# Homework 3

Santiago Ruiz, Nikita Karetnikov, Felix Ubl

2025-06-02

## Exercise 2

Consider the use of Gibbs sampling to generate samples from a bivariate normal distribution. Let the means be 0, the variances be 1, and the correlation be .

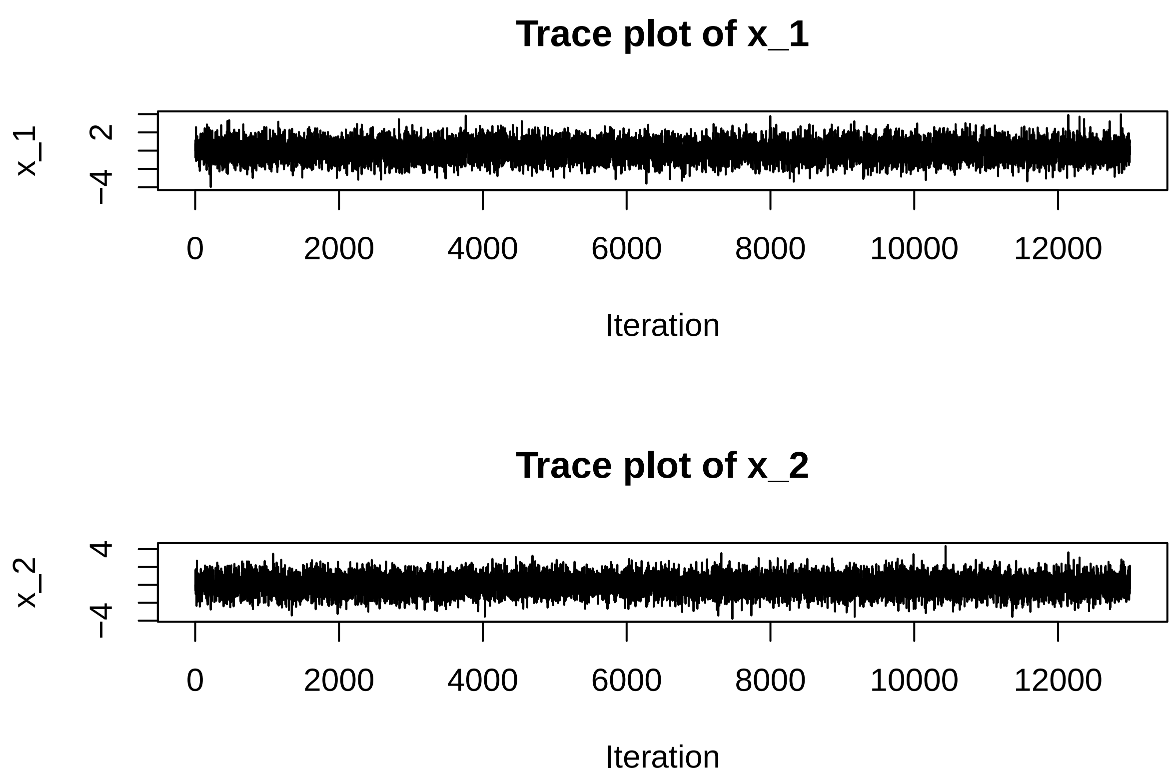
Answer: The Gibbs sampler yields more reliable results when the correlation is low, as the conditional distributions are less dependent and mix faster. Variance estimates are accurate across settings, and higher burn-in proportions improve the estimation of the correlation. The estimated means remain close to zero (error $< 0.02$), indicating good convergence. Overall, Gibbs sampling is effective for this problem due to the tractability of the conditional distributions.

```
library(phonTools)
```

```
## Warning: package 'phonTools' was built under R version 4.2.3
```

```
ps <- c(0.1, 0.2, 0.3, 0.4, 0.5)
burn_in_props <- c(0.3, 0.4, 0.5)
total_samples <- 13000
results <- data.frame(
    p = numeric(),
    burn_in_prop = numeric(),
    mean_x1 = numeric(),
    mean_x2 = numeric(),
    var_x1 = numeric(),
    var_x2 = numeric(),
    cov_x1x2 = numeric()
)

set.seed(123)

for (p in ps) {
    for (burn_in_prop in burn_in_props) {
        burn_in <- as.integer(total_samples * burn_in_prop)
        keep_draws <- total_samples - burn_in
        mu <- c(0, 0)
        sigma <- matrix(c(1, p, p, 1), nrow = 2)
        x_1 <- rep(0, total_samples)
        x_2 <- rep(0, total_samples)
        for (i in 2:total_samples) {
            x_2[i] <- sqrt(1 - p^2) * rnorm(1) + p * x_1[i - 1]
            x_1[i] <- sqrt(1 - p^2) * rnorm(1) + p * x_2[i]
        }
        ygs <- cbind(x_1, x_2)
        ygs <- ygs[(burn_in + 1):total_samples, ]
```

```r
        m <- colMeans(ygs)
        v <- cov(ygs)
        results <- rbind(
            results,
            data.frame(
                p = p,
                burn_in_prop = burn_in_prop,
                mean_x1 = m[1],
                mean_x2 = m[2],
                var_x1 = v[1, 1],
                var_x2 = v[2, 2],
                cov_x1x2 = v[1, 2]
            )
        )
    }
}
        par(mfrow = c(2, 1))
        plot(x_1, type = "l", main = "Trace plot of x_1", xlab = "Iteration", ylab = "x_1")
        plot(x_2, type = "l", main = "Trace plot of x_2", xlab = "Iteration", ylab = "x_2")
```




```r
        par(mfrow = c(1, 1))

knitr::kable(results, digits = 4, caption = "Comparison of Gibbs Sampling Results for Different p and Bu
```

