University of Edinburgh, School of Mathematics

Incomplete Data Analysis, 2020/2021

Bivariate normal data: one variable subject to missingness

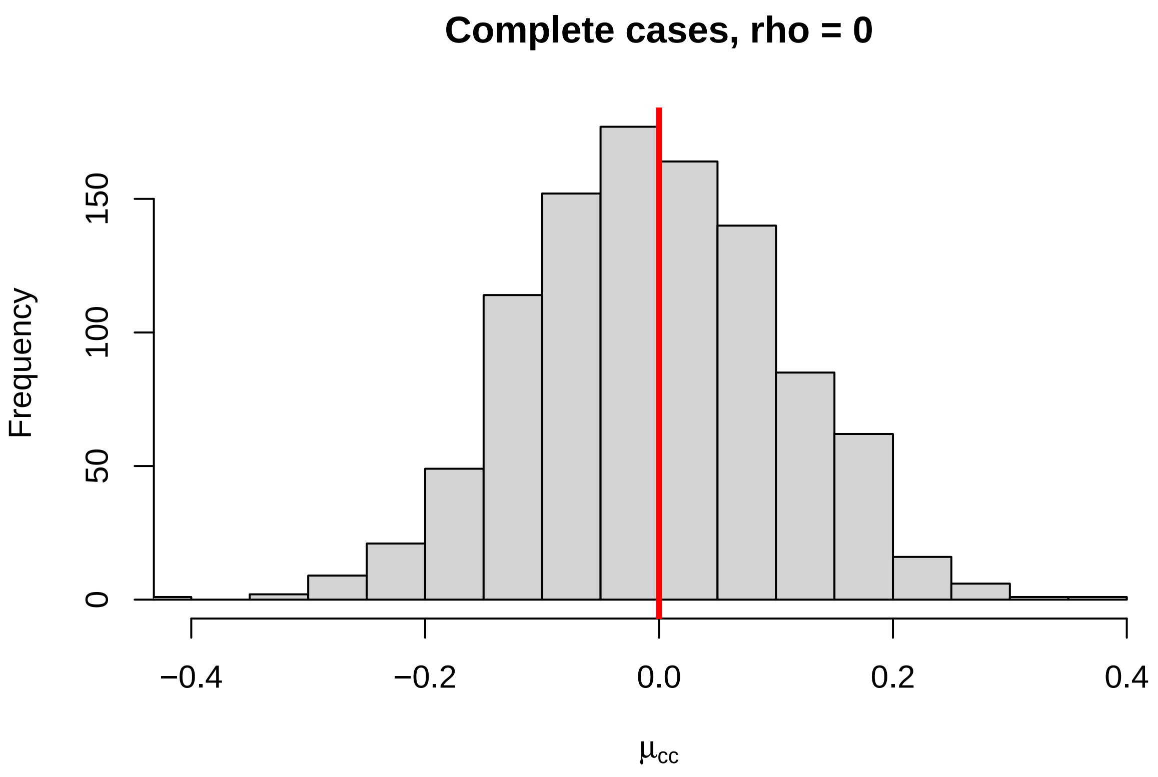
Vanda Inácio

We start by simulating 1000 data sets of size $n = 100$ from a bivariate normal distribution with the following structure

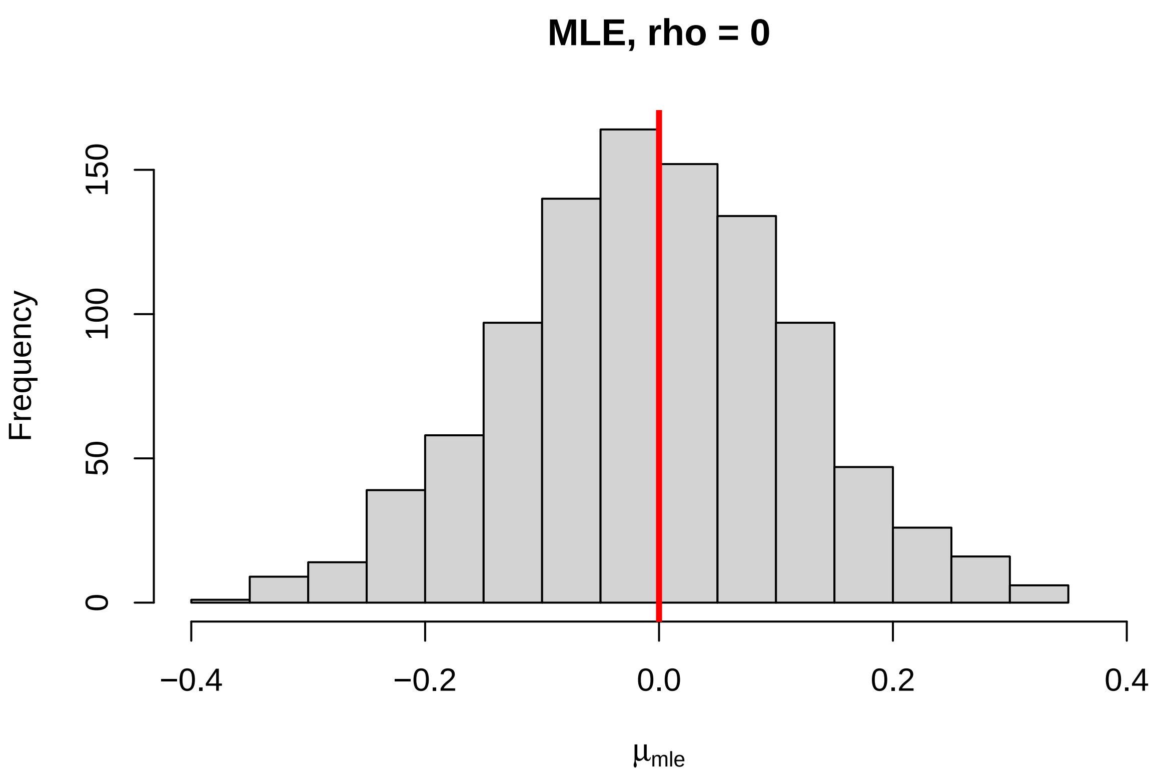
$$\boldsymbol{\mu} = (0, 0)', \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

