How to simulate missing values?

Teresa Alves de Sousa, Imke Mayer

17 June 2020

Contents

Missing values occur in many domains and most datasets contain missing values (due to non-responses, lost records, machine failures, dataset fusions, etc.). These missing values have to be considered before or during analyses of these datasets.

Now, if you have a method that deals with missing values, for instance imputation or estimation with missing values, how can you assess the performance of your method on a given dataset? If the data already contains missing values, than this does not help you since you generally do not have a ground truth for these missing values. So you will have to simulate missing values, i.e. you remove values – which you therefore know to be the ground truth – to generate missing values.

The mechanisms generating missing values can be various but usually they are classified into three main categories defined by (Rubin 1976): *missing completely at random* (MCAR), *missing at random* (MAR) and *missing not at random* (MNAR). The first two are also qualified as *ignorable* missing values mechanisms, for instance in likelihood-based approaches to handle missing values, whereas the MNAR mechanism generates *nonignorable* missing values. In the following we will briefly introduce each mechanism (with the definitions used widely in the literature) and propose ways of simulations missing values under these three mechanism assumptions. For more precise definitions we refer to references in the bibliography on the R-miss-tastic website.

Notations

Let's denote by $\mathbf{X} \in \mathcal{X}_\infty \times \cdots \times \mathcal{X}_{\vee}$ the complete observations. We assume that $\mathbf{X}$ is a concatenation of p columns $X_j \in \mathcal{X}_|$, $j \in \{1, \ldots, p\}$, where $dim(\mathcal{X}_|) = n$ for all j.

The data can be composed of quantitative and/or qualitative values, hence $\mathcal{X}_|$ can be $\mathbb{R}^n$, $\mathbb{Z}^n$ or more generally $\mathcal{S}^n$ for any discrete set S.

Missing values are indicated as `NA` (not available) and we define an indicator matrix $\mathbf{R} \in \{0, 1\}^{n \times p}$ such that $R_{ij} = 1$ if X_{ij} is observed and $R_{ij} = 0$ otherwise. We call this matrix $\mathbf{R}$ the response (or missingness) pattern of the observations $\mathbf{X}$. According to this pattern, we can partition the observations $\mathbf{X}$ into observed and missing: $\mathbf{X} = (\mathbf{X}^{obs}, \mathbf{X}^{mis})$.

We generate a small example of observations $\mathbf{X}$:

```
suppressPackageStartupMessages(require(MASS))
suppressPackageStartupMessages(require(norm))
suppressPackageStartupMessages(require(VIM))
suppressPackageStartupMessages(require(ggplot2))
suppressPackageStartupMessages(require(naniar))

source("amputation.R")
set.seed(1)
# Sample data generation ------------------------------------------------------
# Generate complete data
mu.X <- c(1, 1)
Sigma.X <- matrix(c(1, 1, 1, 4), nrow = 2)
n <- 100
X.complete.cont <- mvrnorm(n, mu.X, Sigma.X)

lambda <- 0.5
X.complete.discr <- rpois(n, lambda)

n.cat <- 5
X.complete.cat <- rbinom(n, size=5, prob = 0.5)

X.complete <- data.frame(cbind(X.complete.cont, X.complete.discr, X.complete.cat))
X.complete[,4] <- as.factor(X.complete[,4])
levels(X.complete[,4]) <- c("F", "E", "D", "C", "B", "A")
```

What data can be handled

With the main function `produce_NA` it is possible to generate missing values for quantitative, categorical or mixed data, provided that it is available in form of a `data.frame` or `matrix`.

Missing values can be generated following one or more of the three main missing values mechanisms (see below for details).

If the data is already incomplete, it is possible to add a specific amount of additional missing values, in the already incomplete features or other complete features.

Important: Currently there is no option available for the mains function `produce_NA` to specify that every observation must contain at least one value after amputation. Hence, in the data.frame output by `produce_NA` there might be empty observations.

Except for the MCAR mechanism, our function `produce_NA` internally calls the `ampute` function of the `mice` R-package. See (Schouten, Lugtig, and Vink 2018) for a detailed description of this latter function.

Use of `produce_NA` with default settings

Minimal set of arguments

In order to generate missing values for given data, `produce_NA` requires the following arguments:

- `data`: the initial data (can be complete or incomplete) as a matrix or data.frame
- `mechanism`: one of "MCAR", "MAR", "MNAR" (default: "MCAR")
- `perc.missing`: the proportion of new missing values among the initially observed values (default: 0.5)

Value

`produce_NA` returns a list containing three elements:

- `data.init`: the initial data
- `data.incomp`: the data with the newly generated missing values (and the initial missing values if applicable)
- `idx_newNA`: a matrix indexing only the newly generated missing values

