

# Assignment 1 - Incomplete Data Analysis

Zhe WANG

**Note:** The R code for Question 3 and Question 4 are available via the link (<https://github.com/ZheWANG331/IDA-Assignment-1.git>) to the repository of Github.

1.

- (a) The probability of ALQ being missing for those with ALQ=No is **(ii) 0.3**.

*Reason:* ALQ is MCAR, which means the missing of ALQ is completely at random and is unrelated to the value Yes/No.

- (b) ALQ being MAR given gender means so **(ii) The probability of ALQ being missing is independent of the Yes/No value of ALQ after adjusting for gender.**

*Reason:* ALQ is MAR given gender, which means the missing of ALQ depends only on gender information but it is further unrelated to the specific missing values Yes/No. Therefore, after adjusting for gender, the probability of ALQ being missing is independent of the value.

- (c) What is the probability of ALQ being missing for women? **(iii) It is impossible to conclude from the information given.**

*Reason:* As mentioned in (b), ALQ is MAR given gender, which means the missing of ALQ depends only on gender, but by the given information, we cannot get the certain probability of missing in each gender.

2. The largest possible subsample under a complete case analysis is **90**, the smallest is **0**.

*Reason:* Complete case analysis is to exclude the data for any case/individual that has one or more missing values and get a subsample.

The largest possible subsample is the case that all the missing values of 10 variables occur in the same 10 objects and we only need to delete these 10;

the smallest possible subsample is the case that all the missing values of 10 variables occur in different objects, which means each object has exactly 1 missing value, so we have to delete all of them, so 0 in subsample.

An example is given in appendix.

3.

```
#simulating a complete dataset
set.seed(331)
Z_1 <- rnorm(500,0,1)
Z_2 <- rnorm(500,0,1)
Z_3 <- rnorm(500,0,1)
Y <- list(Z_1 +1, 5 + 2 * Z_1 + Z_2)
y_1 <- Y[[1]]
y_2 <- Y[[2]]
```

