

# MS2505: Bayesian Statistics

## Course Project

December 22, 2024

<b>Name</b>	Samuel Jonsson
<b>E-Mail</b>	sajs19@student.bth.se
<b>Person Nr.</b>	19990415-5596
<b>Program</b>	DVAMI19h



# 1 Setup

## 1.1 Analysis problem

## 1.2 Data Selection

The dataset selected is a datasets containing a list of emails, as well as a label marking each email as “spam” or “ham” (spam or not spam). The first 10 rows of the dataset looks as follows:

Table 1

<b>mail_data.csv dataset first 10 rows</b>	
<b>Category</b>	<b>Message</b>
ham	“Go until jurong point, crazy.. Available only in bugis n great world la e buffet. . . Cine there got amore wat. . .”
ham	Ok lar. . . Joking wif u oni. . .
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question (std txt rate) T&C's apply 08452810075over18's
ham	U dun say so early ho. . . U c already then say. . .
ham	“Nah I don't think he goes to usf, he lives around here though”
spam	“FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv”
ham	Even my brother is not like to speak with me. They treat me like aids patent.
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
...	...

Then, using a Python script, the labels were converted to 1 if it was “spam” and 0 if it was “ham”, for easier analysis.

## 1.3 Model

The model chosen was a binomial likelihood model with a beta prior. As the goal is to analyse the probability of an email being spam, the fallout will be binary (either it is spam or it is not). Hence, a binomial likelihood, where I want to find the parameter  $\theta$  in a dataset of fixed size with a set number of “successes” and “fails” (spam and ham), is appropriate.

Additionally, as I do not have any prior knowledge in regards to this distribution, a non-informative prior is the most suited option, and as  $Beta(1, 1)$  is a common prior used with binomial likelihood functions, I chose it for this problem.

## 2 Results

Table 2

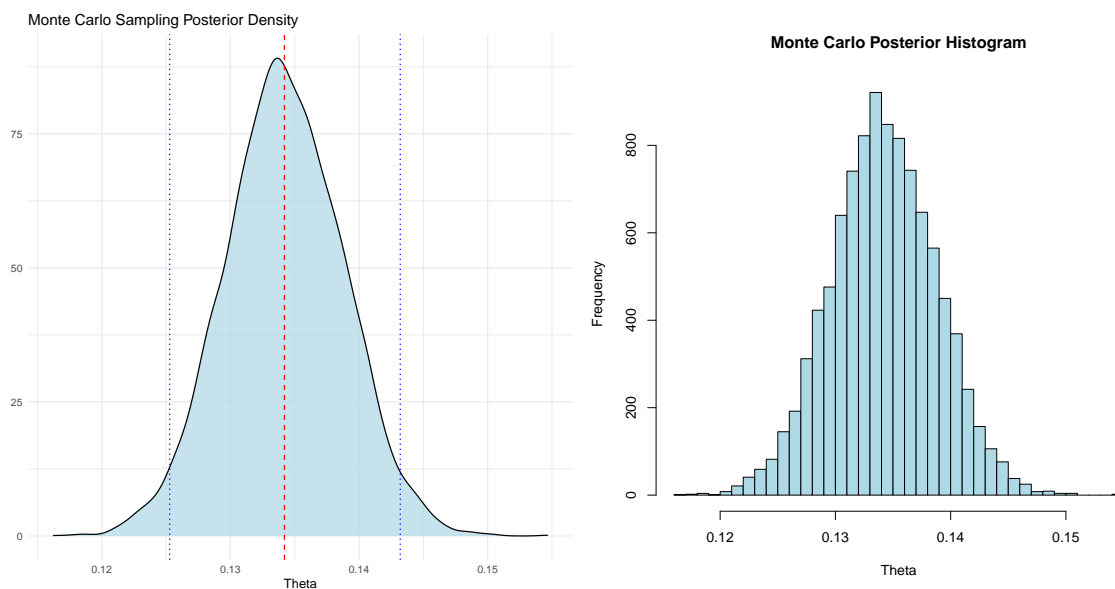
### Monte Carlo Posterior analysis data

<b>Posterior Mean</b>	0.1341994
<b>Posterior SD</b>	0.004576791
<b>95% Credible Interval</b>	[0.1252803, 0.1432051]