where $\rho \in \{0, 0.5, 0.9\}$. Note that when $\rho = 0$, Y_1 and Y_2 are independent. I then created a function that takes as input the 1000 simulated bivariate data sets and the value of a used to induce missingness. I will start with the case of $\rho = 0$ and I have considered $a = 0.8$. Code follows.

```r
require(MASS)

rho <- 0
n <- 100
nsim <- 1000
mu <- c(0,0)
sigma1 <- 1
sigma2 <- 1
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))

set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

bivnorm <- function(y, a){
  n <- dim(y)[1]
  nsim <- dim(y)[3]

  y1 <- y2 <- r <- matrix(0, nrow = n, ncol = nsim)
  mu2mle <- mu2cc <- numeric(nsim)
  treshold <- qnorm(a, 0, 1)

  for(l in 1:nsim){

    y1[, l] <- y[, 1, l]
    y2[, l] <- y[, 2, l]
    r[,l] <- ifelse(y1[, l] > treshold, 0, 1)
    s1squared <- mean((y1[r[, l] == 1, l] - mean(y1[r[, l] == 1, l]))^2)
    s12 <- mean((y1[r[, l] == 1, l] -
                   mean(y1[r[, l] == 1, l]))*(y2[r[, l] == 1, l] - mean(y2[r[, l] == 1, l])))
    beta1 <- s12/s1squared
    beta0 <- mean(y2[r[,l] == 1, l]) - beta1*mean(y1[r[,l] == 1, l])
    mu1 <- mean(y1[,l])

    mu2mle[l] <- beta0 + beta1*mu1
    mu2cc[l] <- mean(y2[r[, l] == 1, l])
  }
  return(list(mu2mle, mu2cc))
}

res_rho0 <- bivnorm(y = y, a = 0.8)

hist(res_rho0[[2]], main = "Complete cases, rho = 0", xlab = expression(mu[cc]), xlim = c(-0.4, 0.4))
abline(v = 0, col = "red", lwd = 3)
```

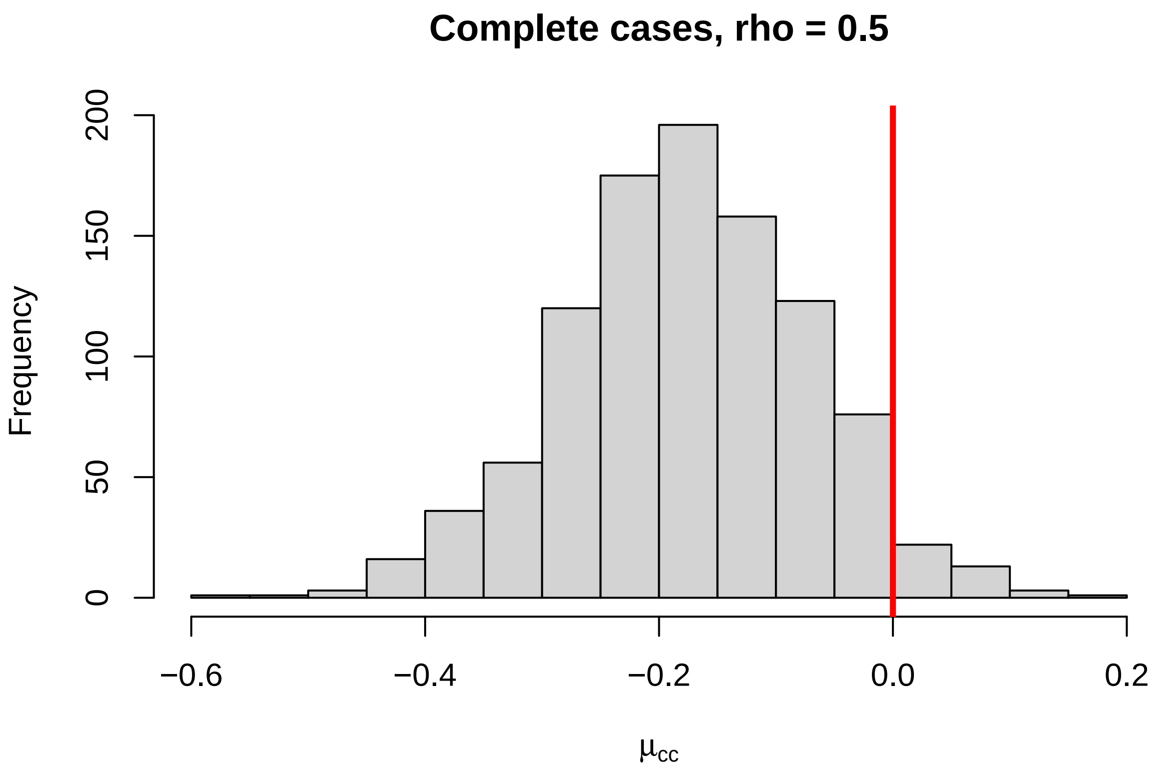


```
hist(res_rho0[[1]], main = "MLE, rho = 0", xlab = expression(mu[mle]), xlim = c(-0.4, 0.4))
abline(v = 0, col = "red", lwd = 3)
```

