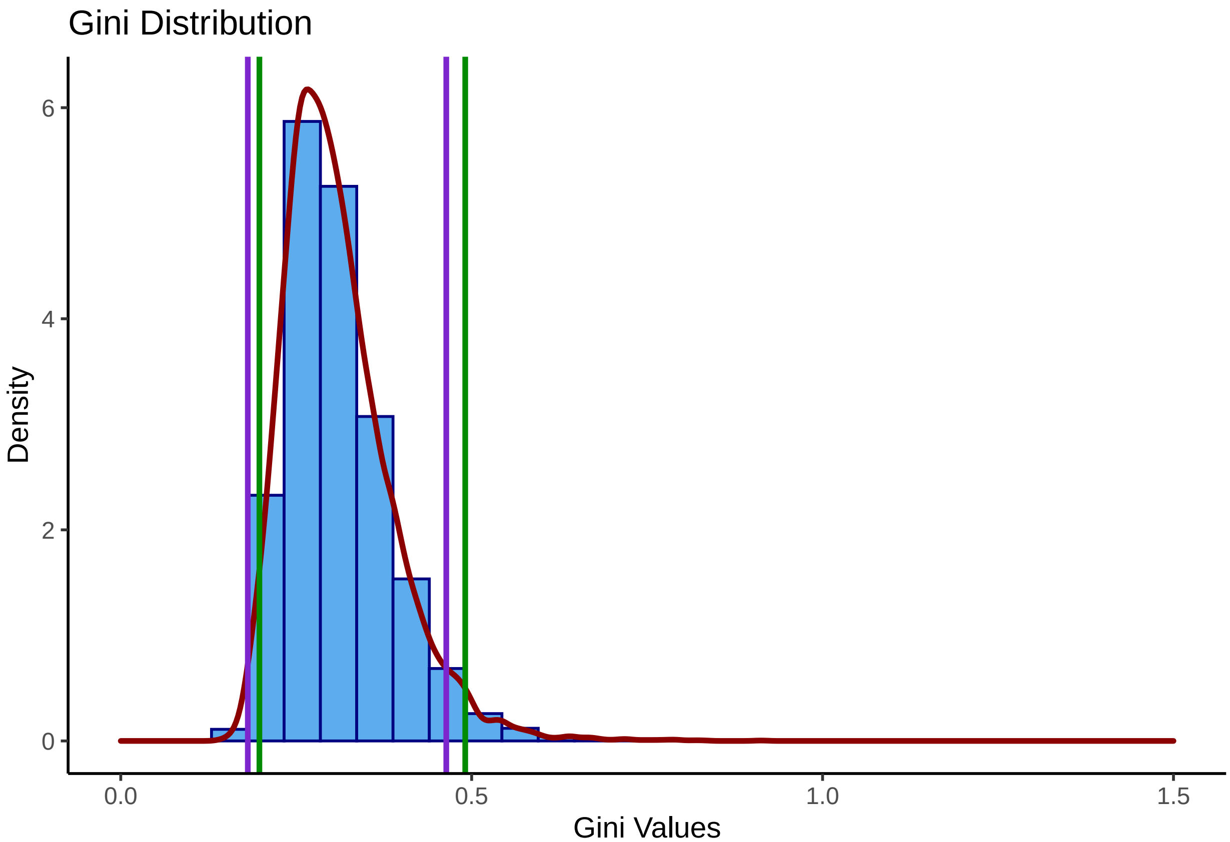
The above plot illustrates the Gini distribution.

Task 2c



The 95% equal tail credible interval for G is 0.1975510 for 2.5% and 0.4908131 for 97,5%.

Task 2d



Assignment 3 *Bayesian inference for the concentration parameter in the von Mises distribution*

Task 3a

The posterior is given by the below expression:

$$
\begin{aligned}
p(\kappa|y,\mu) &= \frac{p(y,\mu|\kappa)\cdot p(\kappa)}{\int_k p(y,\mu|\kappa)\cdot p(\kappa)d\kappa} \\
&\propto p(y,\mu|\kappa)\cdot p(\kappa)
\end{aligned}
$$

Thus, the likelihood $p(y,\mu|\kappa)$ and the prior $p(\kappa)$ need to be calculated.