Table 3

### Markov Chain Monte Carlo Posterior analysis data

<b>Posterior Mean</b>	0.1344085
<b>Posterior SD</b>	0.004670508
<b>95% Credible Interval</b>	[0.1254781, 0.1437142]

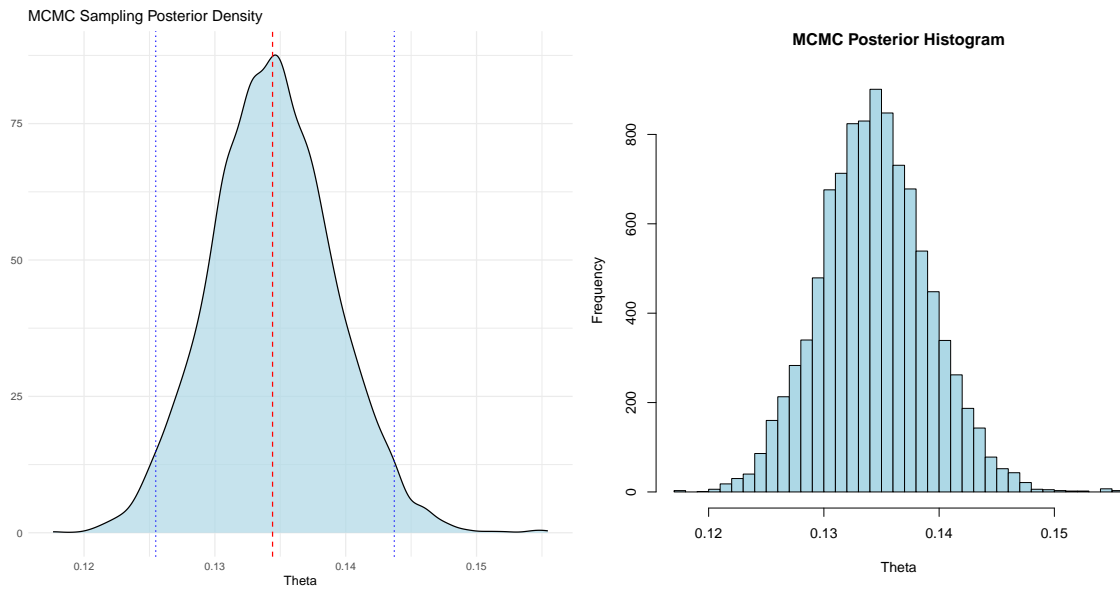


(a) Density plot

(b) Histogram

### Figure 1

Posterior density plot for monte carlo sampling. As can be seen in the plot, the estimated  $\theta$  is approximately 0.135, which corresponds to what was estimated in Table 2.

