Bayesian Learning (732A91) Lab1 Report

Christoforos Spyretos (chrsp415) & Marketos Damigos (marda352)

2022-04-18

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:

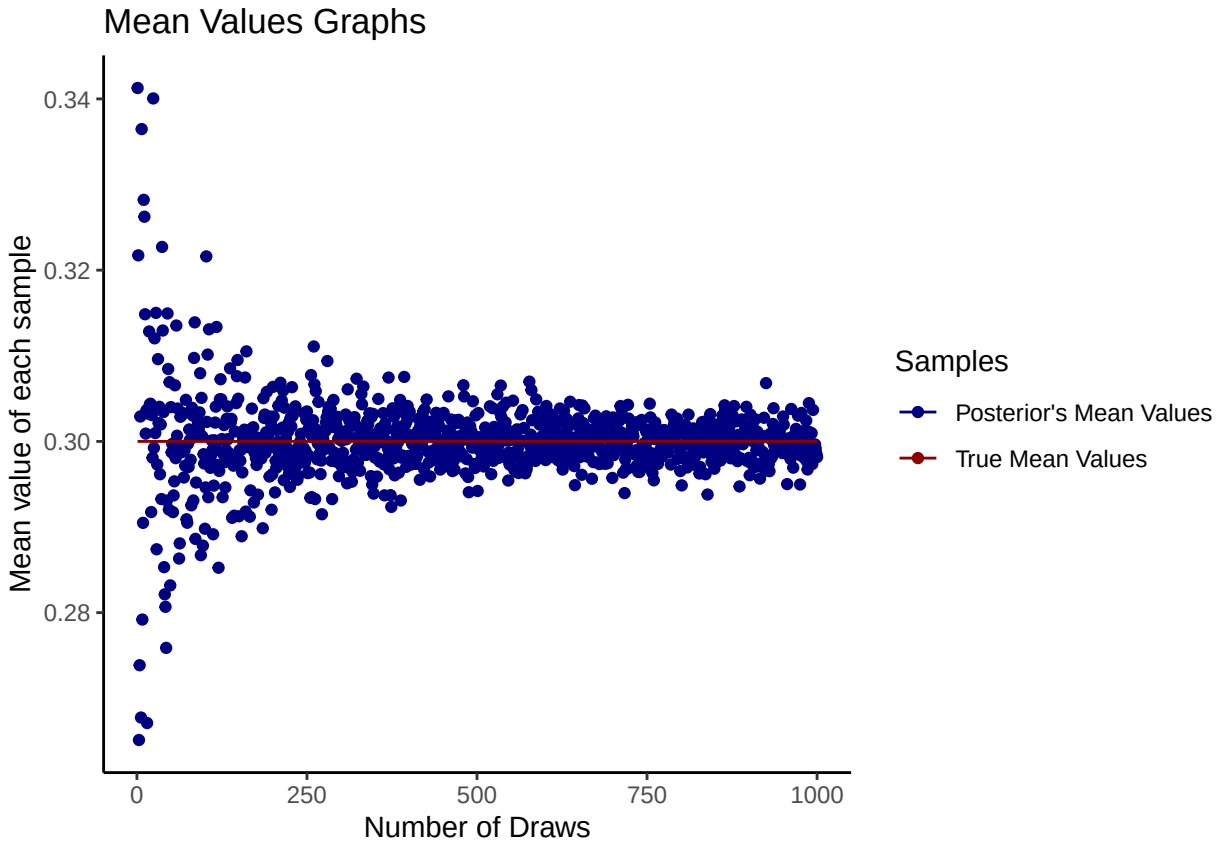
$$\begin{aligned} E[\theta] &= \frac{a_0 + s}{a_0 + s + b_0 + f} \\ &= \frac{18}{60} \\ &= 0.3 \end{aligned}$$

