University of Edinburgh, School of Mathematics

Incomplete Data Analysis, 2020/2021

Workshop 2 – Solutions

Vanda Inácio

1. (a) The overall vitamin D mean provided by a complete case analysis is 16.11 and the associated standard error is 6.243.

```
load("dataex1.Rdata")
ind <- which(is.na(dataex1$VitD) == FALSE)
mccoverall <- mean(dataex1$VitD, na.rm = TRUE)
seccoverall <- sd(dataex1$VitD, na.rm = TRUE)/sqrt(length(ind))
mccoverall; seccoverall
```

```
## [1] 16.11
```

```
## [1] 6.243066
```

Stratifying by sex, we obtain a mean vitamin D of 13.75 (10.157) for females and mean vitamin D of 17.68 (8.627) for males.

```
#cc analysis women
indf <- which(is.na(dataex1$VitD) == FALSE & dataex1$Sex == "Female")
mccf <- mean(dataex1$VitD[indf])
seccf <- sd(dataex1$VitD[indf])/sqrt(length(indf))
mccf; seccf
```

```
## [1] 13.75
```

```
## [1] 10.15747
```

```
#cc analysis men
indm <- which(is.na(dataex1$VitD) == FALSE & dataex1$Sex == "Male")
mccm <- mean(dataex1$VitD[indm])
seccm <- sd(dataex1$VitD[indm])/sqrt(length(indm))
mccm; seccm
```

```
## [1] 17.68333
```

```
## [1] 8.626719
```

(b) The mean vitamin D using mean imputation is 16.11. Obviously, this value is equal to the one obtained in the overall complete case analysis (after mean imputation, the mean of the dataset remains unchanged). The associated standard error is 4.742, which is smaller than the one provided by an overall complete case analysis (in mean imputation, each imputed value contributes with a zero value to the numerator, but the denominator is inflated, when compared to the complete cases analysis).

```
vitdmi <- ifelse(is.na(dataex1$VitD) == TRUE, mean(dataex1$VitD, na.rm = TRUE), dataex1$VitD)
mmi <- mean(vitdmi)
n <- nrow(dataex1)
```

```r
semi <- sd(vitdmi)/sqrt(n)
mmi; semi
```

```
## [1] 16.11
```

```
## [1] 4.741949
```

(c) We will fit a linear regression model to the complete cases, using vitamin D as the response and sex and age as the predictors, that is,

$$\text{VitD} = \beta_0 + \beta_1 \text{Age} + \beta_2 \text{Sex} + \varepsilon, \qquad \varepsilon \sim \text{N}(0, \sigma^2),$$

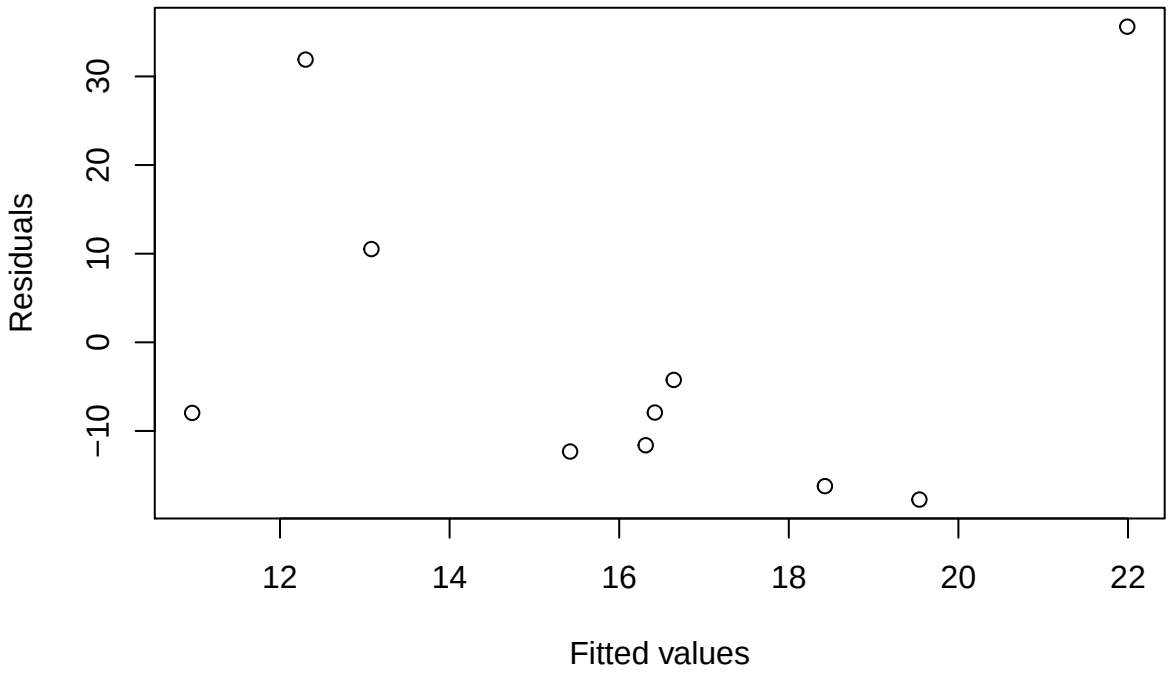
where Sex is a binary variable taking the value 0 if the subject is female and the value 1 if the subject is male. The estimated regression coefficients are $\widehat{\beta}_0 = 25.0033$, $\widehat{\beta}_1 = -0.2228$, and $\widehat{\beta}_2 = 5.0104$.

```r
fitvitd <- lm(VitD ~ Age + Sex, data = dataex1)
summary(fitvitd)
```

```
##
## Call:
## lm(formula = VitD ~ Age + Sex, data = dataex1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.740 -12.144  -7.943   6.831  35.608
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.0033    32.5537   0.768    0.468
## Age          -0.2228     0.6065  -0.367    0.724
## SexMale       5.0104    14.5353   0.345    0.740
##
## Residual standard error: 22.06 on 7 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.0293,    Adjusted R-squared:  -0.248
## F-statistic: 0.1057 on 2 and 7 DF,  p-value: 0.9011
```