Table 1: Comparison of Gibbs Sampling Results for Different p and Burn-in Proportions

| | p | burn_in_prop | mean_x1 | mean_x2 | var_x1 | var_x2 | cov_x1x2 |
|---|---|---|---|---|---|---|---|
| x_1 | 0.1 | 0.3 | -0.0108 | -0.0136 | 1.0047 | 1.0180 | 0.1130 |
| x_11 | 0.1 | 0.4 | 0.0126 | 0.0117 | 1.0303 | 0.9869 | 0.0880 |
| x_12 | 0.1 | 0.5 | 0.0116 | 0.0014 | 0.9805 | 0.9915 | 0.0990 |
| x_13 | 0.2 | 0.3 | -0.0046 | -0.0091 | 0.9884 | 0.9794 | 0.1876 |
| x_14 | 0.2 | 0.4 | -0.0024 | 0.0001 | 0.9873 | 0.9996 | 0.1914 |
| x_15 | 0.2 | 0.5 | -0.0107 | 0.0152 | 1.0111 | 1.0065 | 0.2094 |
| x_16 | 0.3 | 0.3 | 0.0115 | 0.0099 | 1.0023 | 1.0099 | 0.3171 |
| x_17 | 0.3 | 0.4 | 0.0137 | 0.0113 | 1.0127 | 1.0133 | 0.3139 |
| x_18 | 0.3 | 0.5 | -0.0137 | -0.0031 | 1.0083 | 0.9824 | 0.3099 |
| x_19 | 0.4 | 0.3 | -0.0211 | -0.0126 | 1.0027 | 1.0170 | 0.4091 |
| x_110 | 0.4 | 0.4 | 0.0165 | 0.0042 | 1.0097 | 1.0069 | 0.4127 |
| x_111 | 0.4 | 0.5 | -0.0188 | -0.0114 | 1.0044 | 1.0291 | 0.4009 |
| x_112 | 0.5 | 0.3 | -0.0024 | -0.0183 | 1.0003 | 1.0005 | 0.5038 |
| x_113 | 0.5 | 0.4 | 0.0143 | 0.0070 | 1.0263 | 1.0252 | 0.5155 |
| x_114 | 0.5 | 0.5 | -0.0230 | -0.0268 | 0.9697 | 1.0040 | 0.4794 |

## Exercise 3

It is assumed that the lifetime of light bulbs follows an exponential distribution with parameter $\lambda$. To estimate $\lambda$, n light bulbs were tested until they all failed. Their failure times were recorded as $u_1, ..., u_n$. In a separate experiment, m bulbs were tested, but the individual failure times were not recorded. Only the number of bulbs, r, that had failed at time t was recorded. The missing data are the failure times of the bulbs in the second experiment, $v_1, ..., v_m$.

