FOUNDATIONS OF STATISTICAL LEARNING

HOMEWORK 01: MONTE CARLO SIMULATIONS IN R ON MULTIPLE LINEAR REGRESSION

Mohamed Salah Jebali

Matricola: 7078487

School of Engineering - Master of Science degree in Artificial Intelligence

Università degli Studi di Firenze

March 2022

Indice

1 Introduction

In this homework for the *Foundations of Statistical Learning* course we want to set up a Monte Carlo experiment to study the consequences of the omission of a relevant covariate from a multiple linear regression model.

1.1 Setup: Data Generating Process and Scenarios

The data generating process (or DGP) is the following:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \epsilon_i \tag{1}$$

$$\epsilon_i \sim N(0,1) \tag{2}$$

$$X_1, X_2 \sim N_2(0, \Sigma) \tag{3}$$

We consider two different scenarios. Scenario 1 with

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

and Scenario 2 with

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2 Monte Carlo simulations

For each scenarios, I've conducted nsim simulations where, firstly, I've predicted the values of Y with a reduce linear model depending only on X_1 and saved the values of β_1 and, secondly, I've made the same thing with the complete linear model. Then, I've compared the outcomes.

I've made several combinations, with different values of the parameters and different numbers of iterations.

2.1 Simulation 1

In this simulation I've set the parameters to values different from zero as we can see in next chunk of R code:

```
## HOMEWORK 01
## FOUNDATIONS OF STATISTICAL LEARNIG - JEBALI 7078487

## GLOBAL SETTING ##
n <- 100 # sampling size
beta0 <- 5
beta1 <- 0.5
beta2 <- 0.5
```

Scenario 1 In this scenario the random variables X_1 and X_2 are dependent and positive correlated, as we can see in the next R code chunk:

```
## SCENARIO 1 SETTING ##

# In this scenario the RVs X1 and X2 are dependent
# and positive correlated
sigma1 <- matrix(c(1, 0.5, 0.5, 1), nrow = 2, ncol = 2)
mu1 <- c(0,0)
```

I've conducted 1000 iterations for this Monte Carlo simulation:

```r
# SIMULATION SETTING #
nsim <- 1000 # number of simulations

beta00.est <- numeric(nsim) # vector of intercept estimations of mod0
beta01.est <- numeric(nsim) # vector of beta1 estimations of mod0

beta10.est <- numeric(nsim) # vector of intercept estimations of mod1
beta11.est <- numeric(nsim) # vector of beta1 estimations of mod1
beta12.est <- numeric(nsim) # vector of beta2 estimations of mod1

for (i in 1:nsim){
  set.seed(123 + i)
  X1 <- mvtnorm::rmvnorm(n, mean = mu1, sigma =sigma1)
  X2 <- mvtnorm::rmvnorm(n, mean = mu1, sigma =sigma1)
  e <- rnorm(n,0,1)

  y <- beta0 + beta1*X1+beta2*X2 + e

  mod0 <- lm (y ~ X1) # reduced model
  mod1 <- lm (y ~ X1 + X2) # complete model

  # Parameters estimations from the reduced model
  beta00.est[[i]] <- as.vector(mod0$coeff[1])
  beta01.est[[i]] <- as.vector(mod0$coeff[2])

  # Parameters estimations from the complete model
  beta01.est[[i]] <- as.vector(mod1$coeff[1])
  beta11.est[[i]] <- as.vector(mod1$coeff[2])
  beta12.est[[i]] <- as.vector(mod1$coeff[3])
}
```

We can see that, the estimation of β_1 with the reduced model is **wrong** and similar to the intercept value, while the estimation with the complete model is quite near to the **true** value.

```r
# Mean of the estimated values of beta1 from the reduced model
mean(beta01.est)

## [1] 5.001572

# Mean of the estimated values of beta1 from the complete model
mean(beta11.est)

## [1] 0.4949942
```