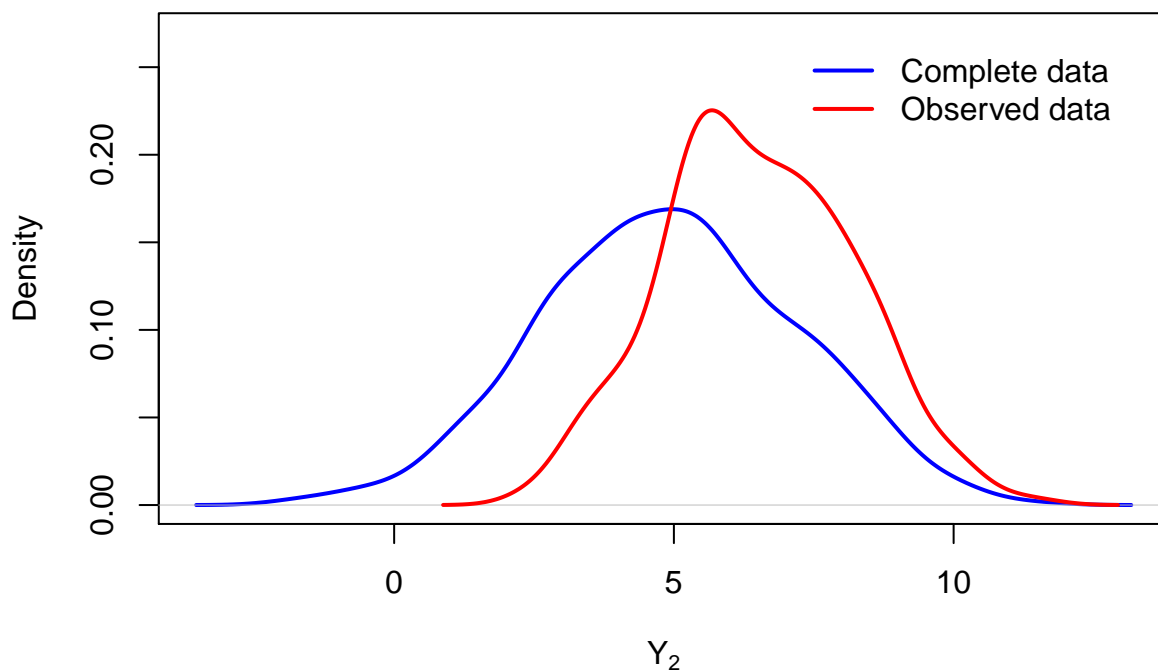
- (a) MAR

Reason: While  $a = 2$  and  $b = 0$ ,  $Y_2$  is missing if  $2 \times (Y_1 - 1) + Z_3 < 0$ , the missing of  $Y_2$  depends only on fully observed data  $Y_1$  and  $Z_3$ , but not  $Y_2$  itself, so this mechanism is MAR.

```
#Stimulate dataset with missing value of MAR
y_2_mis1 <- ifelse(((2 * y_1 - 2 + Z_3) < 0), NA, y_2)

#plotting the marginal distribution
plot(density(y_2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Marginal Distribution", ylim = c(0, 0.27))
lines(density(y_2_mis1, na.rm = TRUE), lwd = 2, col = "red")
legend(7, 0.27, legend = c("Complete data", "Observed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty = "n")
```

## Marginal Distribution



Comment: As expected, the complete data of  $Y_2 \sim N(5, 5)$ . Under this MAR mechanism, observed data has a greater mean and a smaller variance. It is probably because  $Y_2$  tends to be missing while  $Y_1$  is smaller than the mean 1, i.e.,  $Z_1$  is smaller than 0, as a result, smaller  $Y_2$  tends to be missing.

(b)

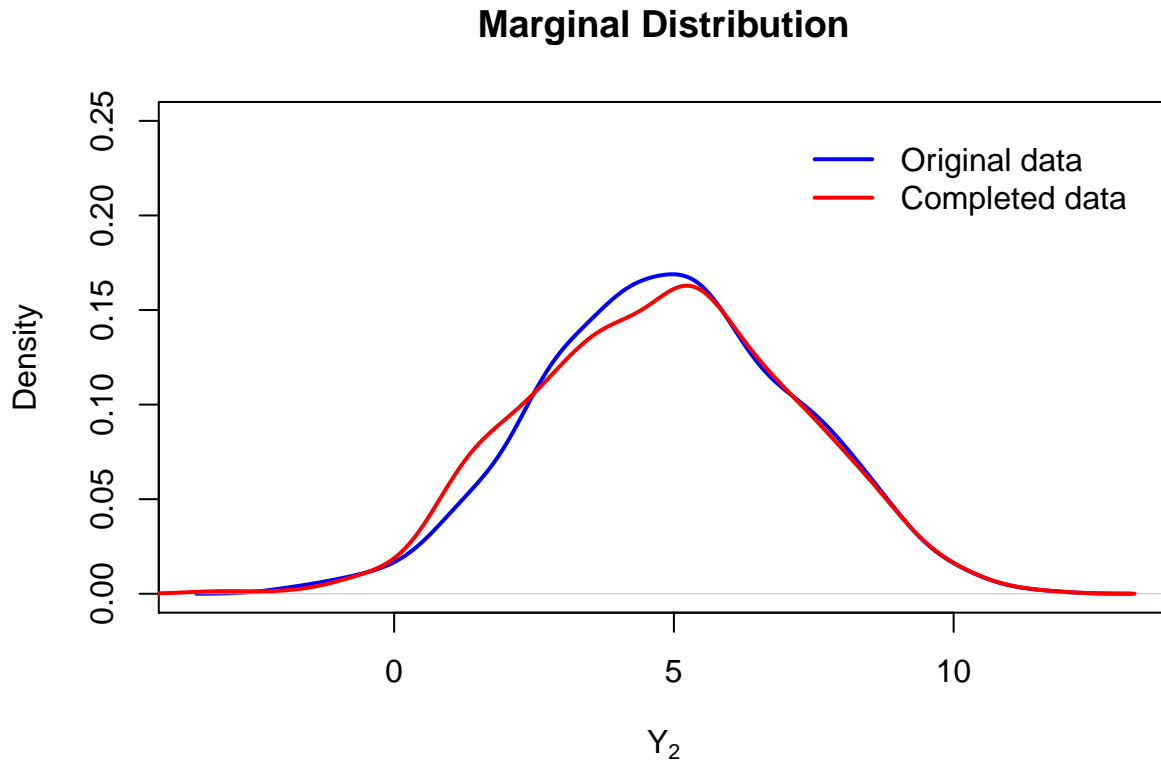
```
#stochastic regression imputation
Y_mis1 <- list(y_1, y_2_mis1)

fit1 <- lm(y_2_mis1~y_1)
predicted_sri1 <- predict(fit1, newdata = Y_mis1) + rnorm(length(y_1), 0, sigma(fit1))

y_2_sri1 <- ifelse(((2 * y_1 - 2 + Z_3) < 0), predicted_sri1, y_2)

#the marginal distribution
plot(density(y_2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Marginal Distribution", ylim = c(0, 0.25))
lines(density(y_2_sri1), lwd = 2, col = "red")
legend(7, 0.25, legend = c("Original data", "Completed data"),
```

```
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty = "n")
```