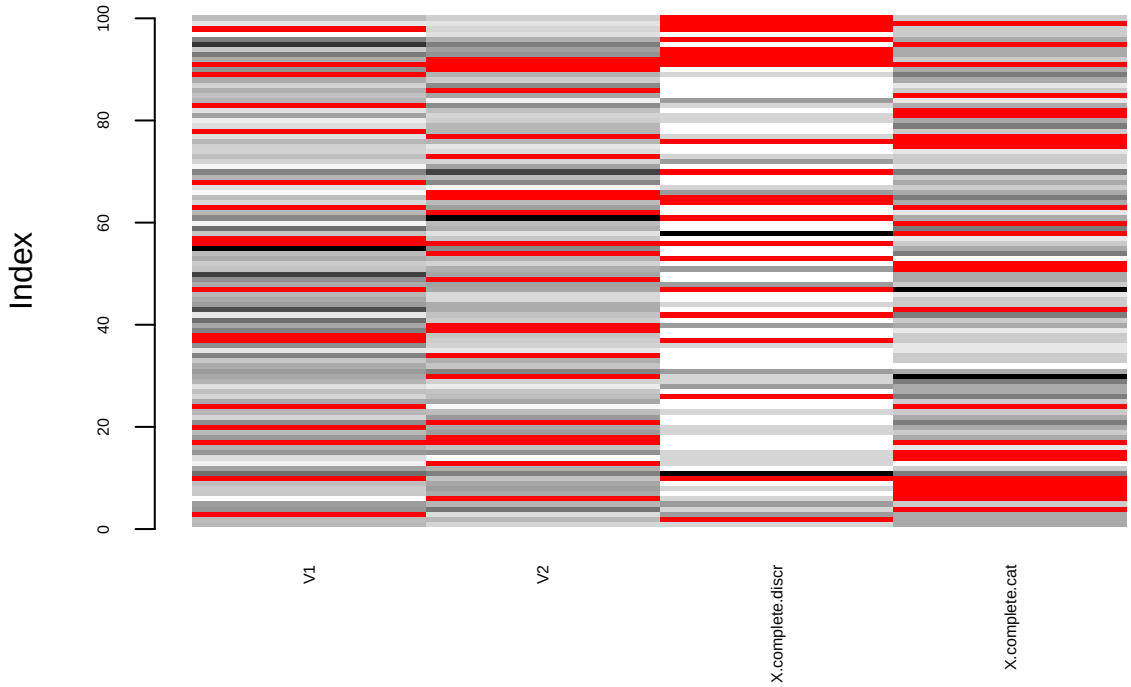
Example

On complete data

```
# Minimal example for generating missing data ------------------------
X.miss <- produce_NA(X.complete, mechanism="MCAR", perc.missing = 0.2)

X.mcar <- X.miss$data.incomp
R.mcar <- X.miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.mcar)/prod(dim(R.mcar))))
```

```
## Percentage of newly generated missing values: 0.21
```

```
matrixplot(X.mcar, cex.axis = 0.5, interactive = F)
```

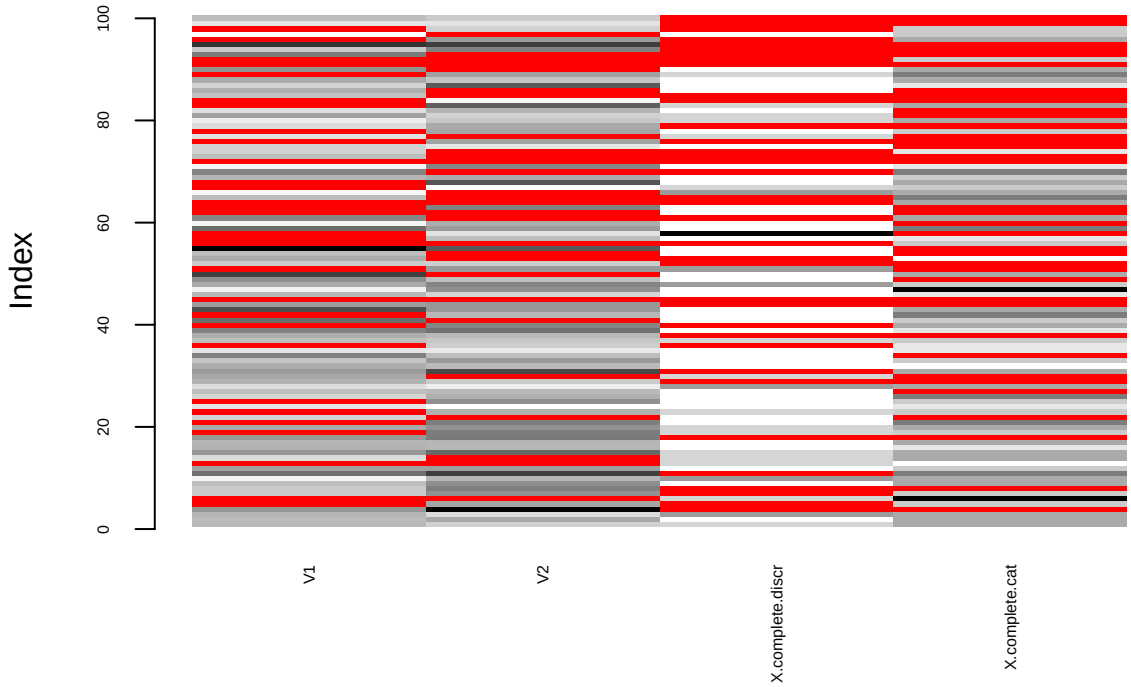


On incomplete data:

```
# Minimal example for generating missing data on an incomplete data set ------------------------
X.miss <- produce_NA(rbind(X.complete[1:50,], X.mcar[51:100,]) , mechanism="MCAR", perc.missing = 0.2)

X.mcar <- X.miss$data.incomp
R.mcar <- X.miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.mcar)/prod(dim(R.mcar))))
```

```
## Percentage of newly generated missing values: 0.225
```

```
matrixplot(X.mcar, cex.axis = 0.5, interactive = F)
```



Details on all available specifications

The main function `produce_NA` allows generating missing values in various ways. These can be specified through different arguments:

produce_NA(data, mechanism = "MCAR", perc.missing = 0.5, self.mask=NULL, idx.incomplete = NULL, idx.covariates = NULL, weights.covariates = NULL, by.patterns = FALSE, patterns = NULL, freq.patterns = NULL, weights.patterns = NULL, logit.model = "RIGHT", seed = NULL)

Mechanisms

In order to define the different missing values mechanisms, both $\mathbf{X}$ and $\mathbf{R}$ are modeled as random variables with probability distributions $\mathbb{P}_X$ and $\mathbb{P}_R$ respectively. We parametrize the missingness distribution $\mathbb{P}_R$ by a parameter ϕ.