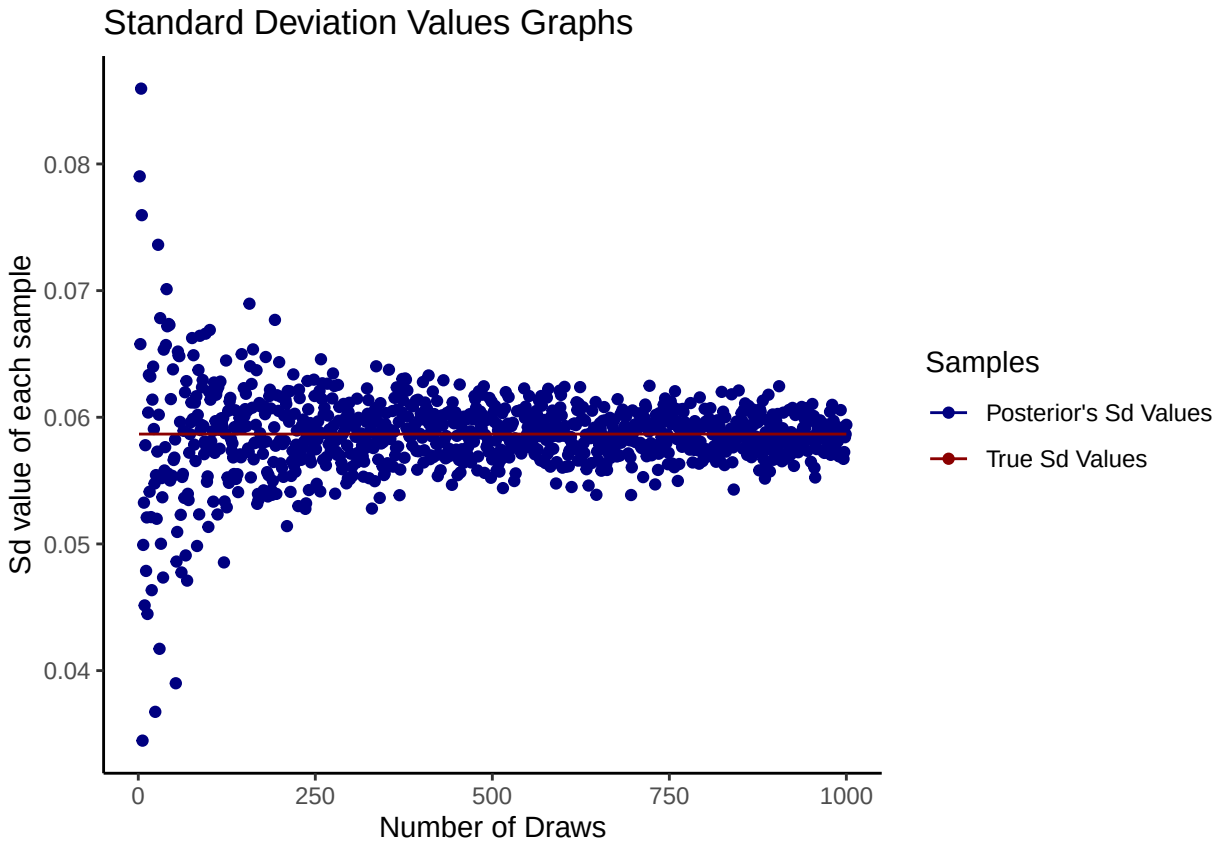
$$\begin{aligned} Var[\theta] &= \frac{(a_0 + s)(b_0 + f)}{\Big((a_0 + s) + (b_0 + f)\Big)^2 \Big((a_0 + s) + (b_0 + f) + 1\Big)} \\ &= \frac{18 \cdot 42}{(18 + 42)^2(18 + 42 + 1)} \\ &= 0.003442623 \end{aligned}$$

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

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.

In the next page, the graphs represent the mean value and the standard deviation of the Beta distribution for θ and the mean value $E[\theta|y]$ (red line) and standard deviation $SD[\theta|y]$ from the posterior $\theta|y \sim Beta(a_0 + s, b_0 + f)$ (blue points), where $y = y_1, y_2, ..., y_n$.