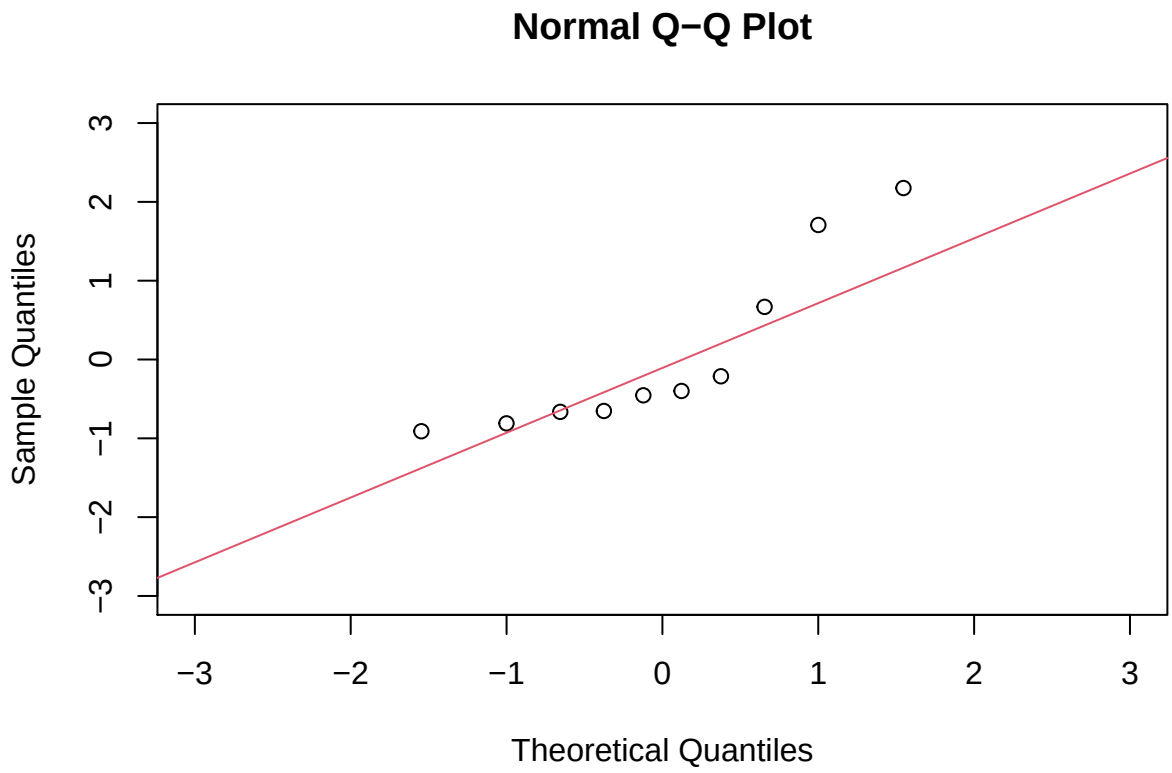
ALthough this is an artificial small dataset we need to (informally) check the validity of model's assumptions. We can look at the plot of the residuals to informally check the assumption of linearity (and homoscedasticity).

```r
plot(fitvitd$fitted.values, residuals(fitvitd), xlab = "Fitted values", ylab = "Residuals")
```