MCAR

Definition

The observations are said to be Missing Completely At Random (MCAR) if the probability that an observation is missing is independent of the variables and observations: the probability that an observation is missing does not depend on $(\mathbf{X}^{obs}, \mathbf{X}^{mis})$. Formally this is:

$$\mathbb{P}_R(R \,|\, X^{obs}, X^{mis}; \phi) = \mathbb{P}_R(R) \qquad \forall\, \phi.$$

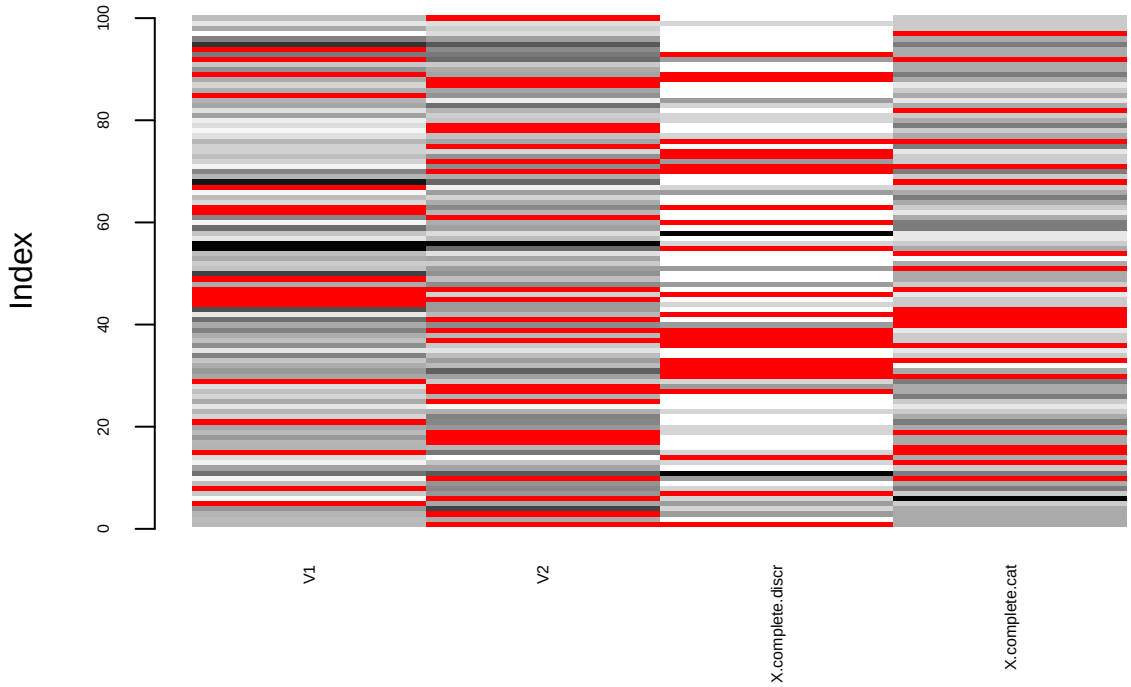
Example

```
# Sample mcar missing data ----------------------------------------
mcar <- produce_NA(X.complete, mechanism="MCAR", perc.missing = 0.2)
```

```
X.mcar <- mcar$data.incomp
R.mcar <- mcar$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.mcar)/prod(dim(R.mcar))))
```

```
## Percentage of newly generated missing values: 0.2175
```

```
matrixplot(X.mcar,  cex.axis = 0.5, interactive = F)
```



MAR

Definition

The observations are said to be Missing At Random (MAR) if the probability that an observation is missing only depends on the observed data $\mathbf{X}^{obs}$. Formally,

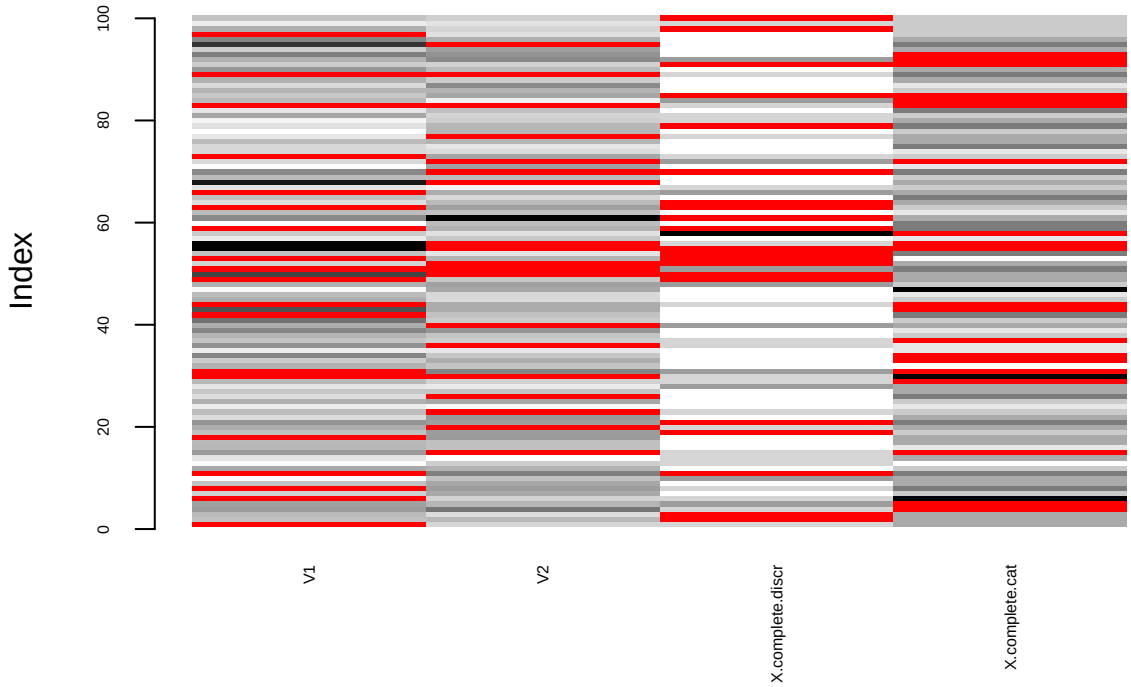
$$\mathbb{P}_R(R \,|\, X^{obs}, X^{mis}; \phi) = \mathbb{P}_R(R \,|\, X^{obs}; \phi) \qquad \forall\, \phi, \forall\, X^{mis}.$$

