Incomplete Data Analysis

Simulation study—naive and single imputation mechanisms

School of Mathematics, University of Edinburgh

V. Inácio de Carvalho & M. de Carvalho

One way to understand missing data mechanisms and naive and single imputation methods (or any other method developed for dealing with missing values) is to generate hypothetical complete data, create missing values by specific mechanisms and then apply the different methods we are interested in studying the performance to the generated data.

A Monte Carlo simulation study generates a large number of samples (e.g., 1000) from a population with a specified set of parameter values. Estimating a statistical model on each sample and saving the resulting parameter estimates creates an empirical sampling distribution for each model parameter. Then, for instance, the difference between the average parameter estimate and the true population parameter quantifies the bias. I have uploaded to Learn three interesting papers (in my opinion!) about simulation studies.

Here in this simulation exercise we will generate 1000 data sets, each consisting of 200 observations, from a bivariate normal distribution with the following structure

$$\boldsymbol{\mu} = \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix} = \begin{pmatrix}0\\ 0\end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix}\sigma_1^2 & \sigma_{1,2}\\ \sigma_{2,1} & \sigma_2^2\end{pmatrix} = \begin{pmatrix}1 & 0.5\\ 0.5 & 1\end{pmatrix}, \qquad \rho = 0.5.$$

We then write

$$\begin{pmatrix}Y_1\\ Y_2\end{pmatrix} \sim \mathrm{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

We will impose missingness on Y_2 only and let R be the missingness indicator. More specifically, we will create MCAR data by imposing

$$\Pr(R = 1 \mid Y_1, Y_2, \boldsymbol{\beta}) = 0.5.$$

In turn, MAR data will be generated by imposing

$$\Pr(R = 1 \mid Y_1, Y_2, \boldsymbol{\beta}) = \frac{e^{\beta_0+\beta_1 Y_1}}{1+e^{\beta_0+\beta_1 Y_1}}, \quad \beta_0 = 1.5, \quad \beta_1 = 3.$$

Finally, MNAR will be generated in the following way

$$\Pr(R = 1 \mid Y_1, Y_2, \boldsymbol{\beta}) = \frac{e^{\beta_0+\beta_1 Y_1+\beta_2 Y_2}}{1+e^{\beta_0+\beta_1 Y_1+\beta_2 Y_2}}, \quad \beta_0 = 1.5, \quad \beta_1 = 3, \quad \beta_2 = 5.$$

The goal of this simulation study is to assess the performance of the different strategies (complete case analysis, mean imputation, regression imputation, and stochastic regression imputation) under the three different missingness mechanisms. We will assess how well μ_2, σ_2^2 and ρ are estimated under the different strategies. We will also assess the empirical coverage probability of the 95% confidence interval for μ_2. In order to calculate the empirical coverage probability, we only need to compute the proportion of the time (over the 1000 intervals) that the interval contains the true value. This should be close to the nominal value of 95%. Remember that because $Y_2 \sim \mathrm{N}(\mu_2, \sigma_2^2)$, we now that a 95% confidence interval for μ_2 is of the form

$$\left[\widehat{\mu}_2 - t_{n-1,0.975}\frac{\widehat{\sigma}_2}{\sqrt{n}}, \widehat{\mu}_2 + t_{n-1,0.975}\frac{\widehat{\sigma}_2}{\sqrt{n}}\right],$$

where n is the sample size (200 in our case) and t stands for the Student's t distribution.

Let us start by simulating the 1000 datasets, each of size 200. Because we are mainly interested on Y_2, I will store the results in two separate variables.

```
require(MASS)

mu1 <- 0; mu2 <- 0; sigma12 <- 1; sigma22 <- 1; rho <- 0.5
cov12 <- rho*sqrt(sigma12)*sqrt(sigma22)
Sigma <- matrix(c(sigma12, cov12, cov12, sigma22), nrow = 2, ncol = 2, byrow = TRUE)

nsim <- 1000; n <- 200
y <- array(0, c(n, 2, nsim))
y1 <- y2 <- matrix(0, nrow = n, ncol = nsim)

set.seed(1)
for(l in 1:nsim){
  y[, , l] <- mvrnorm(n = n, mu = c(mu1, mu2), Sigma = Sigma)
  y1[, l] <- y[, 1, l]
  y2[, l] <- y[, 2, l]
}
```

The next step is to generate the missing data indicator and we will start with the MCAR mechanism.

```
r <- matrix(0, nrow = n, ncol = nsim)
prob <- 0.5
for(l in 1:nsim){
  r[, l] <- rbinom(n = n, size = 1, prob = prob)
}
```

We now have everything we need to start our analysis. I have constructed a function, `single_imp_sim`, that takes as inputs the data on Y_1 and Y_2, the generated missing data indicators and the method that we want to apply. In the following, CCA stands for complete case analysis, MI for mean imputation, RI stands for regression imputation, and SRI for stochastic regression imputation. The output returned by the function, for the selected method, are the estimates (for the 1000 generated data sets) of μ_2, σ_2^2, ρ, and the lower and upper limits of the 95% confidence interval for μ_2.

```
single_imp_sim <- function(y1, y2, r, method = c("CCA", "MI", "RI", "SRI")){
  nsim <- ncol(y1)
  n <- nrow(y1)
  estimates <- matrix(0, nrow = 5, ncol = nsim)

  if(method == "CCA"){
    for(l in 1: nsim){
    estimates[1, l] <- mean(y2[r[, l] == 1, l])
    estimates[2, l] <- var(y2[r[, l] == 1, l])
    estimates[3, l] <- cor(y1[r[, l] == 1 , l], y2[r[, l] == 1, l])
    #lower bound CI
    estimates[4, l] <- estimates[1, l] -
      qt(0.975, sum(r[, l] == 1)-1)*sqrt(estimates[2, l]/sum(r[, l] == 1))
    #upper bound CI
    estimates[5, l] <- estimates[1, l] +
      qt(0.975, sum(r[, l] == 1)-1)*sqrt(estimates[2, l]/sum(r[, l] == 1))
    }
  }
  if(method == "MI"){
    for(l in 1:nsim){
      y2_completed <- ifelse(r[, l] == 0, mean(y2[r[, l] == 1, l]), y2[, l])
      estimates[1, l] <- mean(y2_completed)
      estimates[2, l] <- var(y2_completed)
      estimates[3, l] <- cor(y1[, l], y2_completed)
      estimates[4, l] <- estimates[1, l] - qt(0.975, n - 1)*sqrt(estimates[2, l]/n)
      estimates[5, l] <- estimates[1, l] + qt(0.975, n - 1)*sqrt(estimates[2, l]/n)
    }
  }

  if(method == "RI"){
    for(l in 1:nsim){
      fit <- lm(y2[r[, l] == 1, l] ~ y1[r[, l] == 1, l])
      pred <- fit$coefficients[1] + y1[ ,l]*fit$coefficients[2]
      y2_completed <- ifelse(r[, l] == 0, pred, y2[, l])
      estimates[1, l] <- mean(y2_completed)
      estimates[2, l] <- var(y2_completed)
      estimates[3, l] <- cor(y1[, l],y2_completed)
      estimates[4, l] <- estimates[1, l] - qt(0.975, n-1)*sqrt(estimates[2, l]/n)
      estimates[5, l] <- estimates[1, l] + qt(0.975, n-1)*sqrt(estimates[2, l]/n)
    }
    }

  if(method == "SRI"){
    for(l in 1:nsim){
      fit <- lm(y2[r[, l] == 1, l] ~ y1[r[, l] == 1, l])
      sigmaest <- sigma(fit)
      pred <- fit$coefficients[1] + y1[ ,l]*fit$coefficients[2] + rnorm(n, 0, sigmaest)
      y2_completed <- ifelse(r[, l] == 0, pred, y2[, l])
      estimates[1, l] <- mean(y2_completed)
      estimates[2, l] <- var(y2_completed)
      estimates[3, l] <- cor(y1[, l],y2_completed)
      estimates[4, l] <- estimates[1, l] - qt(0.975, n-1)*sqrt(estimates[2, l]/n)
      estimates[5, l] <- estimates[1, l] + qt(0.975, n-1)*sqrt(estimates[2, l]/n)
    }
  }

  return(list("mu2ests" = estimates[1, ], "var2ests" = estimates[2, ],
              "corrests" = estimates[3, ], "lls" = estimates[4, ],
              "uls" = estimates[5, ]))
}
```