We can also check the QQ-plot.

```
qqnorm(rstandard(fitvitd), xlim = c(-3,3), ylim = c(-3,3))
qqline(rstandard(fitvitd),col=2)
```



The predicted values are

Subject id	Age	Sex	Imputed vitamin D
4	59	Male	16.86626
7	45	Male	19.98600
10	38	Female	16.53548

```
predri <- predict(fitvitd,newdata=dataex1)
predri[4]; predri[7]; predri[10]
```

```
##        4
## 16.86626
```

```
##      7
## 19.986
```

```
##       10
## 16.53548
```

The estimated mean vitamin D is then 16.50 with associated standard error of 4.751.

```
vitdri <- ifelse(is.na(dataex1$VitD) == TRUE, predri, dataex1$VitD)
mri <- mean(vitdri)
seri <- sd(vitdri)/sqrt(n)
mri; seri
```

```
## [1] 16.49906
```

```
## [1] 4.751273
```

(d) We do a similar analysis to the one in (c) but we now add a random noise to the predictions, i.e.,

$$\widehat{\text{VitD}} = \widehat{\beta}_0 + \widehat{\beta}_1 \text{Age} + \widehat{\beta}_2 \text{Gender} + z, \quad z \sim \text{N}(0, \widehat{\sigma}^2), \quad \widehat{\sigma} = 22.06.$$

Note that in this case, depending on the seed, we can obtain negative predicted values, which of course, does not make any sense in practice. We have obtained a mean vitamin D of 19.514 with an associated standard error of 5.606.

```
set.seed(1)
predsri <- predict(fitvitd, newdata = dataex1) + rnorm(n, 0, summary(fitvitd)$sigma)
predsri[4]; predsri[7]; predsri[10]
```

```
##        4
## 52.05056
```

```
##        7
## 30.73636
```

```
##       10
## 9.800066
```

```
vitdsri <- ifelse(is.na(dataex1$VitD) == TRUE, predsri, dataex1$VitD)

msri <- mean(vitdsri)
sesri <- sd(vitdsri)/sqrt(n)
msri; sesri
```

```
## [1] 19.51438
```

```
## [1] 5.60599
```

(e) Using the proposed stratification results in 4 groups: females whose age is ≤ 50, females whose age is > 50, males whose age is ≤ 50, and males whose age is > 50. For instance, for subject 4, a male aged 59, there are 4 possible donors, which are subjects 2, 3, 8, and 12. We randomly pick one out of these four possible donors to donate his vitamin D value to subject 4.

```
indhdm4 <- which(is.na(dataex1$VitD) == FALSE & dataex1$Sex == "Male" & dataex1$Age > 50)
indhdm4
```

```
## [1]  2  3  8 12
```

```
donor4 <- sample(indhdm4, 1, replace = TRUE)
donor4
```

```
## [1] 3
```

```
indhdm7 <- which(is.na(dataex1$VitD) == FALSE & dataex1$Sex == "Male" & dataex1$Age <= 50)
indhdm7
```

```
## [1]  1 13
```

```
donor7 <- sample(indhdm7, 1, replace = TRUE)
donor7
```

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

```
indhdf10 <- which(is.na(dataex1$VitD) == FALSE & dataex1$Sex == "Female" & dataex1$Age <= 50)
indhdf10
```

```
## [1] 5 6
```

```
donor10 <- sample(indhdf10, 1, replace = TRUE)
donor10
```

```
## [1] 5
```

```
vitdhd <- c(dataex1$VitD[is.na(dataex1$VitD) == FALSE],dataex1$VitD[donor4],
           dataex1$VitD[donor7], dataex1$VitD[donor10])
mhd <- mean(vitdhd); sehd <- sd(vitdhd)/sqrt(n)
mhd; sehd
```

```
## [1] 12.93846
```

```
## [1] 5.028503
```

2. With $\mathbf{X} = (X_1, X_2)$, (i) corresponds to the case where the probability of observing a complete case depends only on $\mathbf{X}$, (ii) corresponds to the case where the probability of observing a complete case depends on $\mathbf{X}$ and Y, and (iii) to the case where the probability of observing a complete case does not depend on $\mathbf{X}$ or Y. Under case (i), we expect the complete case least squares estimator to be consistent. Under (ii), we expect it to be possibly inconsistent, and under (iii) we expect it to be consistent.