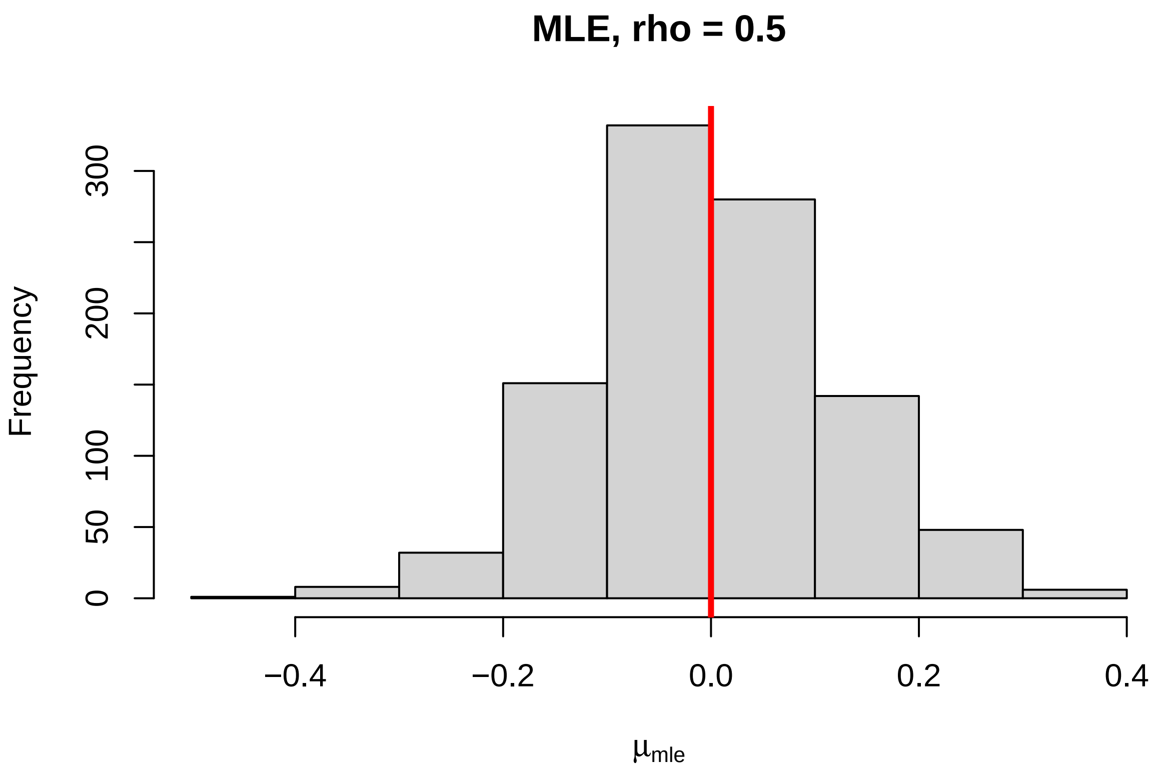


The results depicted in the histograms are not surprising, as in the case of independence between Y_1 and Y_2, s_{12} should be close to 0, and therefore $\widehat{\beta}_1 \approx 0$, $\widehat{\beta}_1 \approx \bar{y}_2$, and thus both the complete cases and the mle estimators coincide. We will now simulate data with $\rho = 0.5$ and we expect the performance of the complete cases estimator to deteriorate, whereas the mle should provide correct estimates.

```
rho <- 0.5
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))
set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

res_rho50 <- bivnorm(y = y, a = 0.8)

hist(res_rho50[[2]], main = "Complete cases, rho = 0.5", xlab = expression(mu[cc]))
abline(v = 0, col = "red", lwd = 3)
```