Let us now apply the function to the generated MCAR data.

```
res_CCA_MCAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "CCA")
#Monte Carlo means
mu2_CCA_MCAR <- mean(res_CCA_MCAR$mu2ests)
sigma22_CCA_MCAR <- mean(res_CCA_MCAR$var2ests)
rho_CCA_MCAR <- mean(res_CCA_MCAR$corrests)
coverage_CCA_MCAR <- sum((res_CCA_MCAR$lls <= mu2) & (res_CCA_MCAR$uls >= mu2))/nsim

res_MI_MCAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "MI")
mu2_MI_MCAR <- mean(res_MI_MCAR$mu2ests)
```

```r
sigma22_MI_MCAR <- mean(res_MI_MCAR$var2ests)
rho_MI_MCAR <- mean(res_MI_MCAR$corrests)
coverage_MI_MCAR <- sum((res_MI_MCAR$lls <= mu2) & (res_MI_MCAR$uls >= mu2))/nsim

res_RI_MCAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "RI")
mu2_RI_MCAR <- mean(res_RI_MCAR$mu2ests)
sigma22_RI_MCAR <- mean(res_RI_MCAR$var2ests)
rho_RI_MCAR <- mean(res_RI_MCAR$corrests)
coverage_RI_MCAR <- sum((res_RI_MCAR$lls <= mu2) & (res_RI_MCAR$uls >= mu2))/nsim

res_SRI_MCAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "SRI")
mu2_SRI_MCAR <- mean(res_SRI_MCAR$mu2ests)
sigma22_SRI_MCAR <- mean(res_SRI_MCAR$var2ests)
rho_SRI_MCAR <- mean(res_SRI_MCAR$corrests)
coverage_SRI_MCAR <- sum((res_SRI_MCAR$lls <= mu2) & (res_SRI_MCAR$uls >= mu2))/nsim

df <- data.frame("Mu2" = c(mu2_CCA_MCAR, mu2_MI_MCAR, mu2_RI_MCAR, mu2_SRI_MCAR),
                 "Sigma22" = c(sigma22_CCA_MCAR, sigma22_MI_MCAR,
                               sigma22_RI_MCAR, sigma22_SRI_MCAR),
                 "Rho" = c(rho_CCA_MCAR, rho_MI_MCAR,
                           rho_RI_MCAR, rho_SRI_MCAR),
                 "Cov" = c(coverage_CCA_MCAR,coverage_MI_MCAR,coverage_RI_MCAR,
                           coverage_SRI_MCAR))
rownames(df) <- c("CCA", "MI", "RI", "SRI")
colnames(df) = c("$\\mu_2$", "$\\sigma_2^2$", "$\\rho$", "Coverage")

knitr::kable(df, escape = FALSE, digits = 4, caption = "MCAR data")
```

Table 1: MCAR data

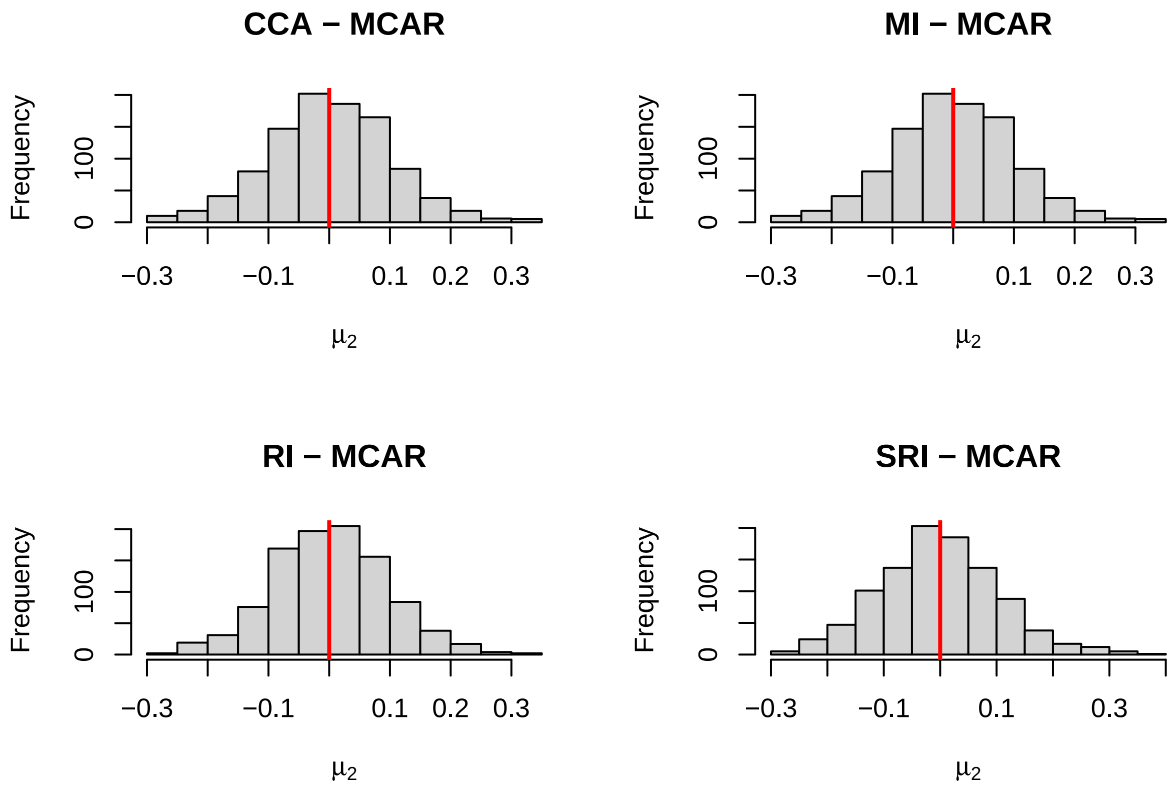
	μ_2	σ_2^2	ρ	Coverage
CCA	0.0007	1.0016	0.4962	0.945
MI	0.0007	0.4986	0.3499	0.683
RI	0.0021	0.6270	0.6279	0.772
SRI	-0.0005	1.0036	0.4971	0.815

```r
par(mfrow = c(2,2))
hist(res_CCA_MCAR$mu2ests, xlab = expression(mu[2]), main = "CCA - MCAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_MI_MCAR$mu2ests, xlab = expression(mu[2]), main = "MI - MCAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_RI_MCAR$mu2ests, xlab = expression(mu[2]), main = "RI - MCAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_SRI_MCAR$mu2ests, xlab = expression(mu[2]), main = "SRI - MCAR")
abline(v = mu2, col = "red", lwd = 2)
```