Example

```
# Sample mar missing data ----------------------------------------
mar <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2)

X.mar <- mar$data.incomp
R.mar <- mar$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.mar)/prod(dim(R.mar))))
```

```
## Percentage of newly generated missing values: 0.1975
```

```
matrixplot(X.mar,  cex.axis = 0.5, interactive = F)
```



MNAR

Definition

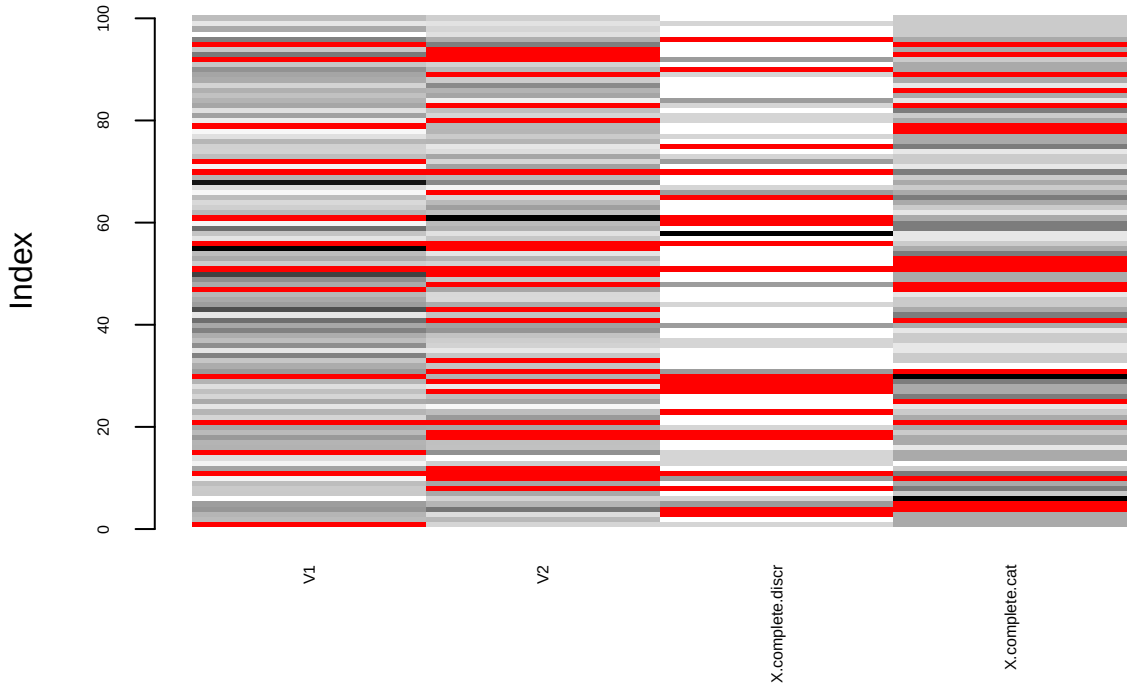
The observations are said to be Missing Not At Random (MNAR) in all other cases.

Example 1: logistic model with missing values as predictors

```
# Sample mnar missing data ----------------------------------------
mnar <- produce_NA(X.complete, mechanism="MNAR", perc.missing = 0.2)

X.mnar <- mnar$data.incomp
R.mnar <- mnar$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.mnar)/prod(dim(R.mnar))))
```

```
## Percentage of newly generated missing values: 0.1975
```

```
matrixplot(X.mnar,  cex.axis = 0.5, interactive = F)
```



Example 2: self-masking MNAR (for quantitative variables)