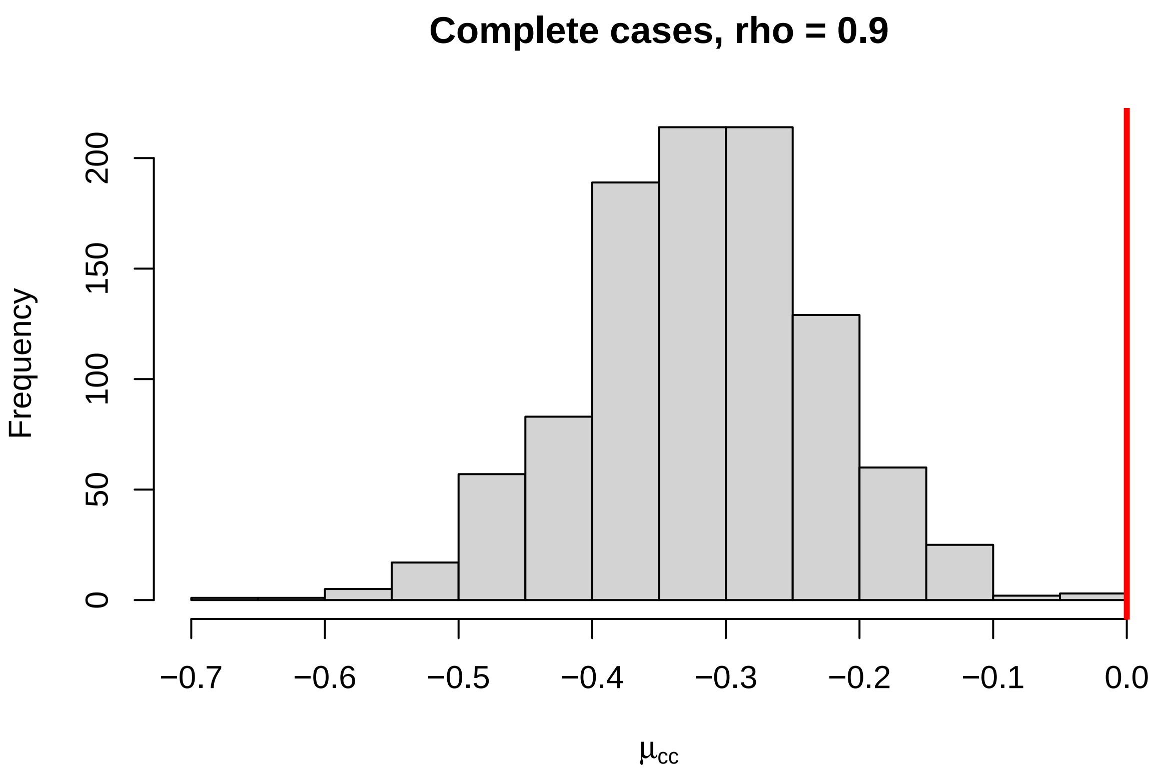


```r
hist(res_rho50[[1]], main = "MLE, rho = 0.5", xlab = expression(mu[mle]))
abline(v = 0, col = "red", lwd = 3)
```

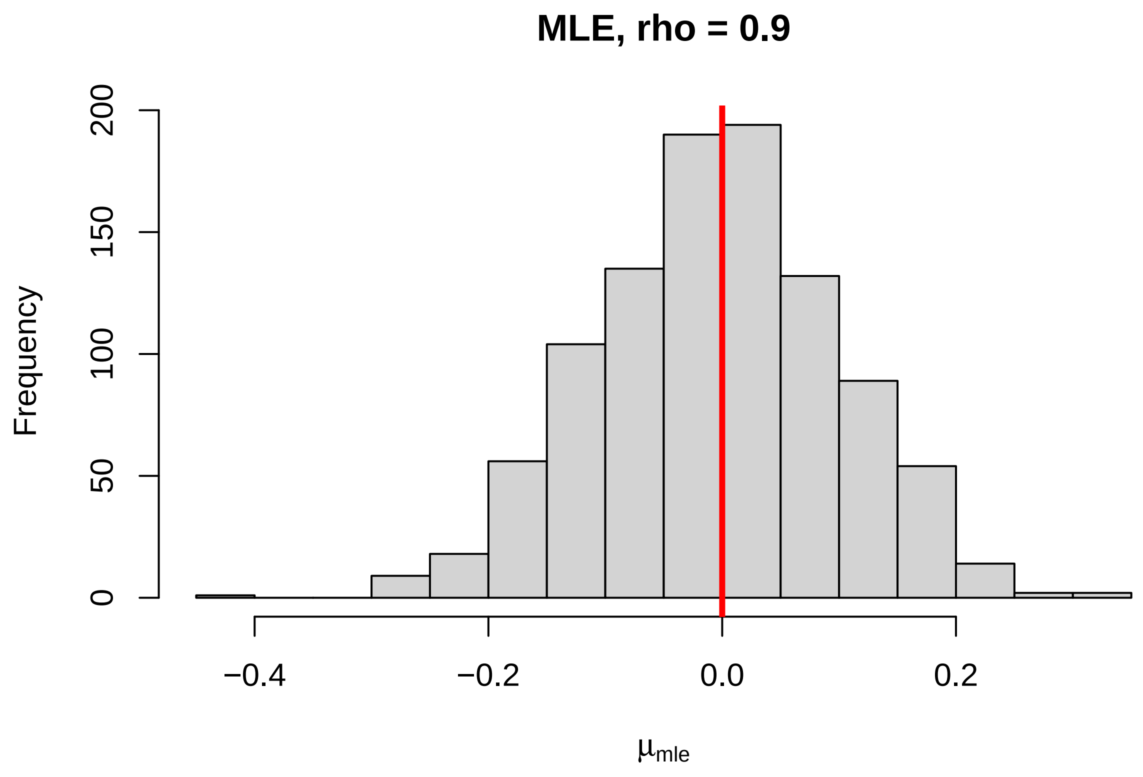


The plots confirm what we just stated we should expect. Now we will try $\rho = 0.9$ and we should expect the complete cases estimator to perform even worse.

```r
rho <- 0.9
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))
set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

res_rho90 <- bivnorm(y = y, a = 0.8)

hist(res_rho90[[2]], main = "Complete cases, rho = 0.9", xlab = expression(mu[cc]))
abline(v = 0, col = "red", lwd = 3)
```