*Comment:* The data imputed by using stochastic regression imputation is quite similar with the original data especially the right half.

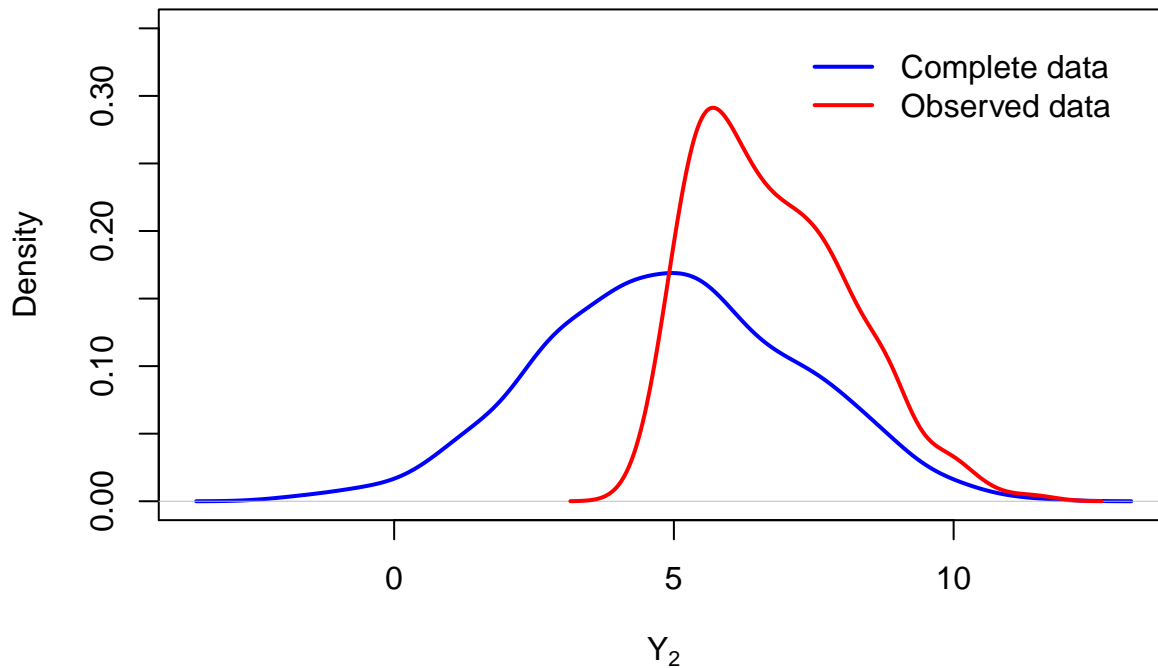
(c) MNAR

*Reason:* While  $a = 0$  and  $b = 2$ , the missing of  $Y_2$  depends on  $Z_3$  and  $Y_2$  itself, recall that the definition of MNAR is the probability of missing values is related to the specific values that should have been obtained, so this mechanism is MNAR.

```
#Stimulate dataset with missing value of MNAR
y_2_mis2 <- ifelse(((2 * y_2 - 10 + Z_3) < 0), NA, y_2)

#plotting the marginal distribution
plot(density(y_2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Marginal Distribution", ylim = c(0, 0.35))
lines(density(y_2_mis2, na.rm = TRUE), lwd = 2, col = "red")
legend(7, 0.35, legend = c("Complete data", "Observed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty = "n")
```

## Marginal Distribution



*Comment:* As we can observe, MNAR case is more extreme in terms of the dissimilarities between the two distributions than the MAR case. Under the MNAR mechanism, the miss dataset are mainly distributed on the left half ( $Y_2 < 5$ ). This is probably because  $Y_2$  tends to be missing when  $Y_2 < 5$ .