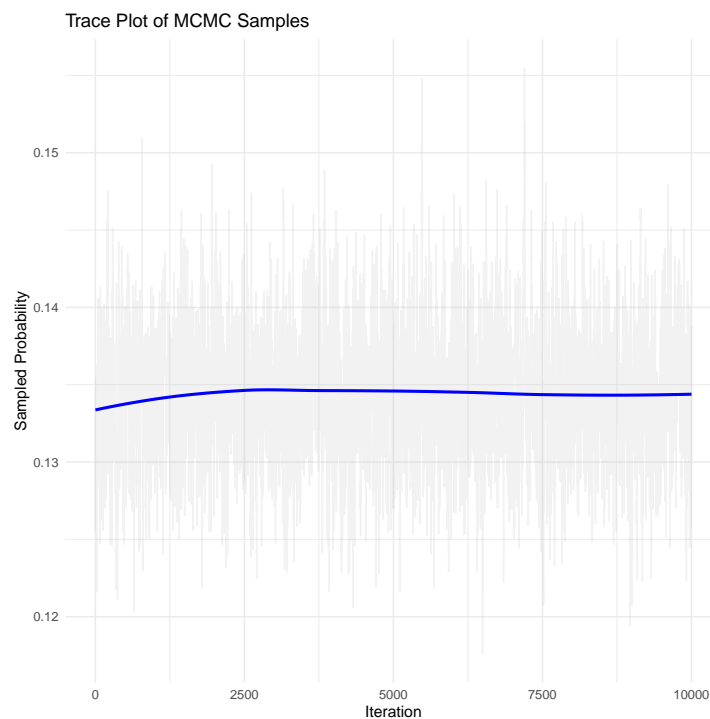


(a) MCMC density plot

(b) MCMC histogram

**Figure 2**

Posterior density plot for Markov Chain Monte Carlo sampling. Similarly to the results in Figure 1, the results show an expected value of 0.135, which follows what is estimated value in the corresponding Table 3