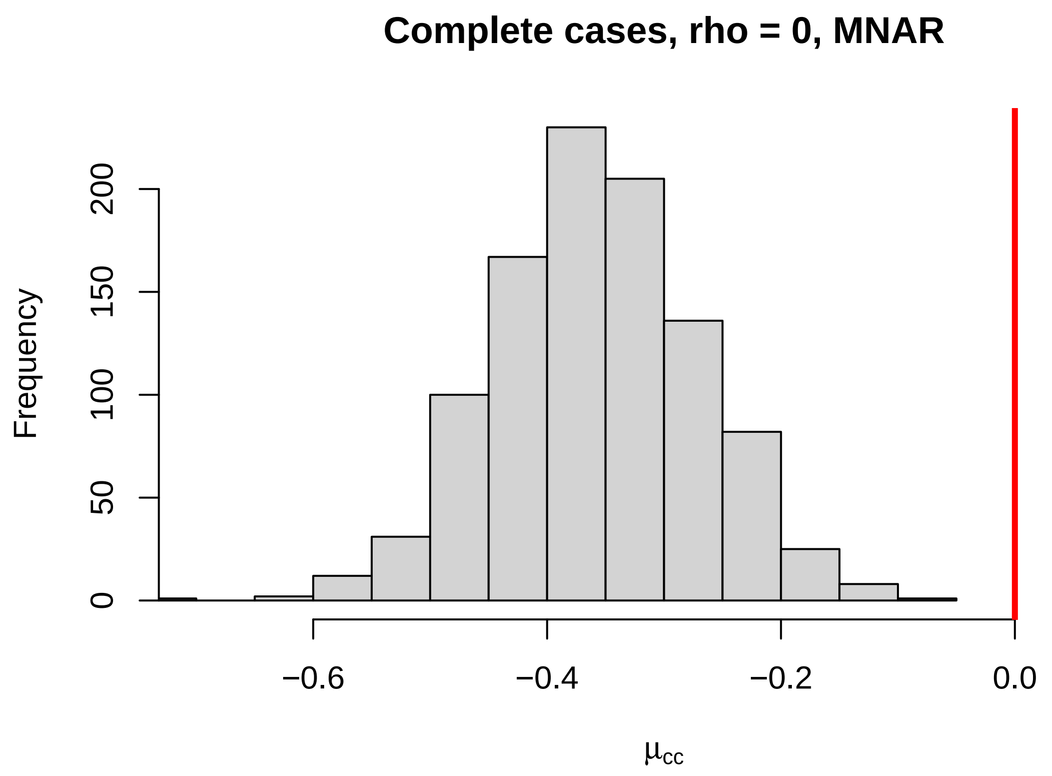
```r
hist(res_rho90[[1]], main = "MLE, rho = 0.9", xlab = expression(mu[mle]))
abline(v = 0, col = "red", lwd = 3)
```