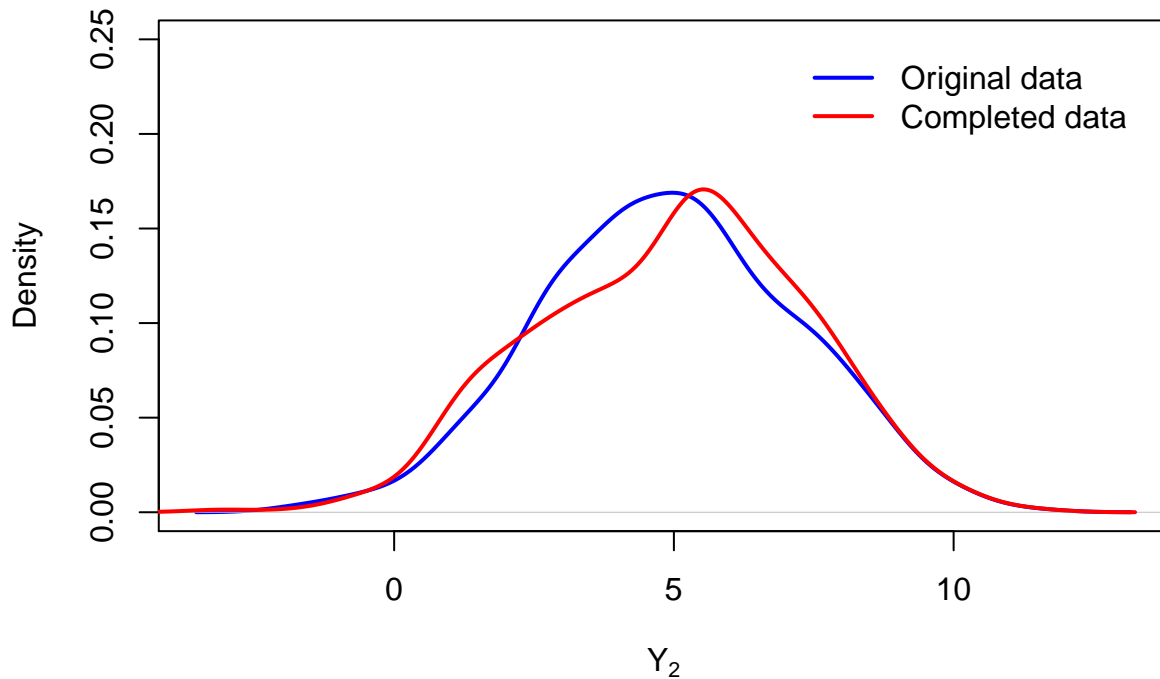
(d)

```
#stochastic regression imputation
Y_mis2 <- list(y_1, y_2_mis2)

fit2 <- lm(y_2_mis2~y_1)
predicted_sri2 <- predict(fit2, newdata = Y_mis2) + rnorm(length(y_1), 0, sigma(fit2))
y_2_sri2 <- ifelse(((2 * y_2 - 10 + Z_3) < 0), predicted_sri1, y_2)

#the marginal distribution
plot(density(y_2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Marginal Distribution", ylim = c(0, 0.25))
lines(density(y_2_sri2), lwd = 2, col = "red")
legend(7, 0.25, legend = c("Original data", "Completed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty = "n")
```

## Marginal Distribution



*Comment:* We can observe that data after imputation shows more difference on both left half and right half compared with 3(b). So it seems that the performance of stochastic regression imputation on MNAR case is not as good as on MAR case.

4.

- (a) The overall recovery time mean provided by a complete case analysis is 19.27273 and the associated standard error is 2.603013.