Let us define the auxiliary variable: $v_1, \dots, v_m \sim \text{Exponential}(\lambda)$: partially observed, with only $r$ failures before time $t$; $r$ are right-censored at time $t$.

We assume a conjugate Gamma prior:

$$\lambda \sim \text{Gamma}(\alpha, \beta), \quad \text{with density } p(\lambda) \propto \lambda^{\alpha-1} e^{-\beta\lambda}$$

The exponential likelihood for fully observed data is:

$$L_{\text{obs}}(\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda u_i} = \lambda^n e^{-\lambda \sum_{i=1}^{n} u_i}$$

The observed failures in the second group (before time $t$) contribute:

$$L_{\text{fail}}(\lambda) = \prod_{j=1}^{r} \lambda e^{-\lambda v_j} = \lambda^r e^{-\lambda \sum_{j=1}^{r} v_j}$$

$$L_{\text{cens}}(\lambda) = \prod_{k=1}^{m-r} P(v_k > t \mid \lambda) = \left(e^{-\lambda t}\right)^{m-r} = e^{-\lambda t(m-r)}$$

Combining all:

$$L(\lambda \mid \mathbf{u}, \mathbf{v}) = \lambda^{n+r} e^{-\lambda(\sum u_i + \sum v_j + t(m-r))}$$

Multiplying by the Gamma prior:

$$p(\lambda \mid \text{data}) \propto L(\lambda) \cdot p(\lambda)$$

$$\propto \lambda^{n+r} e^{-\lambda(\sum u_i + \sum v_j + t(m-r))} \cdot \lambda^{\alpha-1} e^{-\beta\lambda}$$
$$= \lambda^{\alpha+n+r-1} e^{-\lambda(\beta + \sum u_i + \sum v_j + t(m-r))}$$

This is the kernel of a Gamma distribution:

$$\lambda \mid \text{data} \sim \text{Gamma}\left(\alpha + n + r,\ \beta + \sum u_i + \sum v_j + t(m-r)\right)$$

### Step 1: Sample $\lambda$

Given $\mathbf{u}, \mathbf{v}_{1:r}, \mathbf{v}_{r+1:m}$, the conditional posterior is:

$$\lambda \mid \mathbf{u}, \mathbf{v} \sim \text{Gamma}\left(\alpha + n + m,\ \beta + \sum_{i=1}^{n} u_i + \sum_{j=1}^{m} v_j\right)$$

(Note: this includes both observed and imputed $v_j$'s)

### Step 2: Sample each missing $v_j$, for $j = r+1, \dots, m$

Each right-censored $v_j \sim \text{Exponential}(\lambda)$, truncated to the interval $(t, \infty)$:

$$v_j \mid \lambda \sim \text{TruncatedExponential}(\lambda; v_j > t)$$

This can be sampled via (This is because the exponential distribution is memoryless):