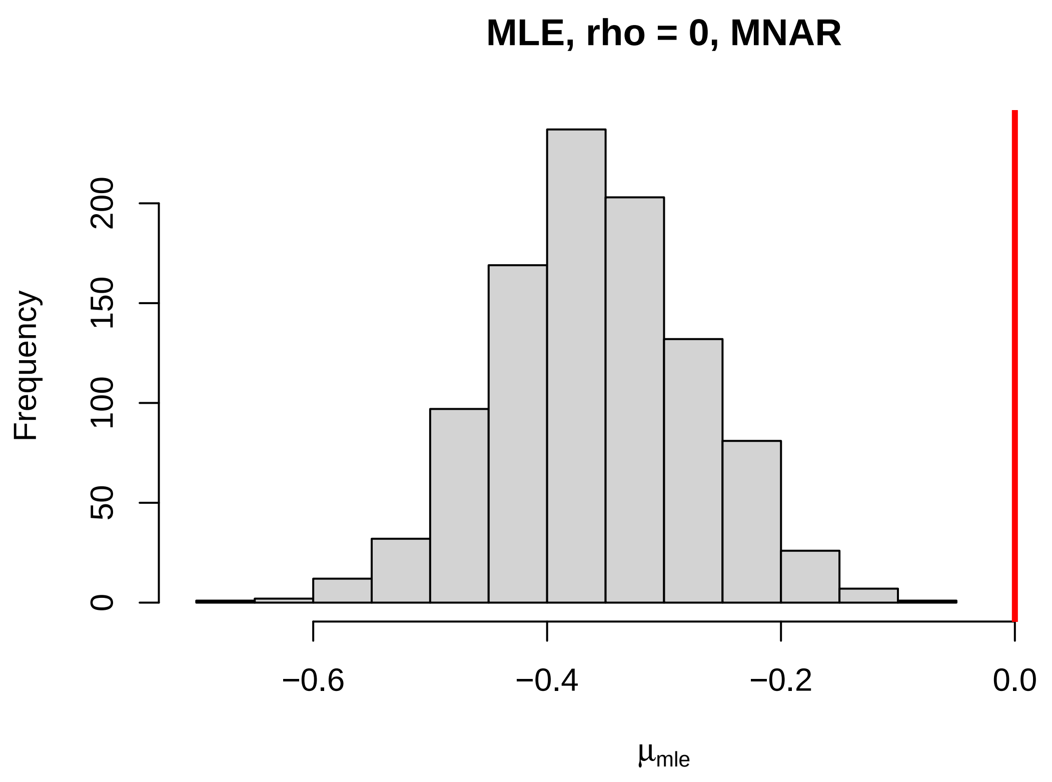
We will now break the assumption of MAR data and, instead, simulate MNAR data. We follow exactly the same procedure, with the only difference being that $r_i = 0$ (i.e., y_{2i} is missing) if $y_{2i} > z_p$, where $z_p = \Phi^{-1}(a)$, and $a = 0.8$. As we know from one of the last week examples, we should expect the mle to be severely biased in this situation.

```
rho <- 0
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))
set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

bivnorm_mnar <- function(y, a){
  n <- dim(y)[1]
  nsim <- dim(y)[3]
  y1 <- y2 <- r <- matrix(0, nrow = n, ncol = nsim)
  mu2mle <- mu2cc <- numeric(nsim)
  treshold <- qnorm(a, 0, 1)

  for(l in 1:nsim){

    y1[, l] <- y[, 1, l]
    y2[, l] <- y[, 2, l]
    r[,l] <- ifelse(y2[, l] > treshold, 0, 1)

    s1squared <- mean((y1[r[, l] == 1, l] - mean(y1[r[, l] == 1, l]))^2)
    s12 <- mean((y1[r[, l] == 1, l] -
                   mean(y1[r[, l] == 1, l]))*(y2[r[, l] == 1, l] - mean(y2[r[, l] == 1, l])))
```

```r
    beta1 <- s12/s1squared
    beta0 <- mean(y2[r[,l] == 1, l]) - beta1*mean(y1[r[,l] == 1, l])
    mu1 <- mean(y1[,l])

    mu2mle[l] <- beta0 + beta1*mu1
    mu2cc[l] <- mean(y2[r[, l] == 1, l])
  }
  return(list(mu2mle, mu2cc))
}

res_rho0_mnar <- bivnorm_mnar(y = y, a = 0.8)

hist(res_rho0_mnar[[2]], main = "Complete cases, rho = 0, MNAR", xlab = expression(mu[cc]),
     xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```

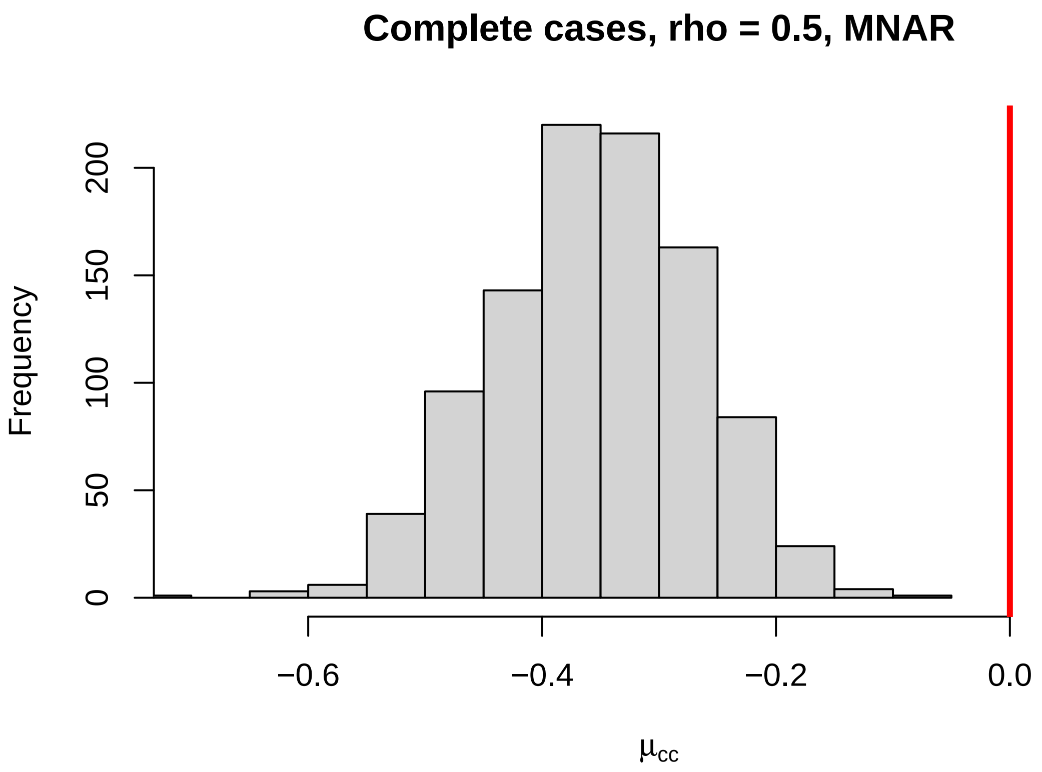