Let us clean the workspace, define S, n, β_0, and simulate the data.

```
rm(list = ls())

S <- 1000; n <- 200; beta0 <- c(20,5,-5)

set.seed(1)
ysim <- x1sim <- x2sim <- matrix(0, nrow = n, ncol = S)
for(l in 1:S){
x1sim[,l] <- rnorm(n, 10, 3)
x2sim[,l] <- rbinom(n, 1, 0.4)
ysim[,l] <- beta0[1] + beta0[2]*x1sim[,l] + beta0[3]*x2sim[,l] + rnorm(n, 0, 8)
}
```

I created a general function, so for each of the three cases, we only need to change the inputs of the function.

```
res_function <- function(x1, x2, y, psi){

n <- nrow(x1); S <- ncol(x1)

c <- matrix(0, nrow = n, ncol = S)
for(l in 1:S){
U <-psi[1] + psi[2]*x1sim[,l] + psi[3]*x2sim[,l] + psi[4]*ysim[,l]+
   psi[5]*ysim[,l]*x1sim[,l] + psi[6]*ysim[,l]*x2sim[,l]

prob <- exp(U)/(exp(U)+1)

c[,l] <- rbinom(n, 1, prob)
}

#complete case estimates
estimatescc <- matrix(0, nrow = 3, ncol = S)
for(l in 1:S){
estimatescc[,l] <- lm(ysim[c[, l] == 1, l] ~ x1sim[c[, l] == 1, l] +
                      x2sim[c[, l] == 1, l])$coefficients
}

#full data (no missingness) estimates
estimatesfull <- matrix(0, nrow = 3, ncol = S)
for(l in 1:S){
estimatesfull[,l] <- lm(ysim[,l] ~ x1sim[,l] + x2sim[,l])$coefficients
}

#computing mean and sd of estimates across the S simulated data sets
meanestimatescc <- apply(estimatescc, 1, mean)
sdestimatescc <- apply(estimatescc, 1, sd)

meanestimatesfull <- apply(estimatesfull, 1, mean)
sdestimatesfull <- apply(estimatesfull, 1, sd)

return(list(meanestimatescc, sdestimatescc, meanestimatesfull, sdestimatesfull))
}
```

Now, I will apply the function to case (i).

```
psi <- c(2, -0.025, 0.5, 0, 0, 0)
resc1 <- res_function(x1 = x1sim, x2 = x2sim, y = ysim, psi = psi)

biascc1 <- resc1[[1]] - beta0
biasfull1 <- resc1[[3]] - beta0
sdcc1 <- resc1[[2]]
sdfull1 <- resc1[[4]]

df1 <- data.frame("MCmeans" = c(resc1[[1]], resc1[[3]]),
                  "bias" = c(biascc1, biasfull1),
                  "sd" = c(sdcc1, sdfull1))

rownames(df1) <- c("CCA $\\beta_1$", "CCA $\\beta_2$","CCA $\\beta_3$",
"Complete data $\\beta_1$", "Complete data $\\beta_2$", "Complete data $\\beta_3$")
```

```r
colnames(df1) <- c("Monte Carlo means", "Bias", "Standard deviation")

knitr::kable(df1, escape = FALSE, digits = 4, caption = "Case (i)")
```

Table 1: Case (i)

	Monte Carlo means	Bias	Standard deviation
CCA β_1	20.0054	0.0054	2.1831
CCA β_2	5.0008	0.0008	0.2036
CCA β_3	-5.0370	-0.0370	1.1810
Complete data β_1	20.0146	0.0146	2.0098
Complete data β_2	4.9993	-0.0007	0.1865
Complete data β_3	-5.0458	-0.0458	1.1233

From the results, we can see that the Monte Carlo bias is very close to zero for both the complete case and full data estimates. As expected, the standard deviation across the 1000 estimates is slightly higher for the complete case estimates, as we already know that inferences under a complete case analysis, even if valid (as in this case), they are less efficient since they are based on a smaller subset of data.