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

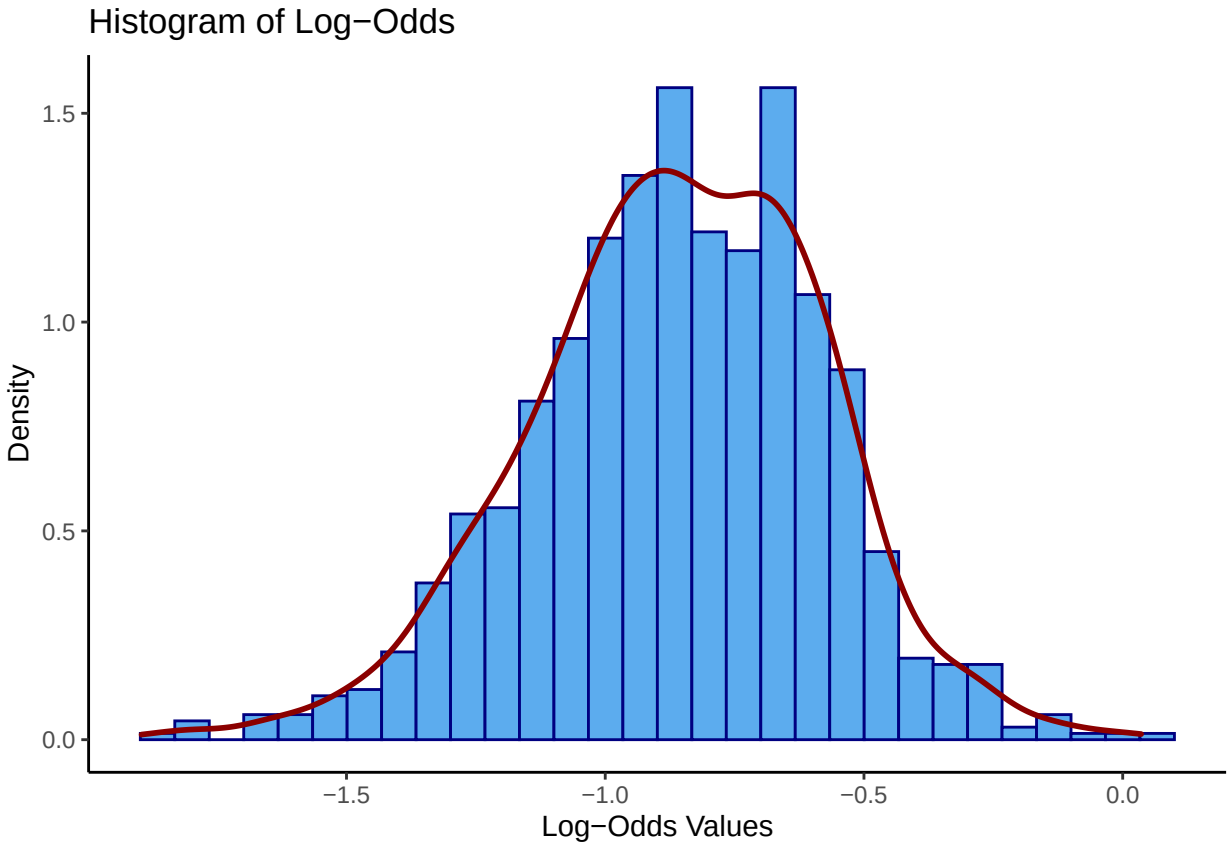

Standard Deviation Values Graphs
Sd value of each sample
0.08
0.07
0.06
0.05
0.04
0
250
500
750
1000
Number of Draws
Samples
Posterior's Sd Values
True Sd Values

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 more and more 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.