The correlations between the recovery time and the dose is 0.2391256,

The correlations between the recovery time and blood pressure is -0.01952862. (All the imputation value are given in Appendix)

```
#load data
load("databp.Rdata")

#complete case analysis
ind <- which(is.na((databp$recovtime)) == TRUE)
data_cc=databp[-ind,]

#mean value of the recovery time
Rmean <- mean(data_cc$recovtime)
Rmean

## [1] 19.27273

#standard error of the recovery time
sd(data_cc$recovtime)/sqrt(nrow(data_cc))

## [1] 2.603013

#correlations between the recovery time and the dose
cor(data_cc$recovtime, data_cc$logdose, use = "complete")
```

```
## [1] 0.2391256
```

```
#correlations between the recovery time and blood pressure  
cor(data_cc$recovtime, data_cc$bloodp, use = "complete")
```

```
## [1] -0.01952862
```

(b) The mean recovery time using mean imputation is 19.27273.

The overall recovery time mean provided by mean imputation is 2.284135 and the associated standard error is 2.284135.

The correlations between the recovery time and the dose is 0.2150612,

The correlations between the recovery time and blood pressure is -0.01934126.

```
#mean imputation  
Recovtime_mi <- ifelse(is.na(databp$recovtime), Rmean, databp$recovtime)  
  
#mean value of the recovery time after mean imputation  
mean(Recovtime_mi)
```

```
## [1] 19.27273
```

```
#standard error of the recovery time after mean imputation  
sd(Recovtime_mi)/sqrt(length(Recovtime_mi))
```

```
## [1] 2.284135
```

```
#correlations between the recovery time and the dose after mean imputation  
cor(Recovtime_mi, databp$logdose)
```

```
## [1] 0.2150612
```

```
#correlations between the recovery time and blood pressure after mean imputation  
cor(Recovtime_mi, databp$bloodp)
```

```
## [1] -0.01934126
```

(c) The overall recovery time mean provided by mean regression imputation is 19.44428 and the associated standard error is 2.312845

The correlations between the recovery time and the dose is 0.2801835,

The correlations between the recovery time and blood pressure is -0.01113646

```
#mean regression imputation  
fit <- lm(databp$recovtime ~ databp$logdose + databp$bloodp, data = databp)  
predicted_ri <- predict(fit, newdata = databp)  
Recovtime_ri <- ifelse(is.na(databp$recovtime), predicted_ri, databp$recovtime)  
  
#mean value of the recovery time after regression imputation  
mean(Recovtime_ri)
```

```
## [1] 19.44428
```

```
#standard error of the recovery time after regression imputation  
sd(Recovtime_ri)/sqrt(length(Recovtime_ri))
```

```
## [1] 2.312845
```

```
#correlations between the recovery time and the dose after regression imputation  
cor(Recovtime_ri, databp$logdose)
```

```
## [1] 0.2801835
```

```
#correlations between the recovery time and blood pressure after regression imputation  
cor(Recovtime_ri, databp$bloodp)
```

```
## [1] -0.0111364
```

- (d) The overall recovery time mean provided by stochastic regression imputation is 20.05764 and the associated standard error is 2.40414.

The correlations between the recovery time and the dose is 0.2701687,

The correlations between the recovery time and blood pressure is 0.02455702.

```
#stochastic regression imputation  
predicted_sri <- predict(fit, newdata = databp) + rnorm(nrow(databp), 0, sigma(fit))  
Recovtime_sri <- ifelse(is.na(databp$recovtime), predicted_sri, databp$recovtime)
```

```
#mean value of the recovery time after regression imputation  
mean(Recovtime_sri)
```

```
## [1] 20.05764
```

```
#standard error of the recovery time after regression imputation  
sd(Recovtime_sri)/sqrt(length(Recovtime_sri))
```

```
## [1] 2.40414
```

```
#correlations between the recovery time and the dose after regression imputation  
cor(Recovtime_sri, databp$logdose)
```

```
## [1] 0.2701687
```

```
#correlations between the recovery time and blood pressure after regression imputation  
cor(Recovtime_sri, databp$bloodp)
```

```
## [1] 0.02455702
```

*Extra care:* In some cases/examples, stochastic regression imputation can lead to implausible predictions by adding a random noise term. For instance, in this example, it may impute a negative value for the recovery time, which does not make sense.