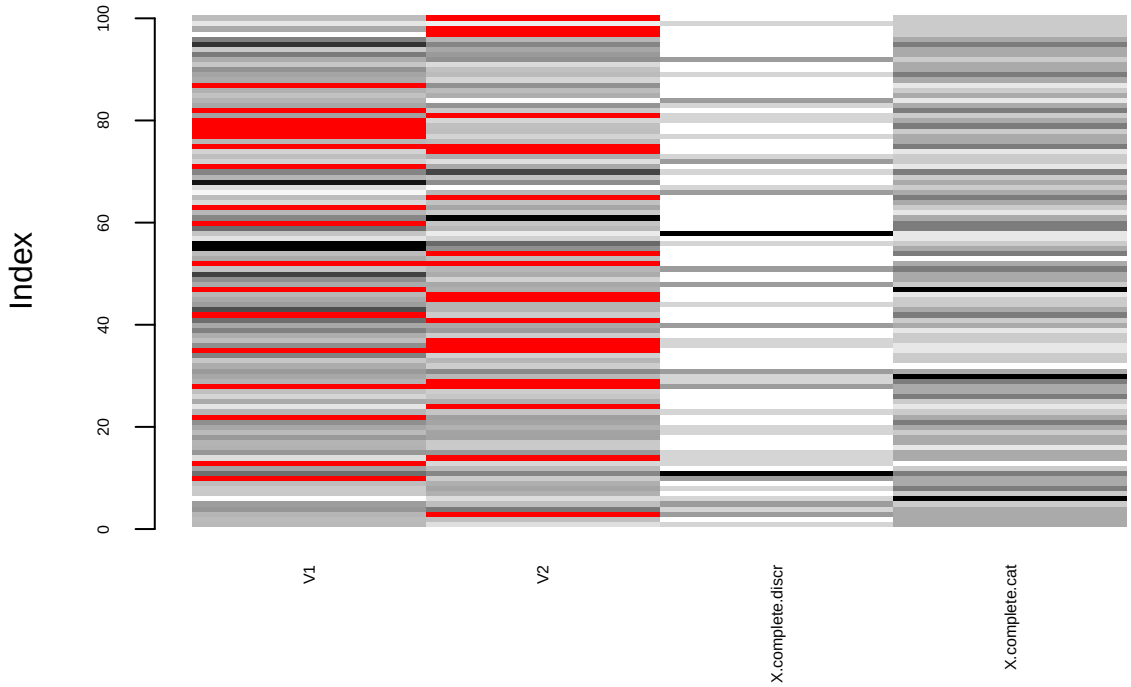
```
# Sample mnar missing data ----------------------------------------
mnar <- produce_NA(X.complete, mechanism="MNAR", perc.missing = 0.2, self.mask="lower", idx.incomplete =

X.mnar <- mnar$data.incomp
R.mnar <- mnar$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", 100*sum(R.mnar)/prod(dim(R.mnar))))
```

```
## Percentage of newly generated missing values: 9.5
```

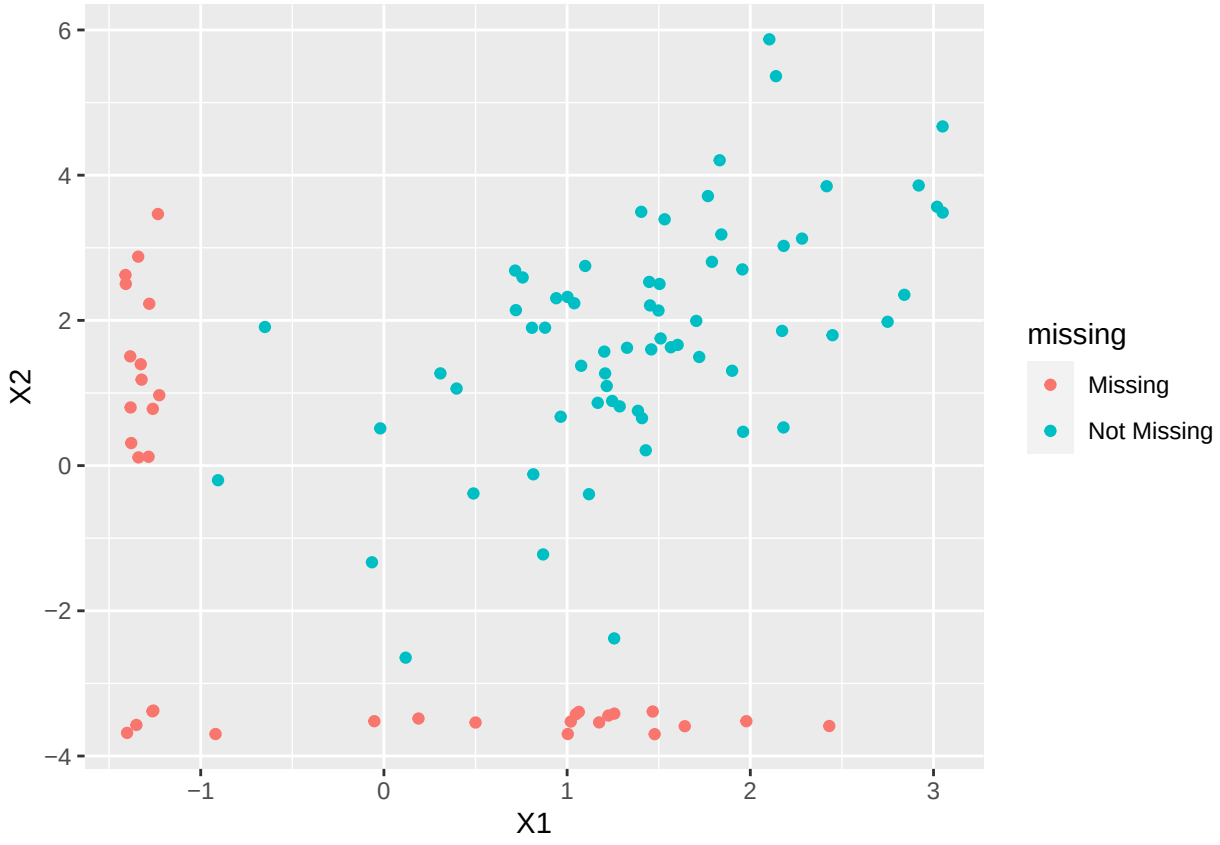
```
writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables):
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 19
```

```
matrixplot(X.mnar,  cex.axis = 0.5, interactive = F)
```




```
ggplot(data=data.frame(X1=X.mnar[,1], X2=X.mnar[,2]),
       aes(x = X1,
           y = X2)) +
  geom_miss_point()
```



Note that the proportion of missing values specified in the function call refers to the proportion w.r.t. the incomplete variables. Hence if you select half of your variables, to contain missing values and choose `perc.missing=0.2`, then the total proportion of missing values in the entire matrix/data.frame will be 0.2/2 = 0.1.