```r
hist(res_rho0_mnar[[1]], main = "MLE, rho = 0, MNAR",
     xlab = expression(mu[mle]), xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```

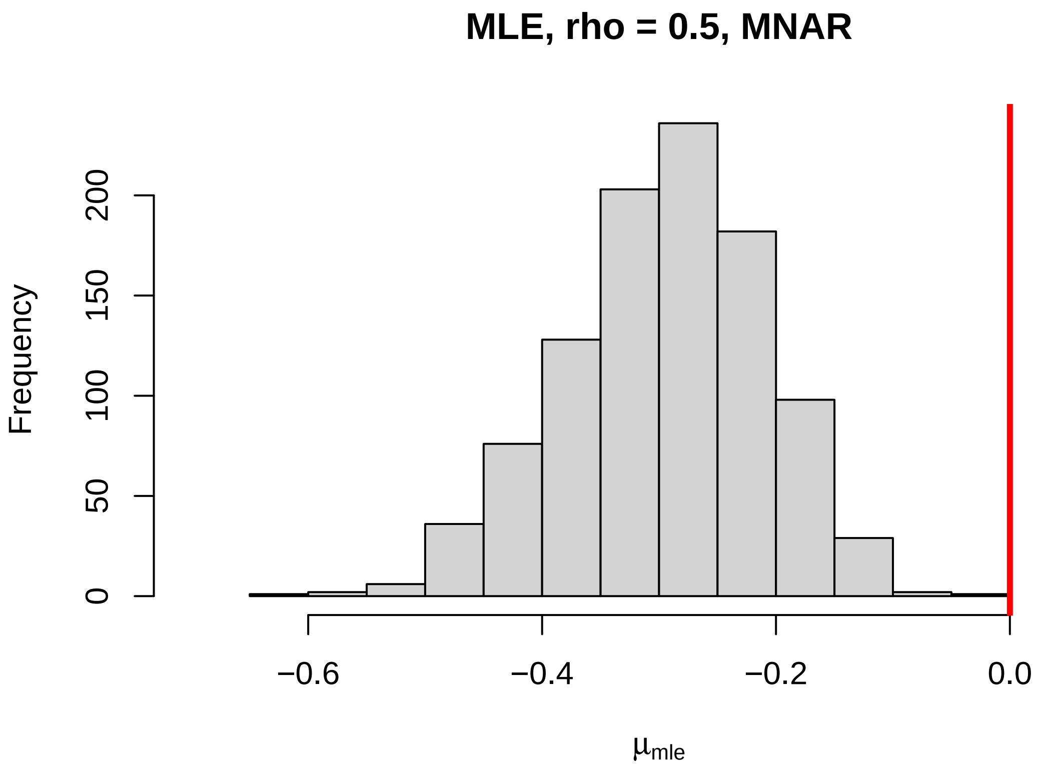


Now trying $\rho = 0.5$.

```
rho <- 0.5
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))
set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

res_rho50_mnar <- bivnorm_mnar(y = y, a = 0.8)

hist(res_rho50_mnar[[2]], main = "Complete cases, rho = 0.5, MNAR", xlab = expression(mu[cc]),
     xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```

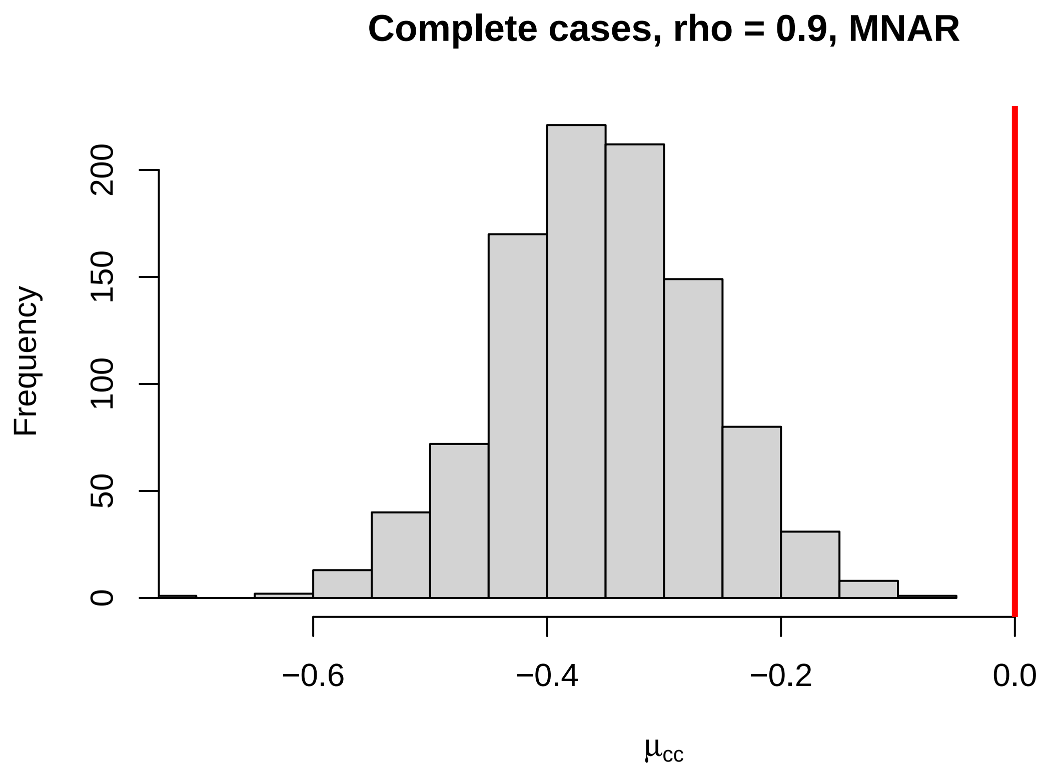