Let us now think for a moment about what can be the missing data mechanism implied by case (i). We know that each of Y and $\mathbf{X}$ is either observed or missing. There are four possible patterns: $(R_Y, R_X) = (1, 1)$ where both Y and $\mathbf{X}$ are observed, $(R_Y, R_X) = (1, 0)$ corresponding to the case where Y is observed and $\mathbf{X}$ is missing, $(R_Y, R_X) = (0, 1)$ corresponding to the case where Y is missing and $\mathbf{X}$ is observed, and $(R_Y, R_X) = (0, 0)$ corresponding to the case where both Y and $\mathbf{X}$ are missing. In the case of (i), we have that the probability of observing a complete case depends only on $\mathbf{X}$, i.e., $\Pr\{(R_Y, R_X) = (1, 1) \mid Y, \mathbf{X}\} = \Pr\{(R_Y, R_X) = (1, 1) \mid \mathbf{X}\}$. Note that this condition does not relate to which variables (response or covariates) have missing values. Let us suppose first that $\mathbf{X}$ can be missing but Y is always observed. Thus,

$$\Pr\{(R_Y, R_X) = (0, 1) \mid \mathbf{X}\} = \Pr\{(R_Y, R_X) = (0, 0) \mid \mathbf{X}\} = 0,$$

because Y is always observed. Also, because we must have

$$\Pr\{(R_Y, R_X) = (0, 1) \mid \mathbf{X}\} + \Pr\{(R_Y, R_X) = (0, 0) \mid \mathbf{X}\} + \Pr\{(R_Y, R_X) = (1, 1) \mid \mathbf{X}\} + \Pr\{(R_Y, R_X) = (1, 0) \mid \mathbf{X}\} = 1,$$

we conclude that $\Pr\{(R_Y, R_X) = (1, 0) \mid \mathbf{X}\} = 1 - \Pr\{(R_Y, R_X) = (1, 1) \mid \mathbf{X}\}$ depends on $\mathbf{X}$. Therefore, the probability of observing the missingness pattern where Y is observed but $\mathbf{X}$ is missing depends on the unobserved $\mathbf{X}$, and thus the missingness mechanism is MNAR. Suppose now, conversely, that Y can be missing but $\mathbf{X}$ is always observed. We have

$$\Pr\{(R_Y, R_X) = (1, 0) \mid \mathbf{X}\} = \Pr\{(R_Y, R_X) = (0, 0) \mid \mathbf{X}\} = 0,$$

because $\mathbf{X}$ is always observed. Hence, $\Pr\{(R_Y, R_X) = (0, 1) \mid \mathbf{X}\} = 1 - \Pr\{(R_Y, R_X) = (1, 1) \mid \mathbf{X}\}$ depends on $\mathbf{X}$. Here, the probability of observing the missingness pattern where $\mathbf{X}$ is observed but Y is missing depends only on $\mathbf{X}$, which is itself always observed. Thus, the missingness mechanism is MAR.

In conclusion, regardless of whether or not the missingness mechanism is MAR or MNAR, the complete case ordinary least squares estimator will be consistent if $\Pr\{(R_Y, R_X) = (1, 1) \mid Y, \mathbf{X}\}$ depends only on $\mathbf{X}$.

Apart: Let us look at the justification for complete case validity. Remember that our assumption for missingness is that $\Pr\{(R_Y, R_X) = (1, 1) \mid Y, \mathbf{X}\} = \Pr\{(R_Y, R_X) = (1, 1) \mid \mathbf{X}\}$. A complete case analysis involves fitting the conditional model for $f(Y \mid \mathbf{X})$ in the subset of subjects with $(R_Y, R_X) = (1, 1)$:

$$
\begin{aligned}
f(Y \mid \mathbf{X}, (R_Y, R_X) = (1,1)) &= \frac{f(Y, \mathbf{X}, (R_Y, R_X) = (1,1))}{f(\mathbf{X}, (R_Y, R_X) = (1,1))} \\
&= \frac{\Pr\{(R_Y, R_X) = (1,1) \mid Y, \mathbf{X}\} f(\mathbf{X}, Y)}{\Pr\{(R_Y, R_X) = (1,1) \mid \mathbf{X}\} f(\mathbf{X})} \\
&= \frac{\Pr\{(R_Y, R_X) = (1,1) \mid \mathbf{X}\} f(\mathbf{X}, Y)}{\Pr\{(R_Y, R_X) = (1,1) \mid \mathbf{X}\} f(\mathbf{X})} \\
&= \frac{f(\mathbf{X}, Y)}{f(\mathbf{X})} \\
&= \frac{f(Y \mid \mathbf{X}) f(\mathbf{X})}{f(\mathbf{X})} \\
&= f(Y \mid \mathbf{X})
\end{aligned}
$$