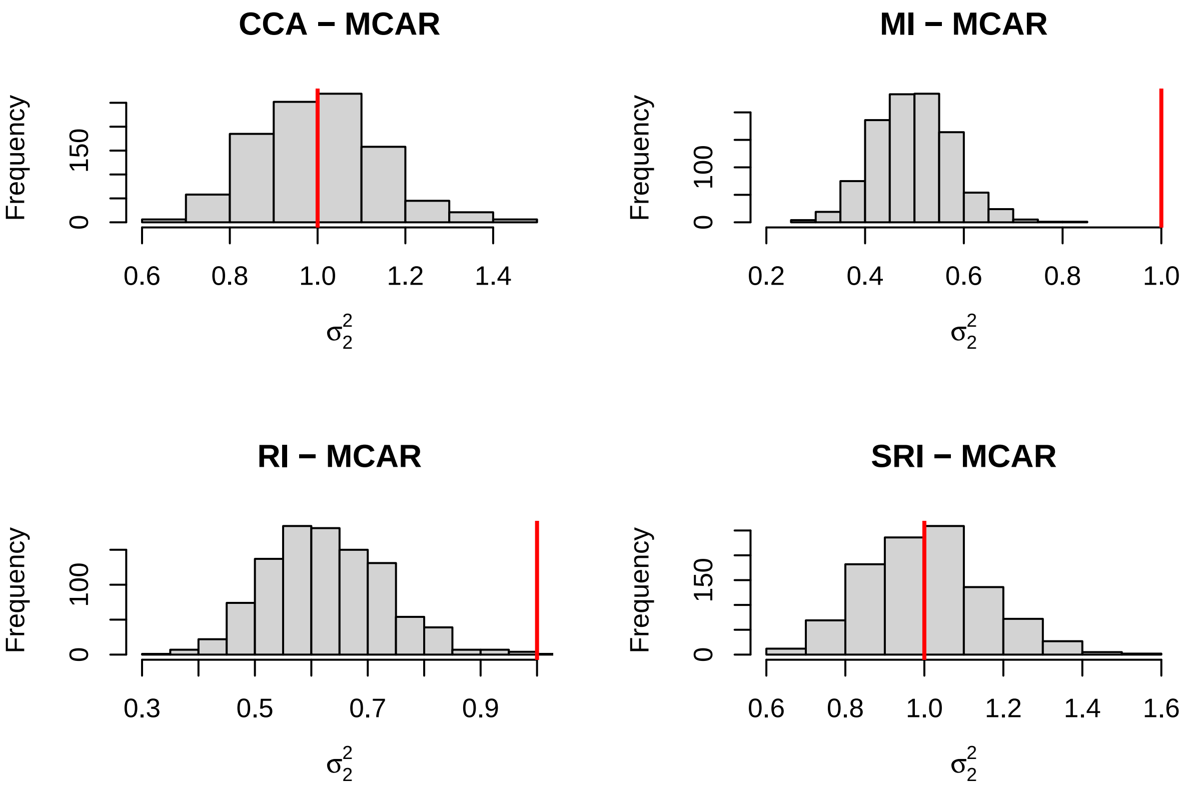


```r
par(mfrow = c(2,2))
hist(res_CCA_MCAR$var2ests, xlab = expression(sigma[2]^2), main = "CCA - MCAR")
abline(v = sigma22, col = "red", lwd = 2)

hist(res_MI_MCAR$var2ests, xlab = expression(sigma[2]^2), main = "MI - MCAR",
      xlim = c(0.2, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_RI_MCAR$var2ests, xlab = expression(sigma[2]^2), main = "RI - MCAR",
     xlim = c(0.3, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_SRI_MCAR$var2ests, xlab = expression(sigma[2]^2), main = "SRI - MCAR")
abline(v = sigma22, col = "red", lwd = 2)
```