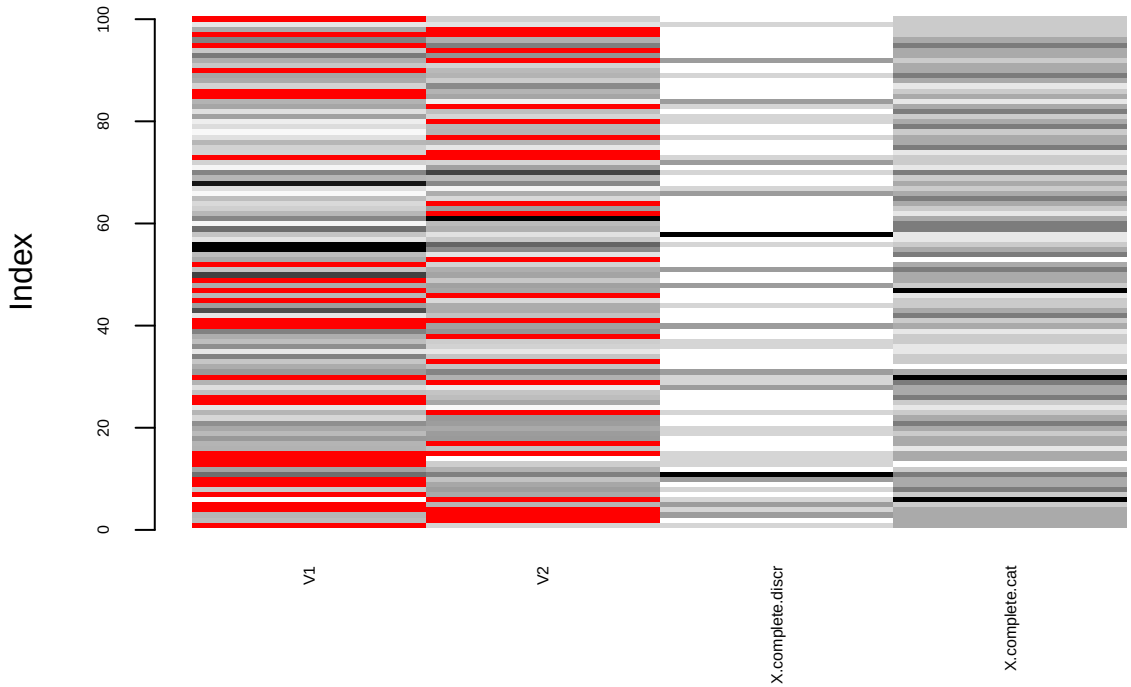
Specify incomplete variables

If you want to generate missing values only for a certain subset of variables, you can specify them by providing their position in the matrix/data.frame:

```
# Sample missing data for the first two variables in X ----------------------------------------
miss <- produce_NA(X.complete, mechanism="MCAR", perc.missing = 0.2, idx.incomplete = c(1, 2))

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables):
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 0.245
```

```
matrixplot(X.miss,  cex.axis = 0.5, interactive = F)
```



Note that the proportion of missing values specified in the function call refers to the proportion w.r.t. the incomplete variables. Hence if you select half of your variables, to contain missing values and choose `perc.missing=0.2`, then the total proportion of missing values in the entire matrix/data.frame will be 0.2/2 = 0.1.

Mice specific arguments

In the `mice` package there exists a function that allows already to generate missing values, `mice::ampute`. Our `produce_NA` function calls this function at some point but we chose to extend certain options, for

instance with `mice::ampute` it currently is not possible to add new missing values to an already incomplete data.frame/matrix.

In order to stay close to this `ampute` function from `mice` we adopted (and adapted) some of its arguments.

Covariates and covariates weights

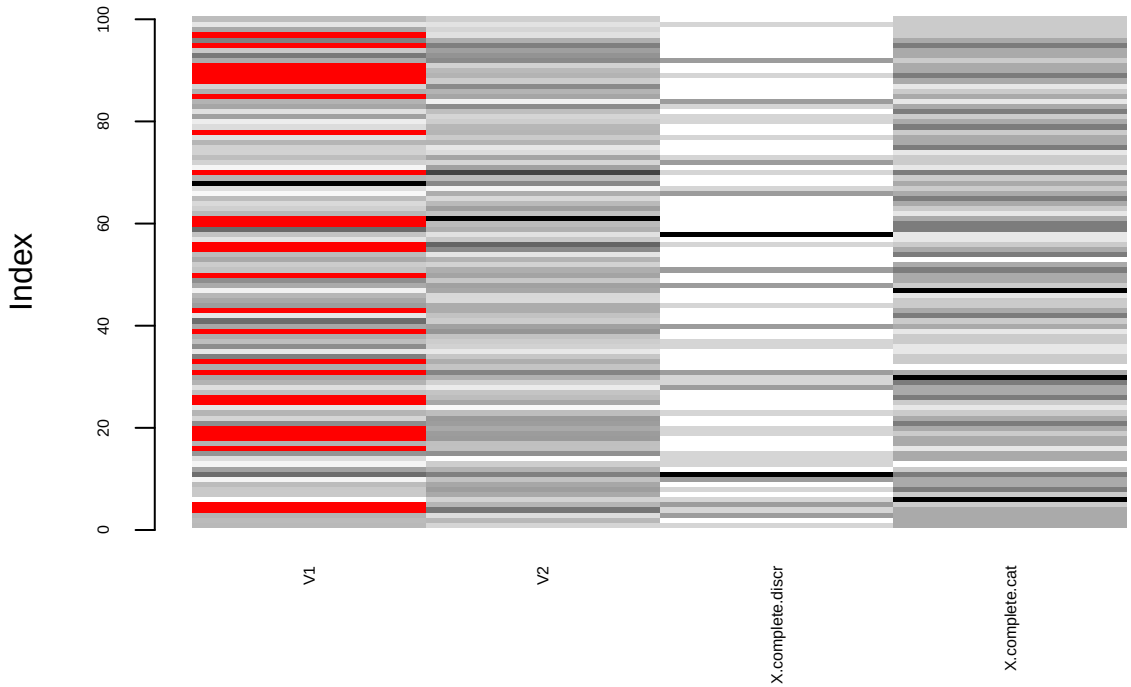
If you want to generate MAR or MNAR missing values, you can specify which variables will be used in the missingness model. You need to specify the variables that you want to use with a binary vector. For instance if you want to use variables 1 to 3 out of 7 variales, then you specify `idx.covariates = c(1,1,1,0,0,0,0)`. And you need to specify their weights as well, i.e. their contribution in the model. For instance `weights.covariates = c(1/3, 1/3, 1/3, 0, 0, 0, 0)`