**Figure 3**

Markov Chain Monte Carlo sampling trace plot

## 3 Discussion

### A R Code

#### Listing 1

##### Project R code

```
1 # =====
2 # Bayesian Analysis for Email Spam Classification
3 # =====
4
5 # Load Required Libraries
6 required_packages <- c("ggplot2", "dplyr", "MCMCpack", "coda", "
  tidyr")
7 installed_packages <- rownames(installed.packages())
8 for (pkg in required_packages) {
9   if (!pkg %in% installed_packages) install.packages(pkg)
10  library(pkg, character.only = TRUE)
11 }
12
13 # Set a global random seed for reproducibility
14 set.seed(123)
15
16 # Output Logs
17 if (!dir.exists("logs")) dir.create("logs")
18 results_log_file <- "logs/combined_results.log"
19
20 # =====
21 # Load Data
22 # =====
23
24 data_path <- "data/mail_data_bin.csv"
25 if (!file.exists(data_path)) stop("Data file not found!")
26 mail_data <- read.csv(data_path)
27
28 # Ensure the response variable is binary and properly coded
29 if (!all(mail_data$Category %in% c(0, 1))) {
30   stop("The response variable 'Category' must be binary (0 or 1).")
31 }
32
33 # Metadata
34 spam_count <- sum(mail_data$Category == 1)
35 ham_count <- sum(mail_data$Category == 0)
36 total_emails <- nrow(mail_data)
37
38
39 sink(results_log_file)
40 cat("--- Data Metadata ---\n")
41 cat("Spam Count:", spam_count, "\n")
42 cat("Ham Count:", ham_count, "\n")
43 cat("Total Emails:", total_emails, "\n")
```

```
44 sink()
45
46 # =====
47 # Beta Posterior Analysis
48 # =====
49
50 # Prior parameters
51 prior_alpha <- 1
52 prior_beta <- 1
53
54 # Posterior parameters
55 posterior_alpha <- prior_alpha + spam_count
56 posterior_beta <- prior_beta + ham_count
57
58 # Monte Carlo Sampling
59 n_samples <- 10000
60 beta_samples <- rbeta(n_samples, posterior_alpha, posterior_beta)
61
62 # Summary statistics
63 beta_mean <- mean(beta_samples)
64 beta_sd <- sd(beta_samples)
65 beta_ci <- quantile(beta_samples, c(0.025, 0.975))
66
67 sink(results_log_file, append = TRUE)
68 cat("\n--- Beta Posterior Analysis ---\n")
69 cat("Posterior Mean:", beta_mean, "\n")
70 cat("Posterior SD:", beta_sd, "\n")
71 cat("95% Credible Interval:", beta_ci, "\n")
72 sink()
73
74 # =====
75 # MCMC Sampling
76 # =====
77
78 # Define log-posterior function
79 log_posterior <- function(params) {
80   theta <- params[1]
81   log_prior <- dbeta(theta, 1, 1, log = TRUE)
82   log_likelihood <- sum(dbinom(mail_data$Category,
83                               size = 1,
84                               prob = theta,
85                               log = TRUE))
86   return(log_prior + log_likelihood)
87 }
88
89 # Run MCMC sampling
90 mcmc_results <- MCMCmetrop1R(
91   fun = log_posterior,
92   theta.init = 0.5,
93   burnin = 1000,
94   mcmc = n_samples,
```

```
95   thin = 1,
96   verbose = 0
97 )
98
99 # Extract posterior samples
100 mcmc_samples <- as.vector(mcmc_results)
101 mcmc_mean <- mean(mcmc_samples)
102 mcmc_sd <- sd(mcmc_samples)
103 mcmc_ci <- quantile(mcmc_samples, c(0.025, 0.975))
104
105 sink(results_log_file, append = TRUE)
106 cat("\n--- MCMC Sampling Analysis ---\n")
107 cat("Posterior Mean:", mcmc_mean, "\n")
108 cat("Posterior SD:", mcmc_sd, "\n")
109 cat("95% Credible Interval:", mcmc_ci, "\n")
110 sink()
111
112 # =====
113 # Diagnostics and Visualization
114 # =====
115
116 if (!dir.exists("figures")) dir.create("figures")
117
118 # 1. Beta Density Plot
119 pdf("figures/beta_posterior_density_plot.pdf")
120 ggplot(data = data.frame(samples = beta_samples), aes(x = samples))
121   +
122   geom_density(fill = "lightblue", alpha = 0.7) +
123   geom_vline(xintercept = beta_mean, color = "red", linetype = "
     dashed") +
124   geom_vline(xintercept = beta_ci, color = "blue", linetype = "
     dotted") +
125   labs(title = "Monte Carlo Sampling Posterior Density", x = "Theta
     ", y = "") +
126   theme_minimal()
127 dev.off()
128
129 # 2. MCMC Density Plot
130 pdf("figures/mcmc_posterior_density_plot.pdf")
131 ggplot(data = data.frame(samples = mcmc_samples), aes(x = samples))
132   +
133   geom_density(fill = "lightblue", alpha = 0.7) +
134   geom_vline(xintercept = mcmc_mean, color = "red", linetype = "
     dashed") +
135   geom_vline(xintercept = mcmc_ci, color = "blue", linetype = "
     dotted") +
136   labs(title = "MCMC Sampling Posterior Density", x = "Theta", y =
     "") +
137   theme_minimal()
138 dev.off()
```

```
138 # 3. Beta Posterior Histogram
139 pdf("figures/beta_posterior_histogram.pdf")
140 hist(beta_samples,
141       breaks = 30, col = "lightblue", border = "black",
142       xlab = "Theta", main = "Monte Carlo Posterior Histogram"
143 )
144 dev.off()
145
146 # 4. MCMC Posterior Histogram
147 pdf("figures/mcmc_posterior_histogram.pdf")
148 hist(mcmc_samples,
149       breaks = 30, col = "lightblue", border = "black",
150       xlab = "Theta", main = "MCMC Posterior Histogram"
151 )
152 dev.off()
153
154 # 5. Trace Plot for MCMC
155 pdf("figures/mcmc_trace_plot.pdf")
156 ggplot(
157   data.frame(Iteration = 1:n_samples, Sample = mcmc_samples),
158   aes(x = Iteration, y = Sample)
159 ) +
160   geom_line(alpha = 0.2, color = "gray") +
161   geom_smooth(color = "blue", method = "loess", se = FALSE) +
162   labs(title = "Trace Plot of MCMC Samples",
163        x = "Iteration",
164        y = "Sampled Probability") +
165   theme_minimal()
166 dev.off()
167
168 # =====
169 # Completion
170 # =====
171
172 cat("Analysis complete. Check 'logs' and 'figures' directories for
    results.\n")
```