The likelihood is given by the below formula:

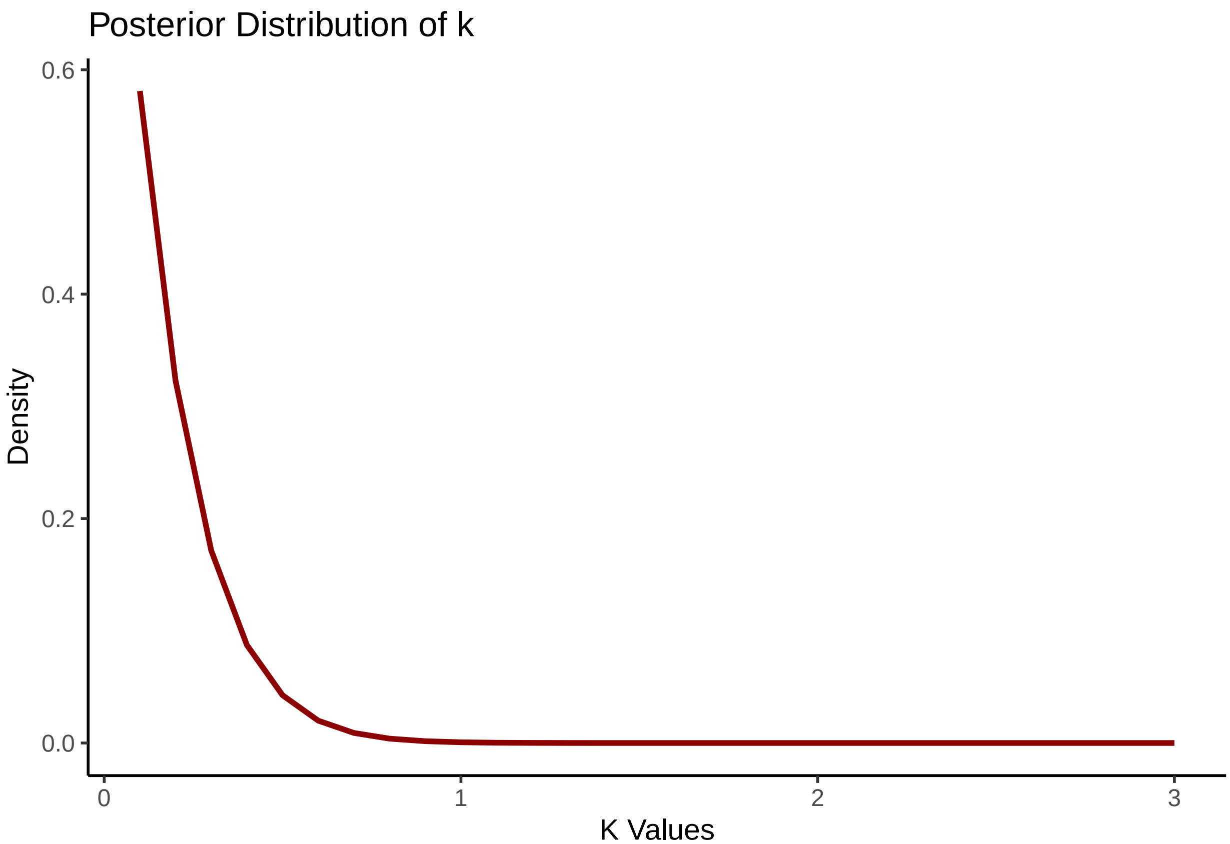
$$
\begin{aligned}
p(y,\mu|\kappa) &= \prod_{i=1}^{n} p(y|\mu,\kappa) \\
&= \prod_{i=1}^{n} \frac{\exp(\kappa\cdot\cos(y_i-\mu))}{2\pi\cdot I_o(\kappa)} \\
&= \Big(\frac{1}{2\pi\cdot I_o(\kappa)}\Big)^n \exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu)) \\
&= \frac{1}{(2\pi)^n}\cdot\frac{1}{I_o(\kappa)^n}\cdot\exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu)) \\
&\propto \frac{1}{I_o(\kappa)^n}\cdot\exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu))
\end{aligned}
$$