Remark: if you choose `mechanism="MAR"` and `idx.incomplete = c(1,2)`, then `idx.covariates` must be of the form `c(0,0,*,*,...,*)` where `*` can be either 0 or a positive weight.

```
# Sample missing data for the first two variables in X ----------------------------------
miss <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2, idx.incomplete = c(1), idx.covariat

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables):
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 0.26
```

```
matrixplot(X.miss,  cex.axis = 0.5, interactive = F)
```



Patterns

One might want to specify certain response/missingness patterns that are more relevant than others for a given application. This is possible by passing a matrix or data.frame whose rows contain the different

patterns one wishes to generate. Additionally it is possible to specify the frequency of each pattern. We refer to the vignette of the mice::ampute function for more details on this and other related options.

This option is only implemented for the MAR and MNAR mechanisms.

Default patterns

If you want to use patterns but do not wish to specify them manually, you can set `by.patterns=T` and the patterns will automatically be of the form:

$$\begin{array}{cccccccc} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 \\ & & & \dots & & & \\ & & & \dots & & & \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 \end{array}$$

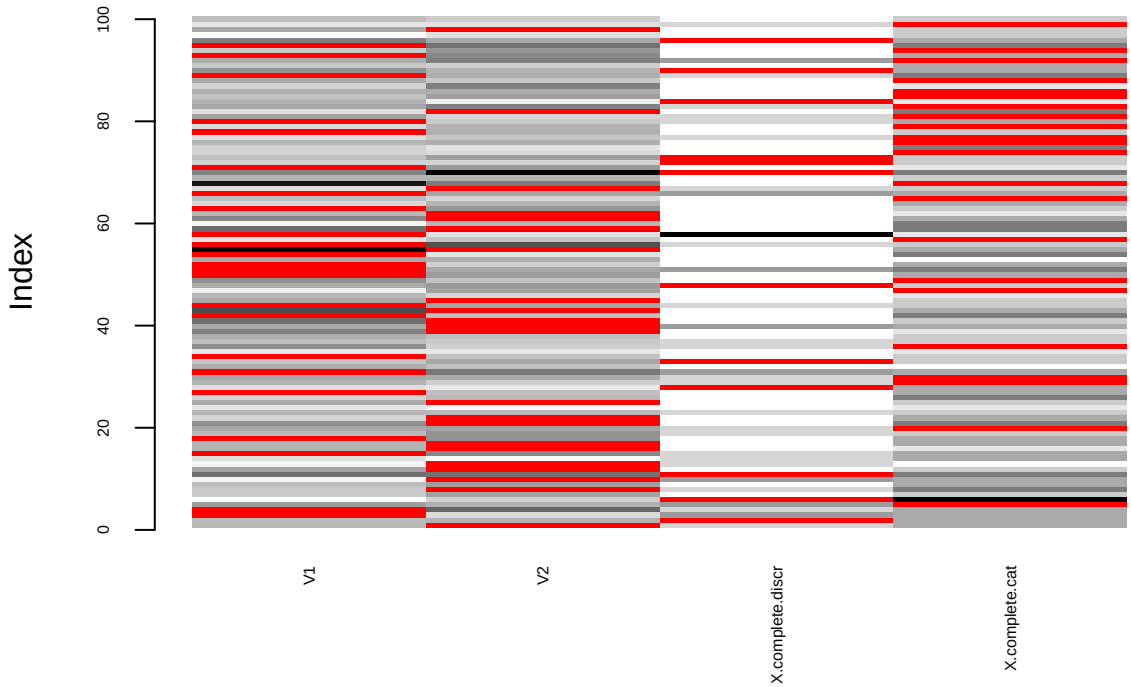
This means that `0` indicates that the variable should have missing values whereas `1` means that it should be observed.