- (e) The overall recovery time mean provided by predictive mean matching is 19.44 and the associated standard error is 2.464467

The correlations between the recovery time and the dose is 0.3037945,

The correlations between the recovery time and blood pressure is -0.03208685

```
#predictive mean matching  
#get the donor list by previous predicted data predicted_ri  
donor = 1:nrow(databp)  
predicted_mis = ifelse(is.na(databp$recovtime), NA, predicted_ri)  
  
for (i in 1:nrow(databp)){  
  donor[i] = databp$recovtime[which.min((predicted_mis - predicted_ri[[i]])**2)]  
}
```

```
Recovtime_pmm <- ifelse(is.na(databp$recovtime), donor, databp$recovtime)
```

```
#mean value of the recovery time after regression imputation  
mean(Recovtime_pmm)
```

```
## [1] 19.44
#standard error of the recovery time after regression imputation
sd(Recovtime_pmm)/sqrt(length(Recovtime_pmm))

## [1] 2.464467
#correlations between the recovery time and the dose after regression imputation
cor(Recovtime_pmm, databp$logdose)

## [1] 0.3037945
#correlations between the recovery time and blood pressure after regression imputation
cor(Recovtime_pmm, databp$bloodp)

## [1] -0.03208685
```

- (f) **Advantage of predictive mean matching:** It can avoid some implausible imputation because it choose a closest value to the predicted value from the original dataset, so every single imputation is proper. Predictive mean matching combines the advantages of stochastic regression imputation and hot deck imputation.

**Potential problem:** Like stochastic regression imputation, it will also attenuate standard errors of the estimates and increase the correlations between variables.

## APPENDIX

As for question 4, the position and values of imputation of each method is:

```
#position
ind

## [1] 4 10 22
#mean imputation
Recovtime_mi[ind]

## [1] 19.27273 19.27273 19.27273
#mean regression imputation
Recovtime_ri[ind]

## [1] 14.26254 21.51562 26.32896
#stochastic regression imputation
Recovtime_sri[ind]

## [1] 18.44350 37.95871 21.03881
#predictive mean matching
Recovtime_pmm[ind]

## [1] 13 10 39
```

As for question 2, a similar example is as follows:

```
a1 = as.vector(rnorm(15,0,1))
a2 = as.vector(rnorm(15,0,1))
a3 = as.vector(rnorm(15,0,1))
A=cbind(a1,a2,a3)

#max 90
```



```
A_max=A
A_max[1:5,]=NA
A_max
```

```
##           a1           a2           a3
## [1,]      NA           NA           NA
## [2,]      NA           NA           NA
## [3,]      NA           NA           NA
## [4,]      NA           NA           NA
## [5,]      NA           NA           NA
## [6,] -0.5467129 -0.8303291  0.3975629
## [7,] -0.6315766 -0.3224250 -0.1138464
## [8,]  0.1229661 -1.0662192  1.9896812
## [9,] -1.3882199  0.2082969  0.1887503
## [10,] -0.5370569 -0.1218543  0.5768689
## [11,]  0.2184279  1.2203151 -1.5678872
## [12,] -1.2438027 -1.5290357  1.0612612
## [13,] -0.8250594 -0.4483500 -0.9457818
## [14,]  0.6436179  1.9465743 -2.0275834
## [15,]  0.5518150  1.1685137 -2.0712742
```

```
#min 0
A_min = A
A_min[1:5,1]=NA
A_min[6:10,2]=NA
A_min[11:15,3]=NA
A_min
```

```
##           a1           a2           a3
## [1,]      NA  1.5993946  0.02342816
## [2,]      NA -0.2947007 -0.69419318
## [3,]      NA  0.9676587  0.72622387
## [4,]      NA  1.8896324  1.64054306
## [5,]      NA  2.5165380  0.56833066
## [6,] -0.5467129           NA  0.39756286
## [7,] -0.6315766           NA -0.11384643
## [8,]  0.1229661           NA  1.98968121
## [9,] -1.3882199           NA  0.18875027
## [10,] -0.5370569           NA  0.57686889
## [11,]  0.2184279  1.2203151           NA
## [12,] -1.2438027 -1.5290357           NA
## [13,] -0.8250594 -0.4483500           NA
## [14,]  0.6436179  1.9465743           NA
## [15,]  0.5518150  1.1685137           NA
```