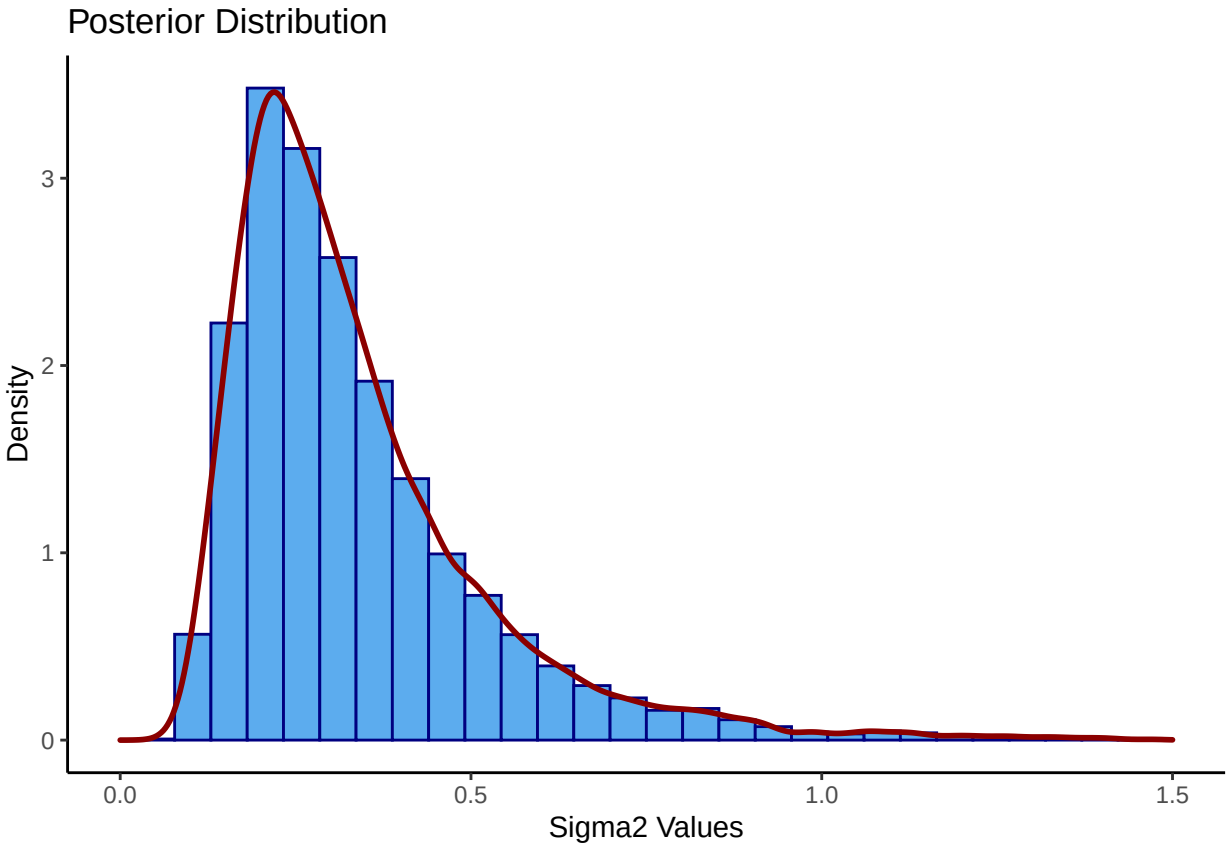
Task 1c



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

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

Task 2a



The above plot illustrates the posterior distribution of σ^2, where σ^2 is unknown with non-informative prior $p(\sigma^2) \propto \frac{1}{\sigma^2}$. The posterior for σ^2 is the $inv - \chi^2(n, \tau^2)$ distribution, $\sigma^2 = \frac{X}{\tau^2 \cdot n}$. The posterior of σ^2 is given by the expression $p(\sigma^2) = \frac{n \cdot \tau^2}{X}$, where $X \sim \chi^2(n)$, n is the number of observations and $\tau^2 = \frac{\sum_{i=1}^{\nu} log(y_i - \mu)^2}{n}$.