Both the estimated values of β_1 seem to have a Normal distribution, as we can see in the plot at Figure 1 and Figure 2, respectively for the reduced and complete model.

```r
# Plot of the estimated values of beta1 from the reduced model
hist(beta01.est, col="#ff6600", prob=TRUE)
lines(density(beta01.est), lty="longdash", col="#6666ff")
# Plot of the estimaed values of beta1 from the complete model
hist(beta11.est, col="#33cc33", prob=TRUE)
lines(density(beta11.est), lty="longdash", col="#ff9900")
```

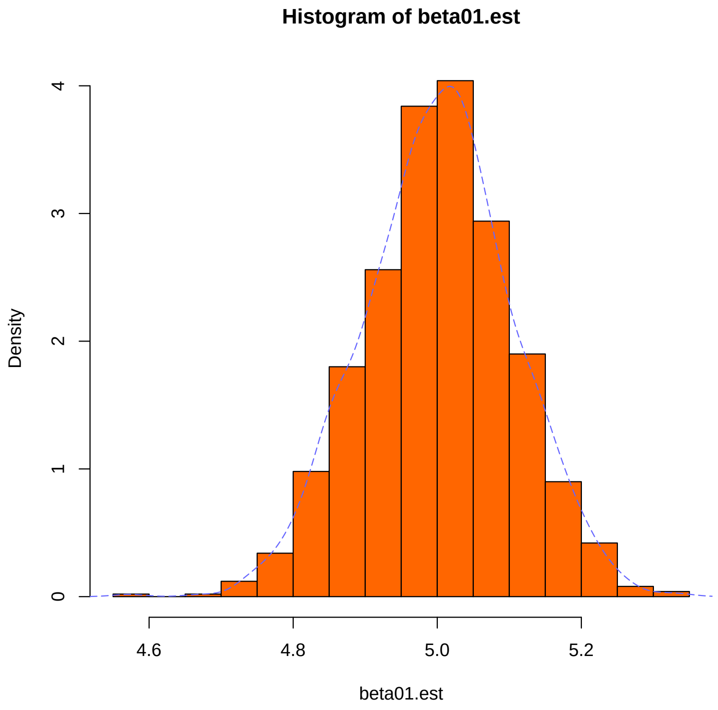

Figura 1: Histogram plot of the estimated values of beta1 from the reduced model

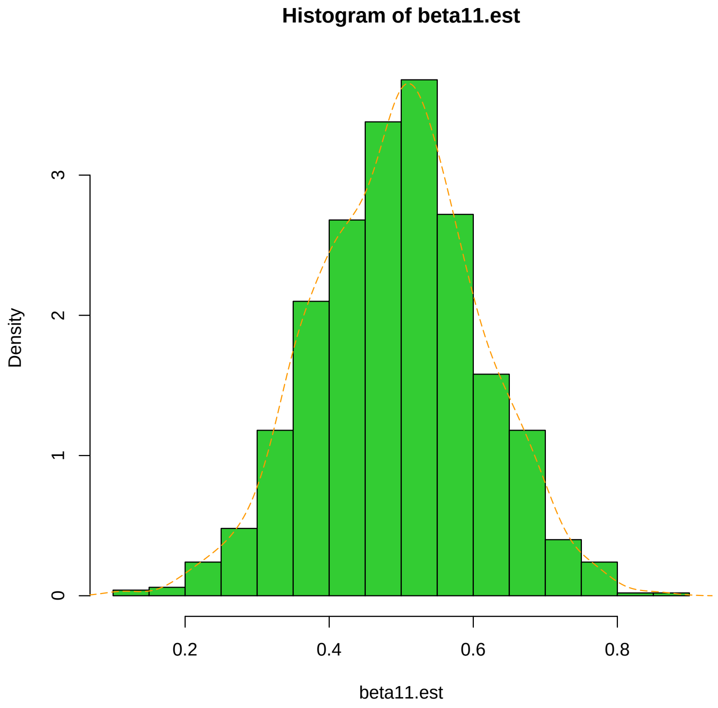

Figura 2: Histogram plot of the estimated values of beta1 from the complete model