It is known that $\kappa \sim Exp(\lambda = 1)$; thus is only needed o calculate the probability density function of κ.

$$
\begin{aligned}
p(\kappa) &= \lambda\cdot\exp(-\lambda x) \\
&= \exp(-\kappa)
\end{aligned}
$$

The posterior is given by the below expression:

$$
\begin{aligned}
p(\kappa|y,\mu) &= \frac{1}{I_o(\kappa)^n}\cdot\exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu))\cdot\exp(-\kappa) \\
&= \frac{1}{I_o(\kappa)^n}\cdot\exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu)-\kappa)
\end{aligned}
$$


Posterior Distribution of k
0.6
0.4
0.2
0.0
Density
0
1
2
3
K Values

Task 3b

Appendix

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
# task 1a true mean, var & sd
a0 <- 5
b0 <- 5
n <- 50
s <- 13
f <- 50 - 13
mean_true <- (a0 + s)/(a0 + s + b0 + f)
var_true <- ((a0 + s)*(b0 + f)) / (((a0 + s + b0 + f)^2)*(a0 + s + b0 + f + 1))
sd_true <- sqrt(var_true)
set.seed(12345)

# calculate posterior's mean
mean_posterior = c()
for (i in 1:1000) {
# rbeta generates random deviates
  mean_posterior[i] = mean(rbeta(n = i, shape1 = a0 + s, shape2 = b0 + f))
}

# calculate posterior's sd
sd_posterior = c()
for (i in 1:1000) {
  sd_posterior[i] = sd(rbeta(n = i, shape1 = a0 + s, shape2 = b0 + f))
}

# df for plots
df_plot1 <- data.frame("draws" = 1:1000,
                       "mean_true" = mean_true,
                       "sd_true" = sd_true,
                       "mean_posterior" = mean_posterior,
                       "sd_posterior" = sd_posterior)

library(ggplot2)

# plot for mean values
ggplot(df_plot1) +
  geom_point(aes( x = draws, y = mean_posterior, color = "nany")) +
  geom_line(aes(x = draws, y = mean_true, color = "red4")) +
  theme(legend.position="right") +
  scale_color_manual(values=c('navy','red4'),
                     name = "Samples",
                     labels = c("Posterior's Mean Values","True Mean Values" )) +
  ggtitle("Mean Values Graphs") +
  xlab("Number of Draws") +
  ylab("Mean value of each sample") +
  theme_classic()
# plot for sd values
ggplot(df_plot1) +
  geom_point(aes( x = draws, y = sd_posterior, color = "nany")) +
  geom_line(aes(x = draws, y = sd_true, color = "red4")) +
```

```r
  theme(legend.position="right") +
  scale_color_manual(values=c('navy','red4'),
                     name = "Samples",
                     labels = c("Posterior's Sd Values","True Sd Values" )) +
  ggtitle("Standard Deviation Values Graphs") +
  xlab("Number of Draws") +
  ylab("Sd value of each sample") +
  theme_classic()
set.seed(12345)

#generates 1,000 random deviates.
posterior_sample <- rbeta(n = 1000, shape1 = a0+s, shape2 = b0+f)

#posterior probability
posterior_prob <- sum(posterior_sample < 0.3)/1000

#exact posterior prob
#pbeta the distribution function
exact_prob <- pbeta(q = 0.3, shape1 = a0+s, shape2 = b0+f)
phi <- log(posterior_sample/(1-posterior_sample))

df_plot2 <- data.frame("phi" =phi)

ggplot(df_plot2, aes(x=phi)) +
  geom_histogram(bins = 30, color = "navy", fill = "steelblue2", aes(y=..density..)) +
  geom_density(colour = "red4", size = 1) +
  ggtitle("Histogram of Log-Odds") +
  xlab("Log-Odds Values") +
  ylab("Density") +
  theme_classic()
#observations
obs <- c(33,24,48,32,55,74,23,76,17)