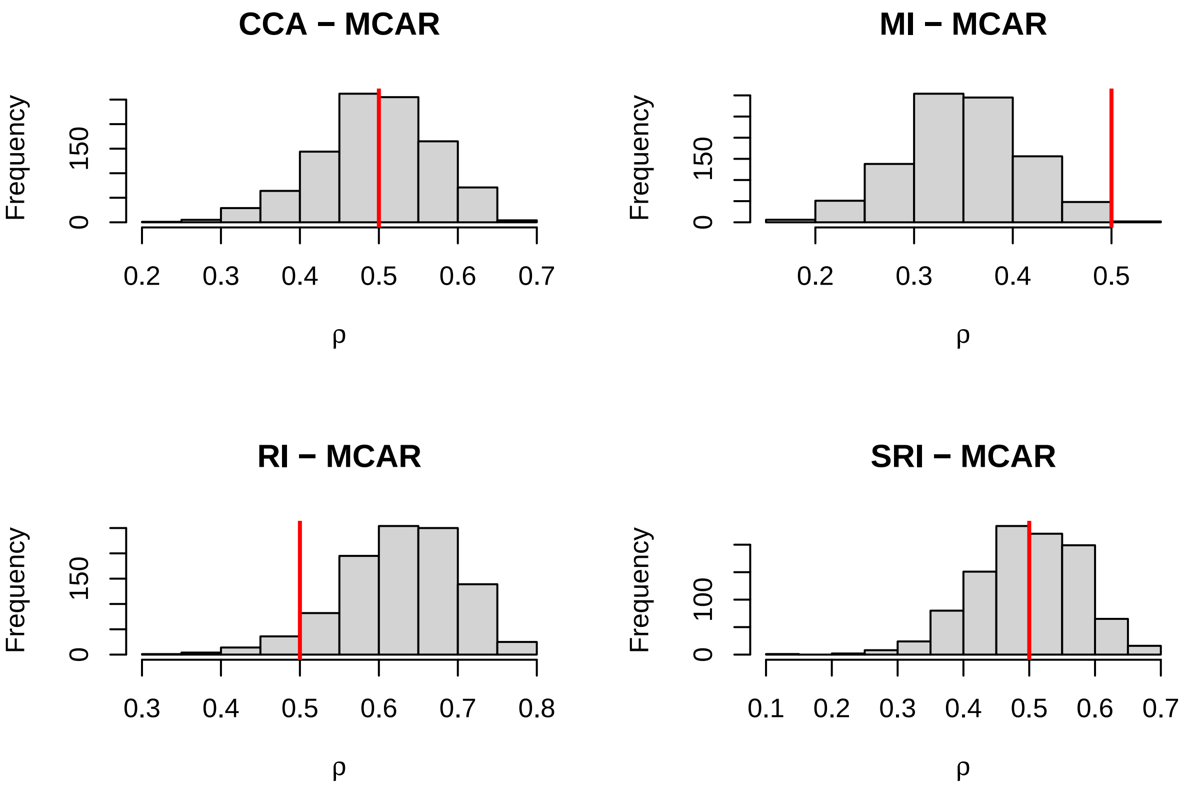


```r
par(mfrow = c(2,2))
hist(res_CCA_MCAR$corrests, xlab = expression(rho), main = "CCA - MCAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_MI_MCAR$corrests, xlab = expression(rho), main = "MI - MCAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_RI_MCAR$corrests, xlab = expression(rho), main = "RI - MCAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_SRI_MCAR$corrests, xlab = expression(rho), main = "SRI - MCAR")
abline(v = rho, col = "red", lwd = 2)
```



From the results above (table and histograms), we can notice that all methods estimate well the mean of Y_2 (the true value is 0). We also see that the variance of Y_2 (true value is 1) is underestimated with the imputation by the mean and by regression. The correlation (between Y_1 and Y_2) is completely destroyed by these two imputation methods as well. Under the the complete case analysis and stochastic regression imputation methods, the variance of Y_2 and the correlation are well estimated. With regard to the coverage probability, only the complete case analysis provides an adequate coverage. Even stochastic regression does not provide an accurate coverage under MCAR. This is because it is a single imputation method and, as such, does not reflect the variability due to the missing values. Indeed, with stochastic regression imputation, the estimators of the mean, of the variance and of the correlation coefficient are unbiased but the variance of the estimators (the variance of the estimator of the mean of Y_2) are too low and this is reflected in the coverage probability. Still, with regard to the coverage of the CCA approach, although it is correct, the width of the intervals is much larger than those based on the complete data, as shown below.

```r
#estimates for the complete data
mu2est_complete_data <- apply(y2, 2, mean)
var2est_complete_data <- apply(y2, 2, var)
ll_complete_data <- ul_complete_data <- numeric(nsim)
for(l in 1:nsim){
   ll_complete_data[l] <- mu2est_complete_data[l] -
     qt(0.975, n-1)*sqrt(var2est_complete_data[l]/n)
   ul_complete_data[l] <- mu2est_complete_data[l] +
     qt(0.975, n-1)*sqrt(var2est_complete_data[l]/n)
}
width_complete_data <- ul_complete_data - ll_complete_data

width_CCA <- res_CCA_MCAR$uls - res_CCA_MCAR$lls

mean(width_complete_data)
```

```
## [1] 0.2788024
```

```
mean(width_CCA)
```

```
## [1] 0.3968747
```

Let us now move to the MAR data case. We will impose the missingness as instructed before and we will then pass this to the function we have created and produce the results for the four methods.

```
beta0 <- 1.5; beta1 <- 3
r <- matrix(0, nrow = n, ncol = nsim)
for(l in 1:nsim){
  r[, l] <- rbinom(n, 1, exp(beta0+beta1*y1[,l])/(1+exp(beta0+beta1*y1[,l])))
}

res_CCA_MAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "CCA")
mu2_CCA_MAR <- mean(res_CCA_MAR$mu2ests)
sigma22_CCA_MAR <- mean(res_CCA_MAR$var2ests)
rho_CCA_MAR <- mean(res_CCA_MAR$corrests)
coverage_CCA_MAR <- sum((res_CCA_MAR$lls <= mu2) & (res_CCA_MAR$uls >= mu2))/nsim

res_MI_MAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "MI")
mu2_MI_MAR <- mean(res_MI_MAR$mu2ests)
sigma22_MI_MAR <- mean(res_MI_MAR$var2ests)
rho_MI_MAR <- mean(res_MI_MAR$corrests)
coverage_MI_MAR <- sum((res_MI_MAR$lls <= mu2) & (res_MI_MAR$uls >= mu2))/nsim

res_RI_MAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "RI")
mu2_RI_MAR <- mean(res_RI_MAR$mu2ests)
sigma22_RI_MAR <- mean(res_RI_MAR$var2ests)
rho_RI_MAR <- mean(res_RI_MAR$corrests)
coverage_RI_MAR <- sum((res_RI_MAR$lls <= mu2) & (res_RI_MAR$uls >= mu2))/nsim

res_SRI_MAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "SRI")
mu2_SRI_MAR <- mean(res_SRI_MAR$mu2ests)
sigma22_SRI_MAR <- mean(res_SRI_MAR$var2ests)
rho_SRI_MAR <- mean(res_SRI_MAR$corrests)
coverage_SRI_MAR <- sum((res_SRI_MAR$lls <= mu2) & (res_SRI_MAR$uls >= mu2))/nsim

df <- data.frame("Mu2" = c(mu2_CCA_MAR, mu2_MI_MAR, mu2_RI_MAR, mu2_SRI_MAR),
                 "Sigma22" = c(sigma22_CCA_MAR, sigma22_MI_MAR,
                               sigma22_RI_MAR, sigma22_SRI_MAR),
                 "Rho" = c(rho_CCA_MAR, rho_MI_MAR,
                           rho_RI_MAR, rho_SRI_MAR),
                 "Cov" = c(coverage_CCA_MAR,coverage_MI_MAR,coverage_RI_MAR,
                           coverage_SRI_MAR))
rownames(df) <- c("CCA", "MI", "RI", "SRI")
colnames(df) = c("$\\mu_2$", "$\\sigma_2^2$", "$\\rho$", "Coverage")

knitr::kable(df, escape = FALSE, digits = 4, caption = "MAR data")
```