Scenario 2 In this scenario the random variables X_1 and X_2 are independent, as can see in the next R code chunk:

```
## SCENARIO 2 SETTING ##

# In this scenario the RVs X1 and X2 are independent #
sigma2 <- matrix(c(1, 0, 0, 1), nrow = 2, ncol = 2)
mu2 <- c(0,0)
```

I've conducted 1000 iterations for this Monte Carlo simulation:

```
# SIMULATION SETTING #
nsim <- 1000 # number of simulations

ind_beta00.est <- numeric(nsim) # vector of intercept estimations of ind_mod0
ind_beta01.est <- numeric(nsim) # vector of beta1 estimations of ind_mod0

ind_beta10.est <- numeric(nsim) # vector of intercept estimations of ind_mod1
ind_beta11.est <- numeric(nsim) # vector of beta1 estimations of ind_mod1
ind_beta12.est <- numeric(nsim) # vector of beta2 estimations of ind_mod1

for (i in 1:nsim){
  set.seed(123 + i)
  ind_X1 <- mvtnorm::rmvnorm(n, mean = mu2, sigma = sigma2)
  ind_X2 <- mvtnorm::rmvnorm(n, mean = mu2, sigma = sigma2)
  e <- rnorm(n,0,1)

  ind_y <- beta0 + beta1*ind_X1+beta2*ind_X2 + e

  ind_mod0 <- lm (ind_y ~ ind_X1) # reduced model
  ind_mod1 <- lm (ind_y ~ ind_X1 + ind_X2) # complete model

  # Parameters estimations from the reduced model
  ind_beta00.est[[i]] <- as.vector(ind_mod0$coeff[1])
  ind_beta01.est[[i]] <- as.vector(ind_mod0$coeff[2])

  # Parameters estimations from the complete model
  ind_beta01.est[[i]] <- as.vector(ind_mod1$coeff[1])
  ind_beta11.est[[i]] <- as.vector(ind_mod1$coeff[2])
  ind_beta12.est[[i]] <- as.vector(ind_mod1$coeff[3])
}
```

As we've observed in the previous scenario, the estimation of β_1 using the reduced model resulted to be **wrong** and similar to the intercept value, while the estimation with the complete model resulted to be near to the **true** value of the parameter.

```
# Mean of the estimated values of beta1 from the reduced model
mean(ind_beta01.est)

## [1] 5.001572

# Mean of the estimated values of beta1 from the complete model
mean(ind_beta11.est)

## [1] 0.495738
```

Both the estimated values of β_1 seem to have a Normal distribution, as we can see in the plot at Figure 3 and Figure 4, respectively for the reduced and complete model.

```
# Plot of the estimated values of beta1 from the reduced model
hist(ind_beta01.est, col="#ff6600", prob=TRUE)
lines(density(ind_beta01.est), lty="longdash", col="#6666ff")
# Plot of the estimaed values of beta1 from the complete model
hist(ind_beta11.est, col="#33cc33", prob=TRUE)
lines(density(ind_beta11.est), lty="longdash", col="#ff9900")
```