tau_2 <- sum((log(obs) - 3.5)^2)/9


#generates 10,000 random deviates.
set.seed(12345)
sigma2 <- c()
for (i in 1:10000){
  values<- rchisq(1,9)
  sigma2[i] <- 9*tau_2 / values
}

df_plot2 <- data.frame("sigma2" = sigma2)

ggplot(df_plot2, aes(x=sigma2)) +
  geom_histogram(bins = 30, color = "navy", fill = "steelblue2", aes(y=..density..)) +
  geom_density(colour = "red4", size = 1) +
  scale_x_continuous(limits = c(0,1.5)) +
  ggtitle("Posterior Distribution") +
  xlab("Sigma2 Values") +
  ylab("Density") +
```

```r
  theme_classic()
#Gini calculation
gini <- 2*pnorm(sqrt(sigma2)/sqrt(2)) - 1

df_plot3 <- data.frame("Gini" = gini)

ggplot(df_plot3, aes(x=Gini)) +
  geom_histogram(bins = 30, color = "navy", fill = "steelblue2", aes(y=..density..)) +
  geom_density(colour = "red4", size = 1) +
  scale_x_continuous(limits = c(0,1.5)) +
  ggtitle("Gini Distribution") +
  xlab("Gini Values") +
  ylab("Density") +
  theme_classic()
#producing sample quantiles corresponding to the probabilities
intervals <- quantile(gini, probs = c(0.025,0.975))

ggplot(df_plot3, aes(x=Gini)) +
  geom_histogram(bins = 30, color = "navy", fill = "steelblue2", aes(y=..density..)) +
  geom_density(colour = "red4", size = 1) +
  scale_x_continuous(limits = c(0,1.5)) +
  ggtitle("Gini Distribution") +
  xlab("Gini Values") +
  ylab("Density") +
  theme_classic() +
  geom_vline(aes(xintercept=intervals[1]), color = "green4", size = 1) +
  geom_vline(aes(xintercept=intervals[2]), color = "green4", size = 1)
#kernel density estimation
gini_density <- density(gini)

df_density <- data.frame(
  #the n coordinates of the points where the density is estimated
  "coord" = gini_density$x,
  #the estimated density values
  "estimated_vals" = gini_density$y)

#order/sort the estimated density values
df_density <- df_density[order(gini_density$y, decreasing = TRUE),]

library("HDInterval")
hpdi <- hdi(gini_density, 0.95)

ggplot(df_plot3, aes(x=Gini)) +
  geom_histogram(bins = 30, color = "navy", fill = "steelblue2", aes(y=..density..)) +
  geom_density(colour = "red4", size = 1) +
  scale_x_continuous(limits = c(0,1.5)) +
  ggtitle("Gini Distribution") +
  xlab("Gini Values") +
  ylab("Density") +
  theme_classic() +
  geom_vline(aes(xintercept=intervals[1]), color = "green4", size = 1) +
  geom_vline(aes(xintercept=intervals[2]), color = "green4", size = 1) +
  geom_vline(aes(xintercept=hpdi[1]), color = "purple3", size = 1) +
```

```r
  geom_vline(aes(xintercept=hpdi[2]), color = "purple3", size = 1)
degrees <- c(285, 296, 314, 20, 299, 296, 40, 303, 326, 308)
radians = c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

#normalasation
mean_rads = mean(radians)
sd_rads = sd(radians)
norm_radians = (radians-mean_rads)/sd_rads

k <- seq(0.1,3,0.1)

posterior <- exp(k*sum(cos(norm_radians-2.51))-k)/besselI(x = k, nu=0)^9

df_plot3 <- data.frame("k"=k, "posterior_vals"=posterior)

ggplot(df_plot3) +
  geom_line(aes(x=k, y=posterior_vals),colour = "red4", size = 1) +
  ggtitle("Posterior Distribution of k") +
  xlab("K Values") +
  ylab("Density") +
  theme_classic()
max_index <- which.max(posterior)
k_mode <- posterior[max_index]
```