Table 2: MAR data

	μ_2	σ_2^2	ρ	Coverage
CCA	0.2352	0.9000	0.4060	0.183
MI	0.2352	0.5984	0.2569	0.062
RI	0.0007	0.7539	0.5715	0.791

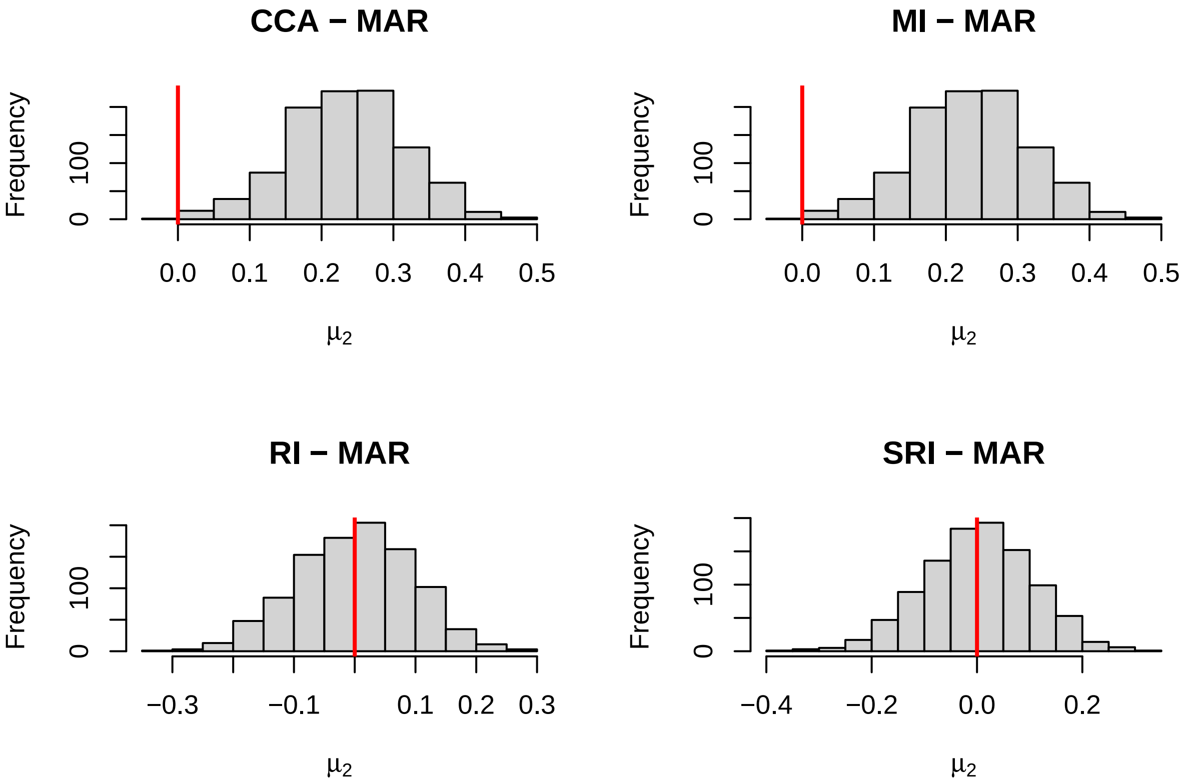
	μ_2	σ_2^2	ρ	Coverage
SRI	0.0025	1.0042	0.4935	0.818

```
par(mfrow = c(2,2))
hist(res_CCA_MAR$mu2ests, xlab = expression(mu[2]), main = "CCA - MAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_MI_MAR$mu2ests, xlab = expression(mu[2]), main = "MI - MAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_RI_MAR$mu2ests, xlab = expression(mu[2]), main = "RI - MAR")
abline(v = mu2, col = "red", lwd = 2)

hist(res_SRI_MAR$mu2ests, xlab = expression(mu[2]), main = "SRI - MAR")
abline(v = mu2, col = "red", lwd = 2)
```

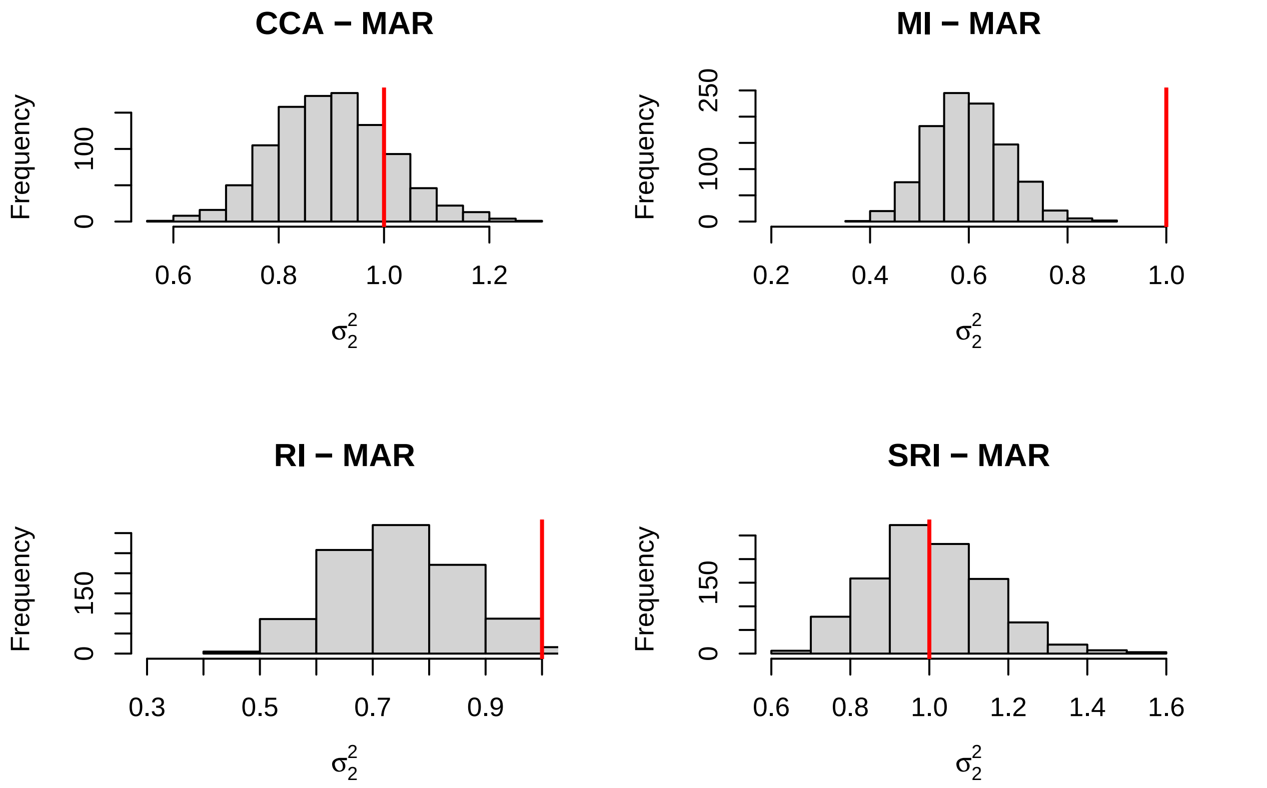


```
par(mfrow = c(2,2))
hist(res_CCA_MAR$var2ests, xlab = expression(sigma[2]^2), main = "CCA - MAR")
abline(v = sigma22, col = "red", lwd = 2)

hist(res_MI_MAR$var2ests, xlab = expression(sigma[2]^2), main = "MI - MAR",
      xlim = c(0.2, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_RI_MAR$var2ests, xlab = expression(sigma[2]^2), main = "RI - MAR",
     xlim = c(0.3, 1))
abline(v = sigma22, col = "red", lwd = 2)
```

```r
hist(res_SRI_MAR$var2ests, xlab = expression(sigma[2]^2), main = "SRI - MAR")
abline(v = sigma22, col = "red", lwd = 2)
```