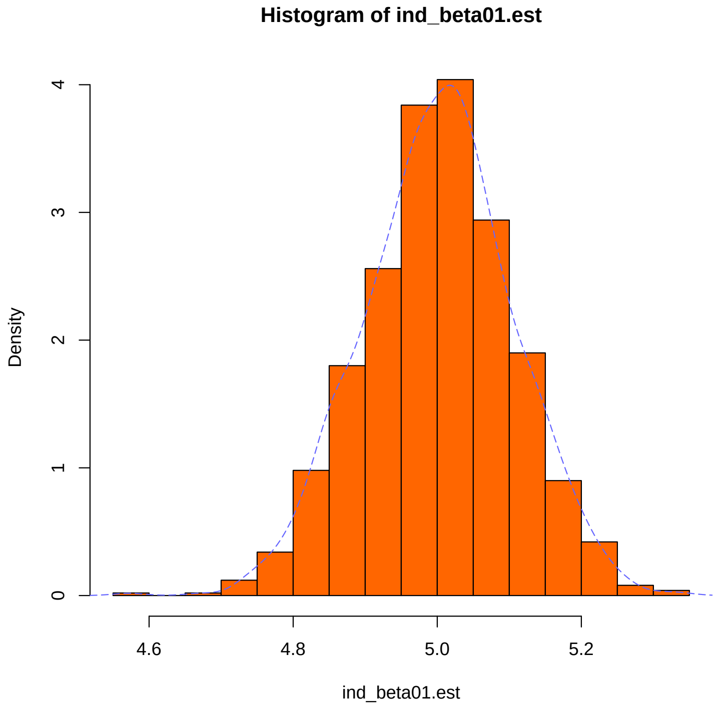

Figura 3: Histogram plot of the estimated values of beta1 from the reduced model

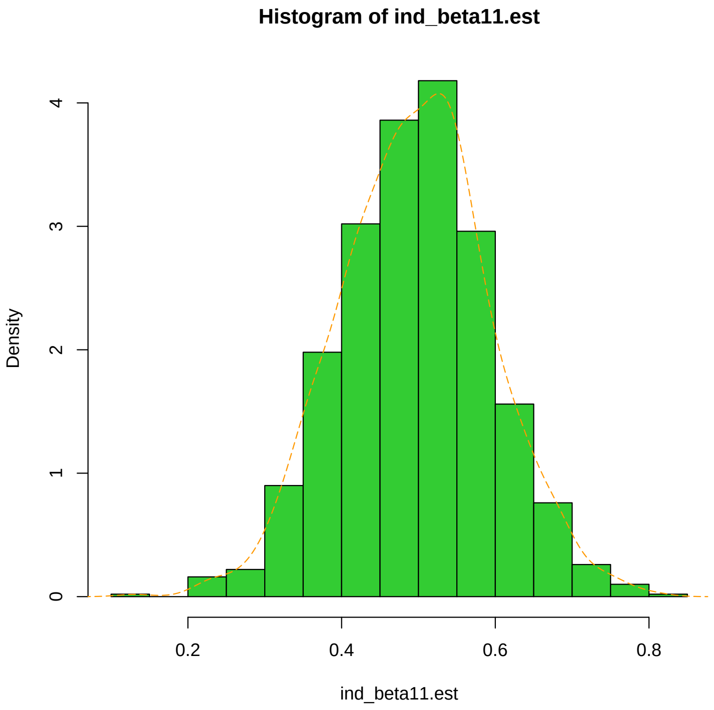

Figura 4: Histogram plot of the estimated values of beta1 from the complete model

2.2 Simulation 2

In this simulation set, I've set the parameters all to be zero and kept the number of iterations equal to 1000. For practical reasons the chunk code won't be displayed, just the results will be showed. To repeat the experiments just use the code showed above and set the values of beta to zero.

```
## GLOBAL SETTING ##
n <- 100 # sampling size
beta0 <- 0
beta1 <- 0
beta2 <- 0
```

Scenario 1 In both cases the value of β_1 seems to be the true one, but we should pay attention to the fact the the intercept value is also zero, so in reality, the estimation of β_1 in the reduced model assumes the value of the intercept, as we've seen in Simulation 1.

```
# Mean of the estimated values of beta1 from the reduced model
mean(beta01.est)

## [1] 0.001572026

# Mean of the estimated values of beta1 from the complete model
mean(beta11.est)

## [1] -0.005005789
```

Both the estimated values of β_1 seem to have a Normal distribution, as we can see in the plot at Figure 5 and Figure 6, respectively for the reduced and complete model.