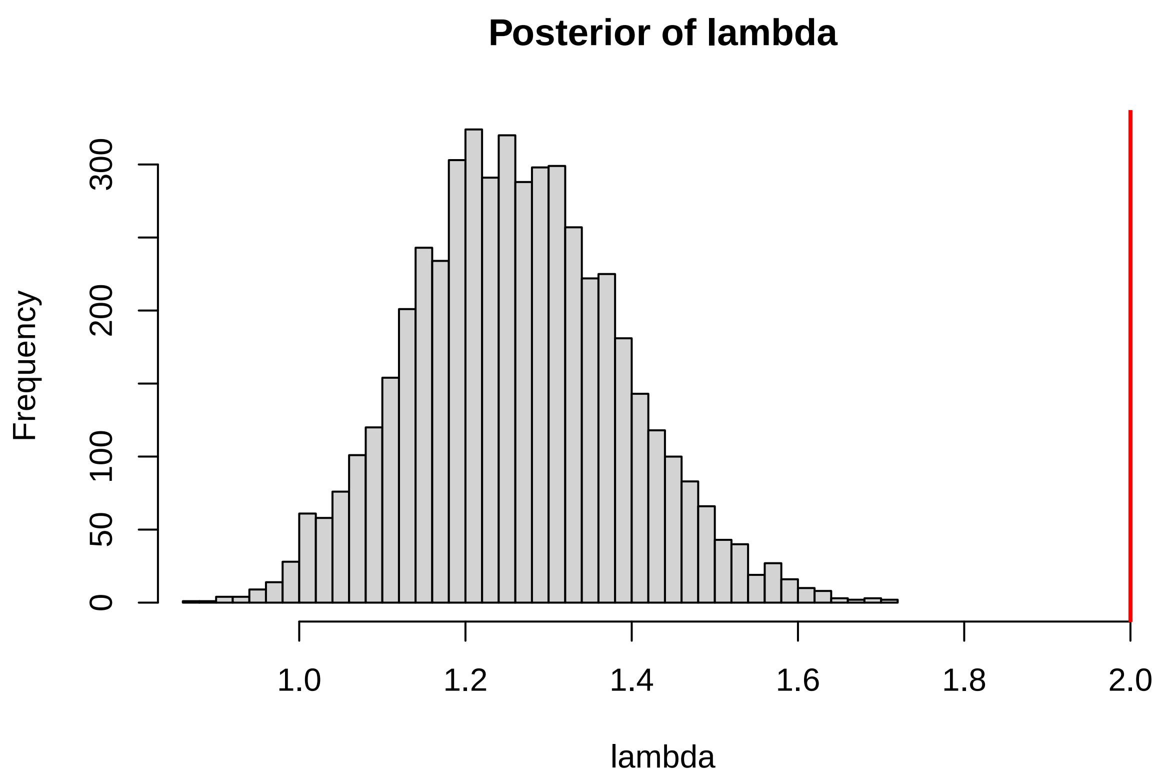
or equivalently:

$$v_j = t + \text{Exponential}(\lambda)$$

For sampling the truncated exponential, and the values from $u_i$ we don't need to use the Gibbs sampler, as the distribution is known and can be sampled directly.

```r
# Data generation
set.seed(123)
n <- 100; m <- 20; t <- 1
lambda_true <- 2

u <- rexp(n, rate = lambda_true)
v <- rexp(m, rate = lambda_true)
r <- sum(v <= t)

# Censored data, meaning those that did not fail before time t, but that we did not observe until they
v_obs <- v[v <= t]
v_censored_count <- r

# Gibbs parameters
alpha <- 1
beta <- 1
n_iter <- 10000
burn_in <- 5000

# Initialization
lambda <- 1
lambda_samples <- numeric(n_iter)
```

```r
# Step 0 : Initial imputation. It is greater than t, because if they had been shorted
# failure times, we would have include them into the v_obs vector.
v_missing <- rep(t + 0.1, v_censored_count)

for (iter in 1:n_iter) {

# Step 1: Sample missing failure times (truncated exponential)

  v_missing <- t + rexp(v_censored_count, rate = lambda)

  # Step 2: Update lambda from Gamma posterior
  total_time <- sum(u) + sum(v_obs) + sum(v_missing)*t
  lambda <- rgamma(1, shape = alpha + n + m, rate = beta + total_time)

  lambda_samples[iter] <- lambda
}

# Discard burn-in
lambda_samples_post <- lambda_samples[(burn_in + 1):n_iter]

# Results
hist(lambda_samples_post, breaks = 30, main = "Posterior of lambda", xlab = "lambda", xlim = range(c(lar
abline(v = lambda_true, col = "red", lwd = 2)
```




As it is possible to observe in the histogram, the posterior distribution of $\lambda$ is not centered around the true value. It seems that the MCMC algorithm is not converging properly. This could be due to the fact that the initial values of the missing data are not representative of the true distribution, or that the number of iterations is not sufficient for convergence. Another possibility is that the prior distribution is not informative enough, leading to a posterior distribution that is not centered around the true value. We also consider that the