QUESTION 2

The Box-Muller method generates standard normal random variables from two independent uniform random variables, $U \sim U(0,1)$ and $V \sim U(0,1)$, using the transformations

$$X1 = \sqrt{-2log(U)} \cdot sin(2\pi V)$$
$$X2 = \sqrt{-2log(U)} \cdot cos(2\pi V)$$

These transformations produce two independent standard normal variables, which can then be scaled and shifted to obtain the desired bivariate normal distribution with mean $\mu = (0,0)$ and covariance matrix $\Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$.

We implemented this method in R and measured the computational time for generating 10000000 random vectors.

To compare efficiency, we also generated the same number of bivariate normal samples using the `rmvnorm` function from the `mvtnorm` package, which is optimized for multivariate normal sampling.

The computation times for both methods are summarized in the table below:

Table 1: Computation time for each method

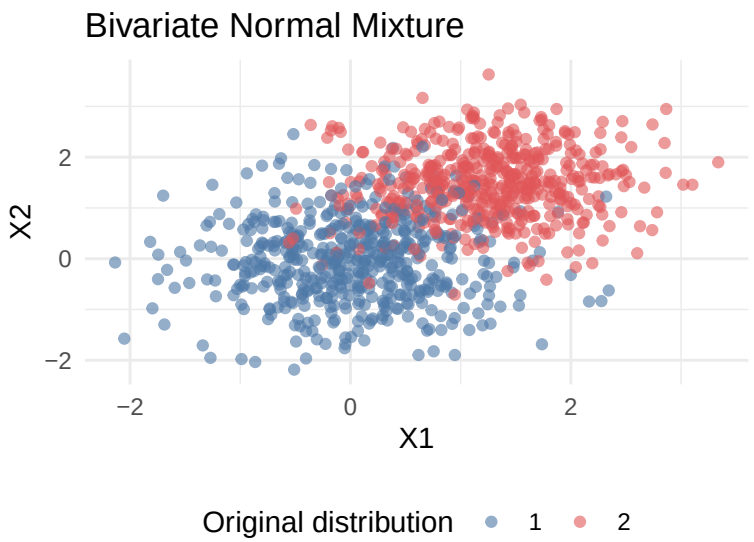
	Computation time (seconds)
Box-Muller method	2.49
rmvnorm function	1.65

As observed, the `rmvnorm` function is significantly faster than the Box-Muller method. The Box-Muller method requires two uniform random variables per standard normal sample, making it computationally expensive. The `rmvnorm` function, on the other hand, utilizes more efficient matrix decomposition techniques, leading to improved performance.

Now, we consider a bivariate normal mixture model where each observation follows one of two distributions with equal probability (50% each):

$$1.\ N(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix})$$
$$2.\ N(\mu = \begin{bmatrix} 1.5 \\ 1.2 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix})$$

We generated 1000 random samples from this mixture distribution, assigning each sample to one of the two components based on a random Bernoulli draw. The resulting scatter plot confirms the expected behavior: the data points are drawn from two distinct clusters corresponding to the two normal distributions.



The visualization effectively shows the structure of the mixture, showing two overlapping normal distributions. This confirms that the mixture model is correctly implemented.

Appendix

Question 2

```r
library(ggplot2)
library(mvtnorm)
library(knitr)

# define n, mu and sigma
n <- 10000000
mu <- c(0, 0)
sigma <- matrix(c(0.6, 0, 0, 0.6), nrow = 2)

# Box Muller method
box_muller_normal <- function(n, mu, sigma){

  # generate U, V as random Uniform
  set.seed(1234)
  u <- runif(n)
  v <- runif(n)

  # Transform to X1 and X2 with the given formula
  x1 <- sqrt(-2*log(u))*cos(2*pi*v)*sqrt(sigma[1, 1]) + mu[1]
  x2 <- sqrt(-2*log(u))*sin(2*pi*v)*sqrt(sigma[2, 2]) + mu[2]

  return(data.frame(x1, x2))
}

# calculate the time to generate n random vectors Box-Muller method
time_box <- system.time({
  dist <- box_muller_normal(n, mu, sigma)
```

```r
})

# calculate the time to generate n random vectors with the rmvnorm function
set.seed(1234)
time_rmvnorm <- system.time({
  dist2 <- rmvnorm(n, mean = mu, sigma = sigma)
})

# results table
df <- data.frame("Time" = c(time_box["elapsed"], time_rmvnorm["elapsed"]))
rownames(df) <- c("Box-Muller method", "rmvnorm function")
colnames(df) <- c("Time (seconds)")
kable(df, caption = "Computation time for each method")

# set initial distribution paraemters
mu1 <- c(0, 0)
mu2 <- c(1.2, 1.5)
mu <- list(mu1, mu2)

sigma1 <- matrix(c(0.6, 0, 0, 0.6), nrow = 2)
sigma2 <- matrix(c(0.5, 0, 0, 0.5), nrow = 2)
sigma <- list(sigma1, sigma2)

# mixture probabilities
prob <- c(0.5, 0.5)

# generate 1000 random vectors from the mixture distribution
n <- 1000
x <- data.frame(matrix(NA, nrow = n, ncol = 3))
set.seed(1234)
for(i in 1:n){
  g <- sample(1:2, 1, replace = TRUE, p = prob)
  x[i, 1:2] <- rmvnorm(n = 1, mean = mu[[g]], sigma = sigma[[g]])
  x[i, 3] <- as.character(g)
}

# plot the mixture distribution
ggplot(x, aes(x = x[, 1], y = x[, 2], color = x[, 3])) +
  geom_point(alpha = 0.6) +
  theme_minimal() +
  scale_color_manual(values = c("#4E79A7", "#E15759")) +
  labs(title = "Bivariate Normal Mixture", x = "X1", y = "X2", color = "Original distribution") +
  theme(legend.position="bottom")
```