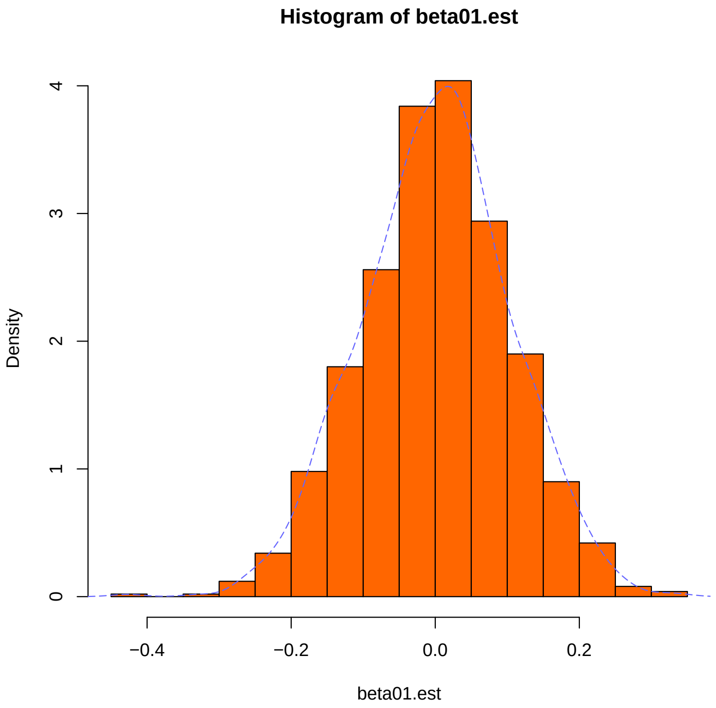

Figura 5: Histogram plot of the estimated values of beta1 from the reduced model

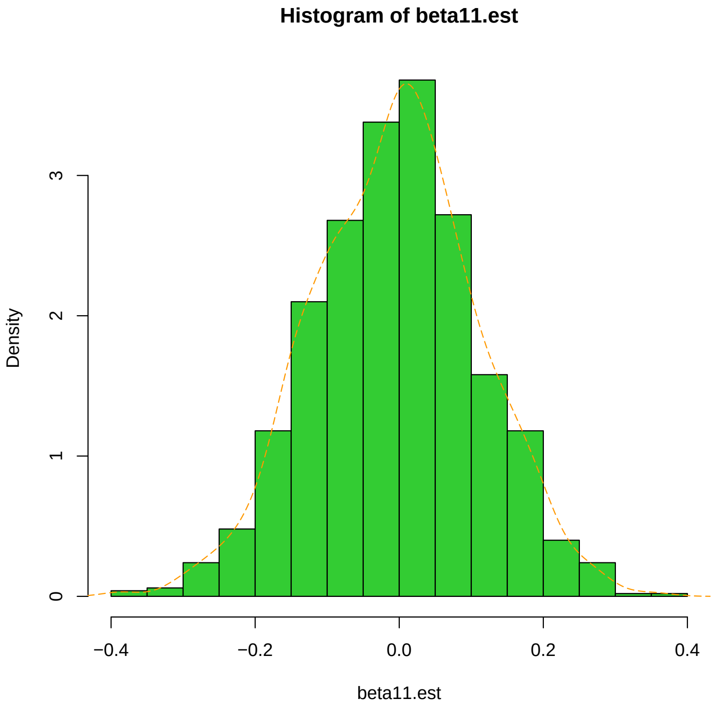

Figura 6: Histogram plot of the estimated values of beta1 from the complete model

Scenario 2 In the second scenario we've obtained the sames results that we got in the first scenario.

2.3 Simulation 3

In this simulation set, I've tried to reduce the number of iterations and set it to 10. I've set also intercept equal to 5, β_1 equal to 1 and β_2 equal to zero. For practical reasons I won't display the chunk of code but only the obtained results for the estimations and hist plots of the estimated values.

Scenario 1 In this scenario setting, we've obtained a similar behaviour that we got in the previous simulations: the value of β_1 estimated from the reduced model is wrong and similiar to the value of the intercept, while the value estimated from the complete one seems to be near to the true value despite two important aspects: we set only 10 iterations and we set β_2 equal to zero.

```
# Mean of the estimated values of beta1 from the reduced model
mean(beta01.est)

## [1] 4.926051

# Mean of the estimated values of beta1 from the complete model
mean(beta11.est)

## [1] 1.032991
```

As we could have forseen, the estimated values of β_1 don't have a Normal distribution, because of the low number of simulations we've set. Is is possible to take a look to the hist plot at Figure 7 and Figure 8.