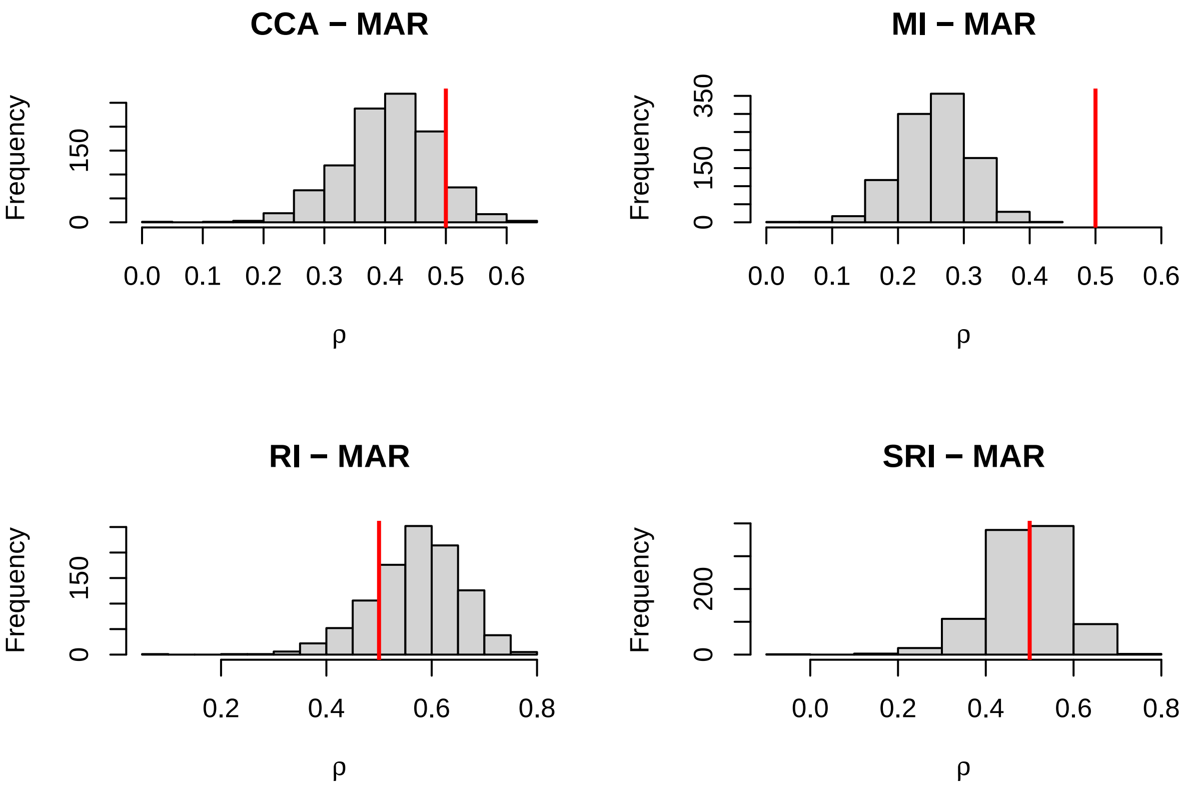


```r
par(mfrow = c(2,2))
hist(res_CCA_MAR$corrests, xlab = expression(rho), main = "CCA - MAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_MI_MAR$corrests, xlab = expression(rho), main = "MI - MAR",
     xlim = c(0, 0.6))
abline(v = rho, col = "red", lwd = 2)

hist(res_RI_MAR$corrests, xlab = expression(rho), main = "RI - MAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_SRI_MAR$corrests, xlab = expression(rho), main = "SRI - MAR")
abline(v = rho, col = "red", lwd = 2)
```



For this scenario (MAR data), only stochastic regression imputation manages to provide accurate estimates of the mean, variance, and correlation although, as expected, the coverage is still quite below the nominal level (0.95).

Lastly, let us see what happens with missingness imposed under a MNAR mechanism.

```
beta0 <- 1.5; beta1 <- 3; beta2 <- 5
r <- matrix(0, nrow = n, ncol = nsim)
for(i in 1:nsim){
  r[, i] <- rbinom(n, 1, exp(beta0+beta1*y1[,i]+beta2*y2[,i])/(1+exp(beta0+beta1*y1[,i]+beta2*y2[,i])))
}

res_CCA_MNAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "CCA")
mu2_CCA_MNAR <- mean(res_CCA_MNAR$mu2ests)
sigma22_CCA_MNAR <- mean(res_CCA_MNAR$var2ests)
rho_CCA_MNAR <- mean(res_CCA_MNAR$corrests)
coverage_CCA_MNAR <- sum((res_CCA_MNAR$lls <= mu2) & (res_CCA_MNAR$uls >= mu2))/nsim

res_MI_MNAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "MI")
mu2_MI_MNAR <- mean(res_MI_MNAR$mu2ests)
sigma22_MI_MNAR <- mean(res_MI_MNAR$var2ests)
rho_MI_MNAR <- mean(res_MI_MNAR$corrests)
coverage_MI_MNAR <- sum((res_MI_MNAR$lls <= mu2) & (res_MI_MNAR$uls >= mu2))/nsim

res_RI_MNAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "RI")
mu2_RI_MNAR <- mean(res_RI_MNAR$mu2ests)
sigma22_RI_MNAR <- mean(res_RI_MNAR$var2ests)
rho_RI_MNAR <- mean(res_RI_MNAR$corrests)
coverage_RI_MNAR <- sum((res_RI_MNAR$lls <= mu2) & (res_RI_MNAR$uls >= mu2))/nsim
```

```r
res_SRI_MNAR <- single_imp_sim(y1 = y1, y2 = y2, r = r, method = "SRI")
mu2_SRI_MNAR <- mean(res_SRI_MNAR$mu2ests)
sigma22_SRI_MNAR <- mean(res_SRI_MNAR$var2ests)
rho_SRI_MNAR <- mean(res_SRI_MNAR$corrests)
coverage_SRI_MNAR <- sum((res_SRI_MNAR$lls <= mu2) & (res_SRI_MNAR$uls >= mu2))/nsim

df <- data.frame("Mu2" = c(mu2_CCA_MNAR, mu2_MI_MNAR, mu2_RI_MNAR, mu2_SRI_MNAR),
                 "Sigma22" = c(sigma22_CCA_MNAR, sigma22_MI_MNAR,
                               sigma22_RI_MNAR, sigma22_SRI_MNAR),
                 "Rho" = c(rho_CCA_MNAR, rho_MI_MNAR,
                           rho_RI_MNAR, rho_SRI_MNAR),
                 "Cov" = c(coverage_CCA_MNAR,coverage_MI_MNAR,coverage_RI_MNAR,
                           coverage_SRI_MNAR))
rownames(df) <- c("CCA", "MI", "RI", "SRI")
colnames(df) = c("$\\mu_2$", "$\\sigma_2^2$", "$\\rho$", "Coverage")

knitr::kable(df, escape = FALSE, digits = 4, caption = "MNAR data")
```