Thus the conditional distribution $Y \mid \mathbf{X}$ in the complete cases is the same as in the complete data.

Moving to case (ii) now.

```
psi <- c(6, 0, 0, -0.075, -0.003, 0.05)
resc2 <- res_function(x1 = x1sim, x2 = x2sim, y = ysim, psi = psi)

biascc2 <- resc2[[1]] - beta0
biasfull2 <- resc2[[3]] - beta0
sdcc2 <- resc2[[2]]
sdfull2 <- resc2[[4]]

df2 <- data.frame("MCmeans" = c(resc2[[1]], resc2[[3]]),
                  "bias" = c(biascc2, biasfull2),
                  "sd" = c(sdcc2, sdfull2))

rownames(df2) <- c("CCA $\\beta_1$", "CCA $\\beta_2$","CCA $\\beta_3$",
"Complete data $\\beta_1$","Complete data $\\beta_2$", "Complete data $\\beta_3$")

colnames(df2) <- c("Monte Carlo means", "Bias", "Standard deviation")

knitr::kable(df2, escape = FALSE, digits = 4, caption = "Case (ii)")
```

Table 2: Case (ii)

	Monte Carlo means	Bias	Standard deviation
CCA β_1	19.8662	-0.1338	2.6016
CCA β_2	4.6726	-0.3274	0.3022
CCA β_3	-2.2287	2.7713	1.6836
Complete data β_1	20.0146	0.0146	2.0098
Complete data β_2	4.9993	-0.0007	0.1865
Complete data β_3	-5.0458	-0.0458	1.1233

In this case, where missingness depends on the response Y, we can appreciate that the Monte Carlo biases of the complete case estimates are far from zero (this is especially true for β_3). The results from the full data are the same as in (i) because the data are the same. And finally case (iii).

```
psi <- c(0.5, 0, 0, 0, 0, 0)
resc3 <- res_function(x1 = x1sim, x2 = x2sim, y = ysim, psi = psi)

biascc3 <- resc3[[1]] - beta0
```

```r
biasfull3 <- resc3[[3]] - beta0
sdcc3 <- resc3[[2]]
sdfull3 <- resc3[[4]]

df3 <- data.frame("MCmeans" = c(resc3[[1]], resc3[[3]]),
                  "bias" = c(biascc3, biasfull3),
                  "sd" = c(sdcc3, sdfull3))

rownames(df3) <- c("CCA $\\beta_1$", "CCA $\\beta_2$","CCA $\\beta_3$",
"Complete data $\\beta_1$","Complete data $\\beta_2$", "Complete data $\\beta_3$")

colnames(df3) <- c("Monte Carlo means", "Bias", "Standard deviation")

knitr::kable(df3, escape = FALSE, digits = 4, caption = "Case (iii)")
```

Table 3: Case (iii)

	Monte Carlo means	Bias	Standard deviation
CCA β_1	19.9408	-0.0592	2.5340
CCA β_2	5.0046	0.0046	0.2379
CCA β_3	-5.0338	-0.0338	1.4493
Complete data β_1	20.0146	0.0146	2.0098
Complete data β_2	4.9993	-0.0007	0.1865
Complete data β_3	-5.0458	-0.0458	1.1233

We can notice that the difference between the mean across estimates and the true value (i.e., the bias) is close to zero. Here $\Pr\{(R_Y, R_X) = (1,1) \mid Y, \mathbf{X}\}$ does not depend on $(Y, \mathbf{X})$. This implies that $\Pr\{(R_Y, R_X) = (1,1) \mid Y, \mathbf{X}\} = \Pr\{(R_Y, R_X) = (1,1)\}$ is a constant, which is of course true if the missingness mechanism is MCAR.

In conclusion, all simulations appear to bear out the result.