```r
hist(res_rho50_mnar[[1]], main = "MLE, rho = 0.5, MNAR",
     xlab = expression(mu[mle]), xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```

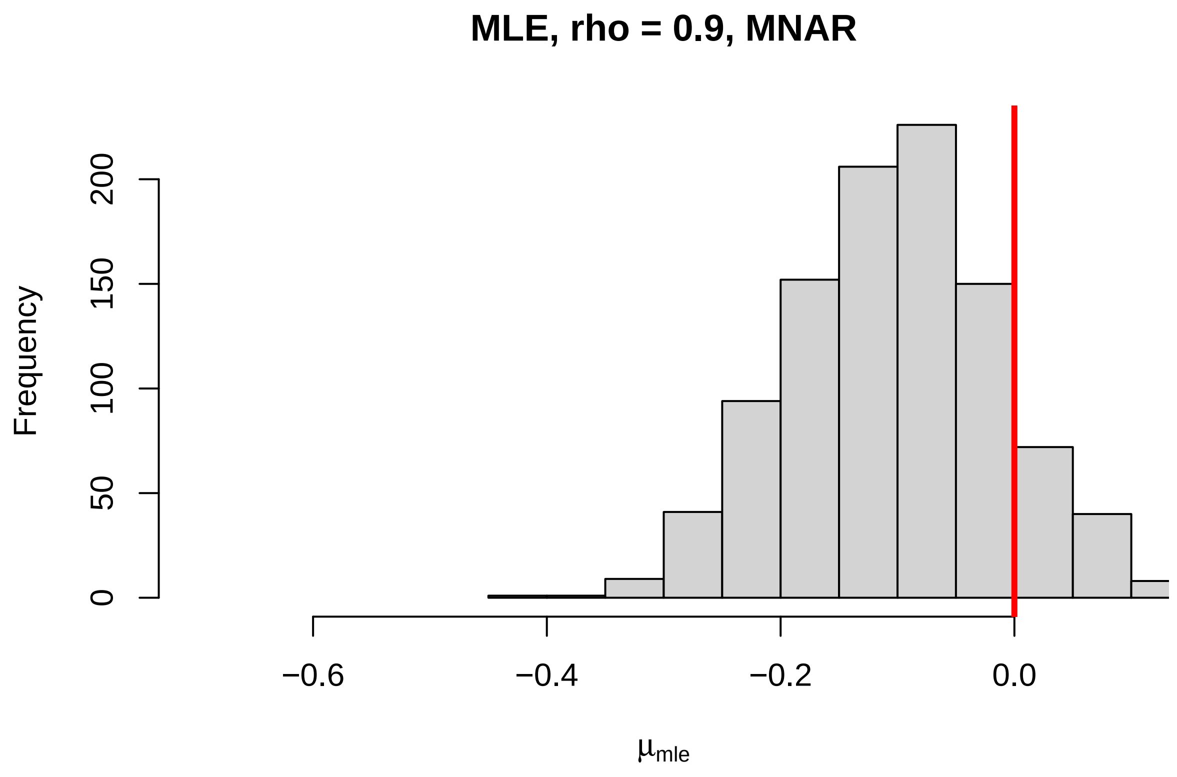


And finally for $\rho = 0.9$.

```r
rho <- 0.9
Sigma <- matrix(c(sigma1^2, rho*sigma1*sigma2, rho*sigma1*sigma2, sigma2^2),
                nrow = 2, ncol = 2, byrow = T)
y <- array(0, c(n, 2, nsim))
set.seed(1)
for(l in 1:nsim){
y[, , l] <- mvrnorm(n, mu = mu, Sigma = Sigma)
}

res_rho50_mnar <- bivnorm_mnar(y = y, a = 0.8)

hist(res_rho50_mnar[[2]], main = "Complete cases, rho = 0.9, MNAR", xlab = expression(mu[cc]),
     xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```



```r
hist(res_rho50_mnar[[1]], main = "MLE, rho = 0.9, MNAR",
     xlab = expression(mu[mle]), xlim = c(-0.7,0.1))
abline(v = 0, col = "red", lwd = 3)
```


MLE, rho = 0.9, MNAR
Frequency
0
50
100
150
200
−0.6
−0.4
−0.2
0.0
μ_{mle}