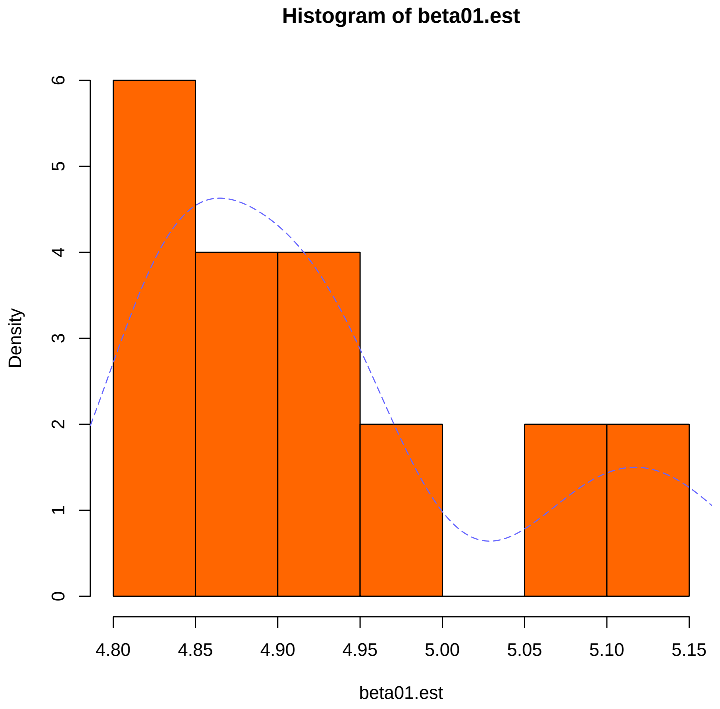

Figura 7: Histogram plot of the estimated values of beta1 from the reduced model

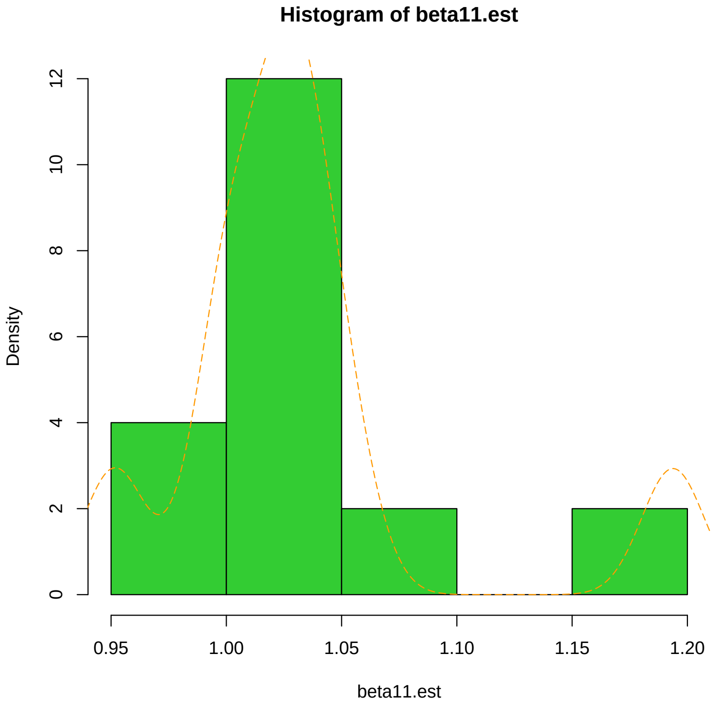

Figura 8: Histogram plot of the estimated values of beta1 from the complete model

3 Conclusion

The Monte Carlo experiments I've conducted led to a conclusion that I didn't expect: despite the dependence or indepedence between the variables of the model, the estimation of the values of β_1 doesn't work when we omit a relevant covariate from a multiple linear regression model, while it seems to work when we consider the complete model. I expected this behaviour for the model with dependent random variables, while I didn't for the one with independent random variables. Moreover, as we could expect, under a limit number of iterations, the estimated values of β_1 doesn't follow the Normal distribution, as we could have seen with nsim equal to 10.