Task 2b

```
# Task 2b

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

Task 2c

	lower bound	upper bound
95% Equal Tail Credible Interval	0.197551	0.4908131

The above table illustrates the 95% equal tail credible interval for the Gini coefficient.

Task 2d

	lower_bound	upper_bound
95% Equal Tail Credible Interval	0.1975510	0.4908131
95% Highest Posterior Density Interval	0.1810286	0.4621952

The above table illustrates the 95% equal tail credible interval and the 95% highest posterior density interval for the Gini coefficient. It could be assumed that both intervals have almost similar lower and upper bounds.

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

Task 3a

The posterior is given by the below expression:

$$p(\kappa|y,\mu) = \frac{p(y,\mu|\kappa)\cdot p(\kappa)}{\int_k p(y,\mu|\kappa)\cdot p(\kappa)d\kappa}$$

The numerator is consisted of the likelihood $p(y,\mu|\kappa)$ and the prior $p(\kappa)$. The denominator is the marginal likelihood, which is the normalising constant, ensuring that the posterior distribution of κ adds up to one.

The likelihood is given by the below formula:

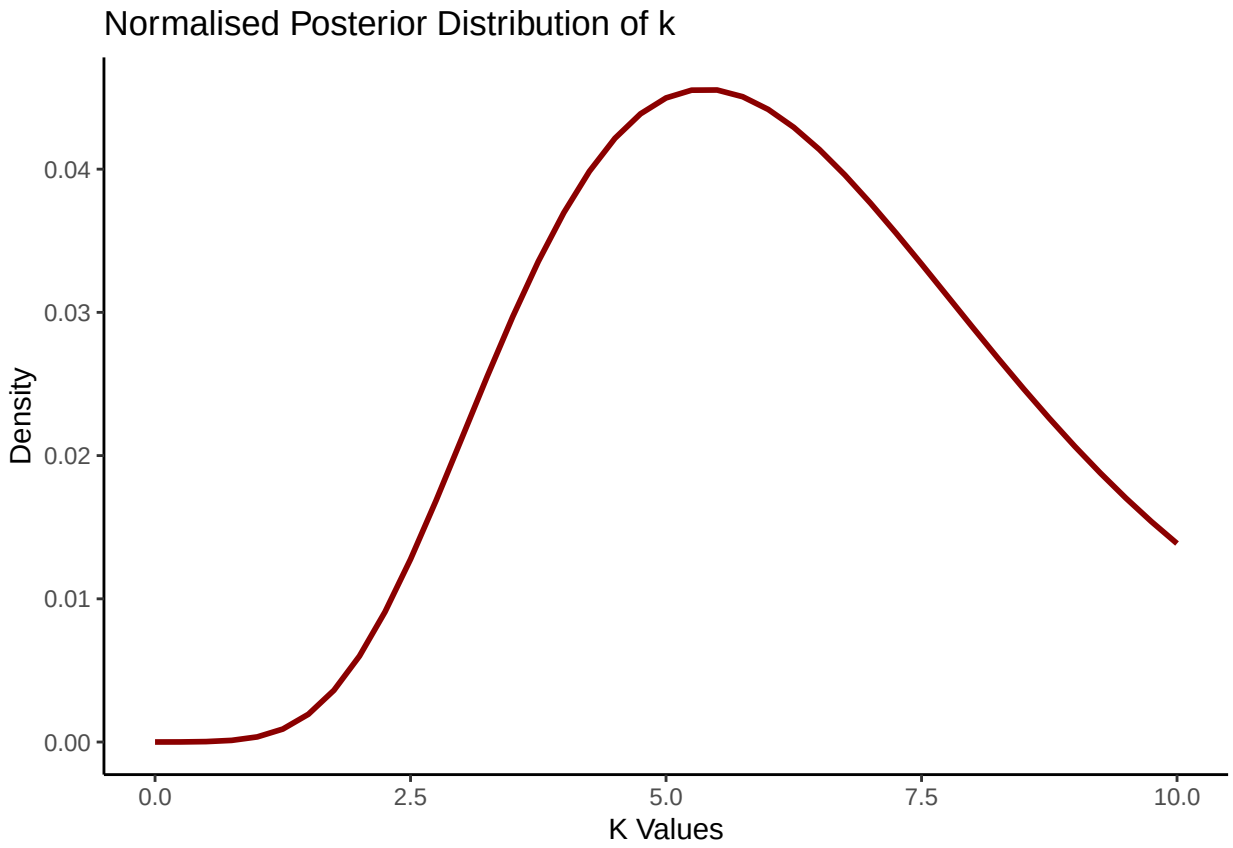
$$\begin{aligned}
p(y,\mu|\kappa) &= \prod_{i=1}^{n} p(y|\mu,\kappa) \\
&= \prod_{i=1}^{n} \frac{\exp(\kappa\cdot\cos(y_i-\mu))}{2\pi\cdot I_o(\kappa)} \\
&= \Big(\frac{1}{2\pi\cdot I_o(\kappa)}\Big)^n \exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu)) \\
&= \frac{1}{(2\pi)^n}\cdot\frac{1}{I_o(\kappa)^n}\cdot\exp(\kappa\cdot\sum_{i=1}^{n}\cos(y_i-\mu))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

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

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

Hence the numerator is given by the below expression:

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



The above density graph illustrates the normalised posterior distribution of κ for the wind direction data over a fine grid of κ values.

Task 3b