Table 3: MNAR data

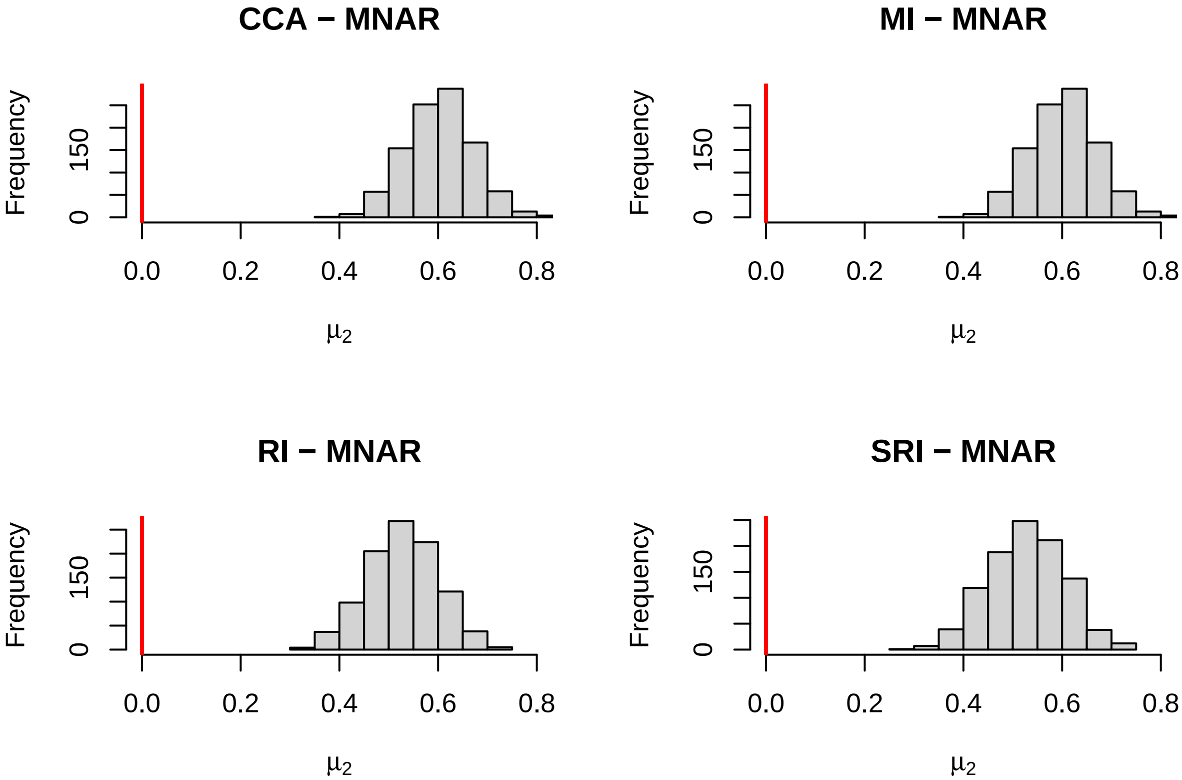
	μ_2	σ_2^2	ρ	Coverage
CCA	0.6030	0.5212	0.1626	0
MI	0.6030	0.3025	0.1006	0
RI	0.5286	0.3204	0.2531	0
SRI	0.5289	0.5337	0.1963	0

```r
par(mfrow = c(2,2))
hist(res_CCA_MNAR$mu2ests, xlab = expression(mu[2]), main = "CCA - MNAR",
     xlim = c(0,0.8))
abline(v = mu2, col = "red", lwd = 2)

hist(res_MI_MNAR$mu2ests, xlab = expression(mu[2]), main = "MI - MNAR",
     xlim = c(0,0.8))
abline(v = mu2, col = "red", lwd = 2)

hist(res_RI_MNAR$mu2ests, xlab = expression(mu[2]), main = "RI - MNAR",
     xlim = c(0,0.8))
abline(v = mu2, col = "red", lwd = 2)

hist(res_SRI_MNAR$mu2ests, xlab = expression(mu[2]), main = "SRI - MNAR",
     xlim = c(0,0.8))
abline(v = mu2, col = "red", lwd = 2)
```