```
# Task 3b

# get the index of the max value
max_index <- which.max(norm_posterior)

# get the max value
k_mode <- norm_posterior[max_index]
```

The approximate posterior mode of κ equals 0.04552635

Appendix

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)

#_____________________________Assignment 1_____________________________#

#Task 1a

#parameters
a0 <- 5
b0 <- 5
n <- 50
s <- 13
f <- 50 - 13

#true mean
mean_true <- (a0 + s)/(a0 + s + b0 + f)
#true var
var_true <- ((a0 + s)*(b0 + f)) / (((a0 + s + b0 + f)^2)*(a0 + s + b0 + f + 1))
#true sd
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))
}

#data 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 of 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 of 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")) +
  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()

#Task 1b

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)

#Task 1c

#log-odds
phi <- log(posterior_sample/(1-posterior_sample))

#data for plot
df_plot2 <- data.frame("phi" = phi)

#plot of log-odds
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()

#_____________________________Assignment 2_____________________________#
```

```r
#Task 2a

#observations
obs <- c(33,24,48,32,55,74,23,76,17)

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


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

#data for plot
df_plot2 <- data.frame("sigma2" = sigma2)

#plot
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") +
  theme_classic()

#Task 2b

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

#Task 2c

#producing sample quantiles corresponding to the probabilities
interval <- quantile(gini, probs = c(0.025,0.975))

#table for the interval
df_intervals <- data.frame("lower_bound" = interval[1], "upper_bound" = interval[2])
colnames(df_intervals) <- c("lower bound", "upper bound")
rownames(df_intervals) <- c("95% Equal Tail Credible Interval")
knitr::kable(df_intervals)

#Task 2d

#kernel density estimation
gini_density <- density(gini)

df_density <- data.frame(
  #the n coordinates of the points where the density is estimated
```

```r
  "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),]

#calculate the probs
df_density$probs <- cumsum(df_density$estimated_vals)/sum(df_density$estimated_vals)

#get the indexes
index <- which(df_density$probs <= 0.95)

#get the probs
hdps <- df_density[index,]

#low hdpi
low_hdpi <- min(hdps$coord)
#upper hdpi
upp_hdpi <- max(hdps$coord)

intervals <- data.frame("lower_bound" = c(interval[1],low_hdpi), "upper_bound" = c(interval[2],upp_hdpi
rownames(intervals) <- c("95% Equal Tail Credible Interval","95% Highest Posterior Density Interval")

knitr::kable(intervals)
#_____________________________Assignment 3_____________________________#

#Task 3a

#data
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)

#kappa values
k <- seq(0,10,0.25)

#unormalised posterior
posterior <- exp(k*sum(cos(radians-2.51))-1)/(besselI(x = k, nu=0)^9)

#normalising constant/marginal likelihood
norm_constant <- sum(posterior)

#normalised posterior
norm_posterior <- posterior/norm_constant

#data for plotting
df_plot3 <- data.frame("k"=k, "posterior_vals"=norm_posterior)

#plot
ggplot(df_plot3) +
  geom_line(aes(x=k, y=posterior_vals),colour = "red4", size = 1) +
  ggtitle("Normalised Posterior Distribution of k") +
  xlab("K Values") +
```

```
  ylab("Density") +
  theme_classic()

#Task 3b

#get the index of the max value
max_index <- which.max(norm_posterior)

#get the max value
k_mode <- norm_posterior[max_index]
```