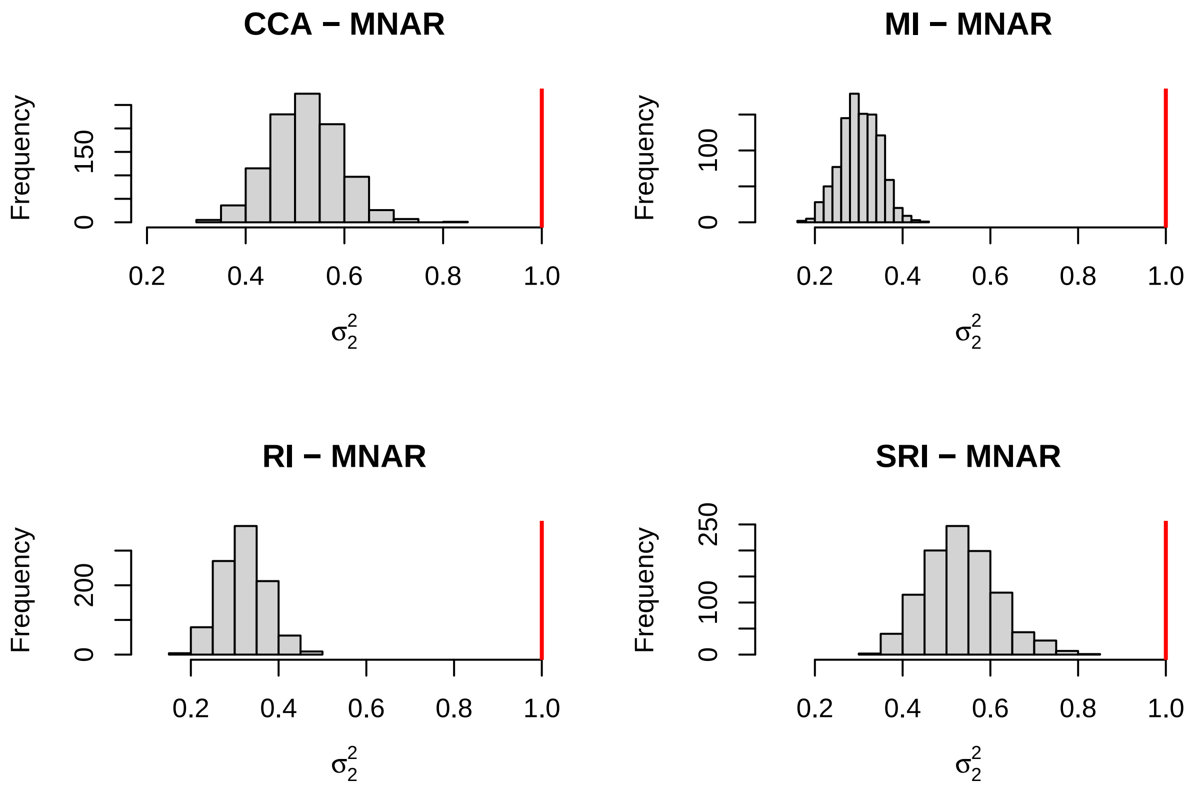


```
par(mfrow = c(2,2))
hist(res_CCA_MNAR$var2ests, xlab = expression(sigma[2]^2), main = "CCA - MNAR",
     xlim = c(0.2, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_MI_MNAR$var2ests, xlab = expression(sigma[2]^2), main = "MI - MNAR",
      xlim = c(0.1, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_RI_MNAR$var2ests, xlab = expression(sigma[2]^2), main = "RI - MNAR",
     xlim = c(0.1, 1))
abline(v = sigma22, col = "red", lwd = 2)

hist(res_SRI_MNAR$var2ests, xlab = expression(sigma[2]^2), main = "SRI - MNAR",
     xlim = c(0.1, 1))
abline(v = sigma22, col = "red", lwd = 2)
```

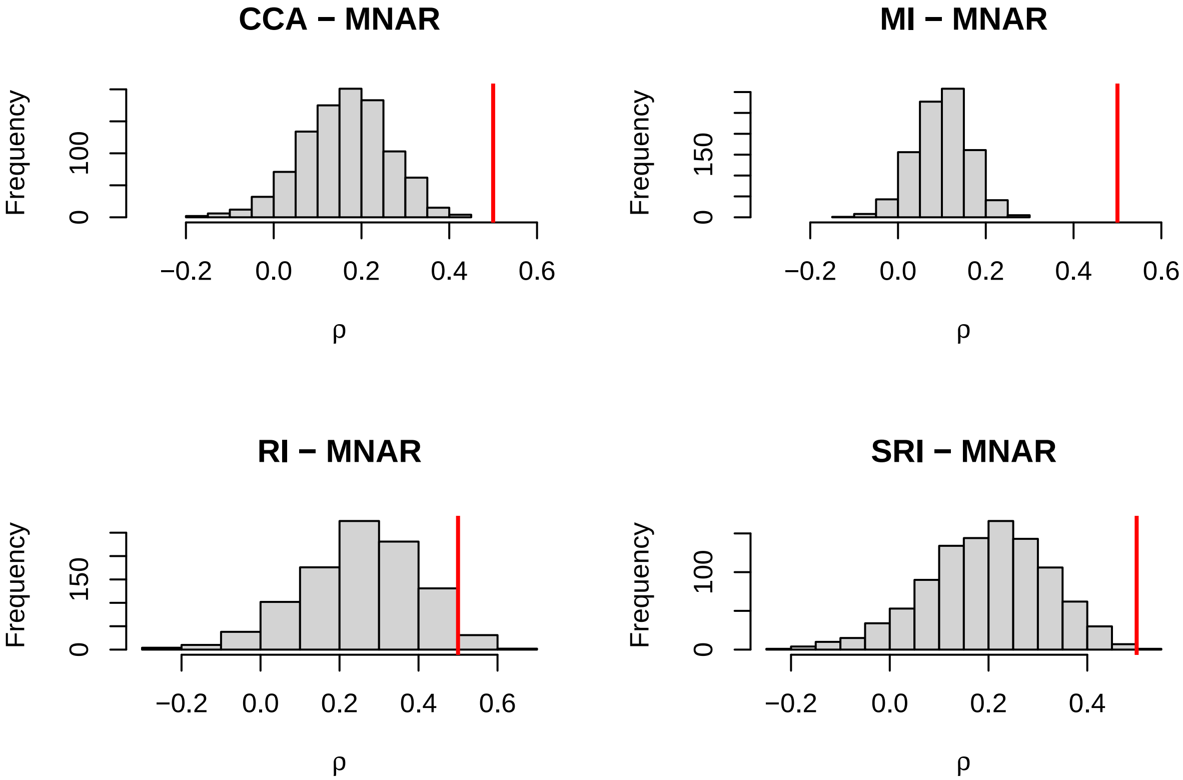


```
par(mfrow = c(2,2))
hist(res_CCA_MNAR$corrests, xlab = expression(rho), main = "CCA - MNAR",
     xlim = c(-0.3, 0.6))
abline(v = rho, col = "red", lwd = 2)

hist(res_MI_MNAR$corrests, xlab = expression(rho), main = "MI - MNAR",
     xlim = c(-0.3, 0.6))
abline(v = rho, col = "red", lwd = 2)

hist(res_RI_MNAR$corrests, xlab = expression(rho), main = "RI - MNAR")
abline(v = rho, col = "red", lwd = 2)

hist(res_SRI_MNAR$corrests, xlab = expression(rho), main = "SRI - MNAR")
abline(v = rho, col = "red", lwd = 2)
```



As we can see, under MNAR, all methods provide biased results of all quantities. MNAR data are hard to handle!

As a final note I remark that the way to deal with missing data depends on the final aim. If the aim is to do inference, estimating parameters and their variances as well as possible, then single imputation is clearly inadequate.