Use default patterns:

```
# Sample missing data by using the by.patterns option --------------------------------
miss <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2, by.patterns = T)

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables): "
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 0.1975
```

```
matrixplot(X.miss,  cex.axis = 0.5, interactive = F)
```

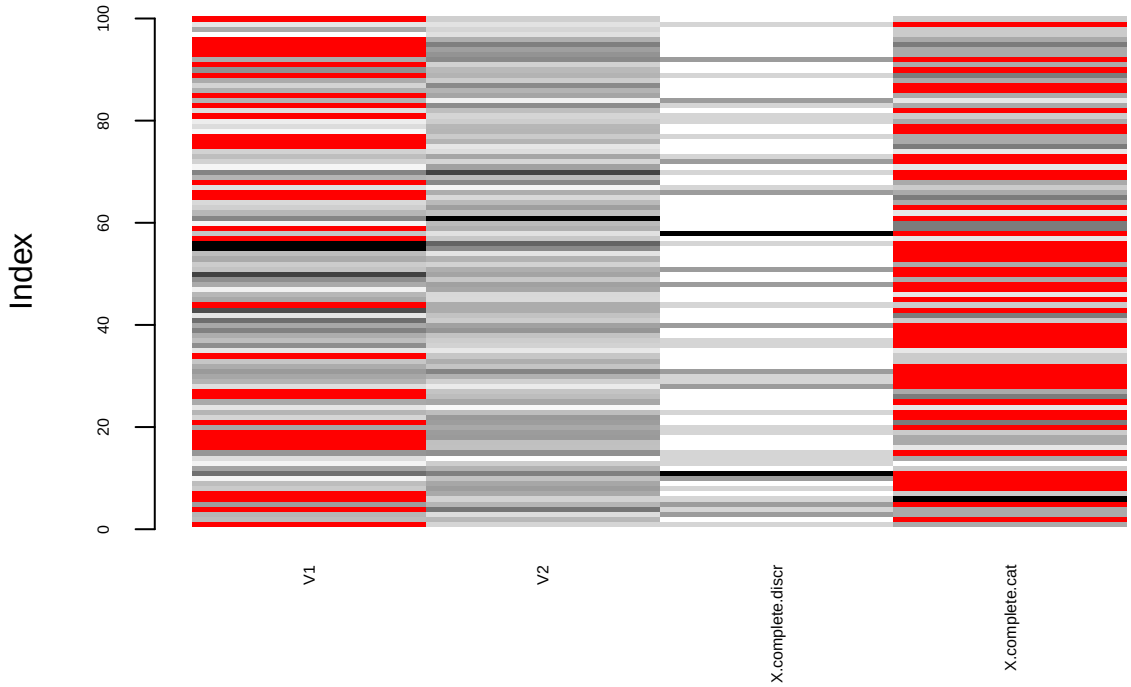


Specify different patterns:

```r
# Sample missing data by using the by.patterns option and user-specified patterns ----
miss <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2, idx.incomplete = c(1,4), by.patterns

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables): "
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 0.385
```

```r
matrixplot(X.miss,  cex.axis = 0.5, interactive = F)
```

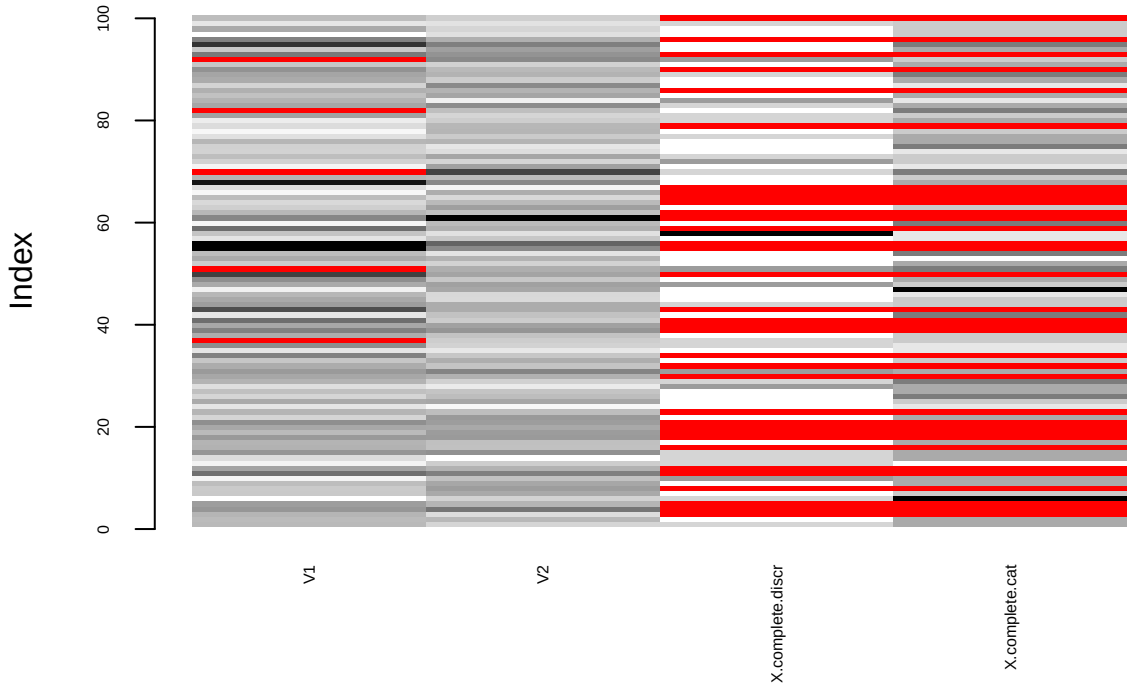


Additionally specify the frequency of each pattern:

```r
# Sample missing data by using the by.patterns option and user-specified patterns ----
miss <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2, idx.incomplete = c(1,3,4), by.patte

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values (only w.r.t. to incomplete variables): "
```

```
## Percentage of newly generated missing values (only w.r.t. to incomplete variables): 0.25
```

```r
matrixplot(X.miss,  cex.axis = 0.5, interactive = F)
```



Logistic model

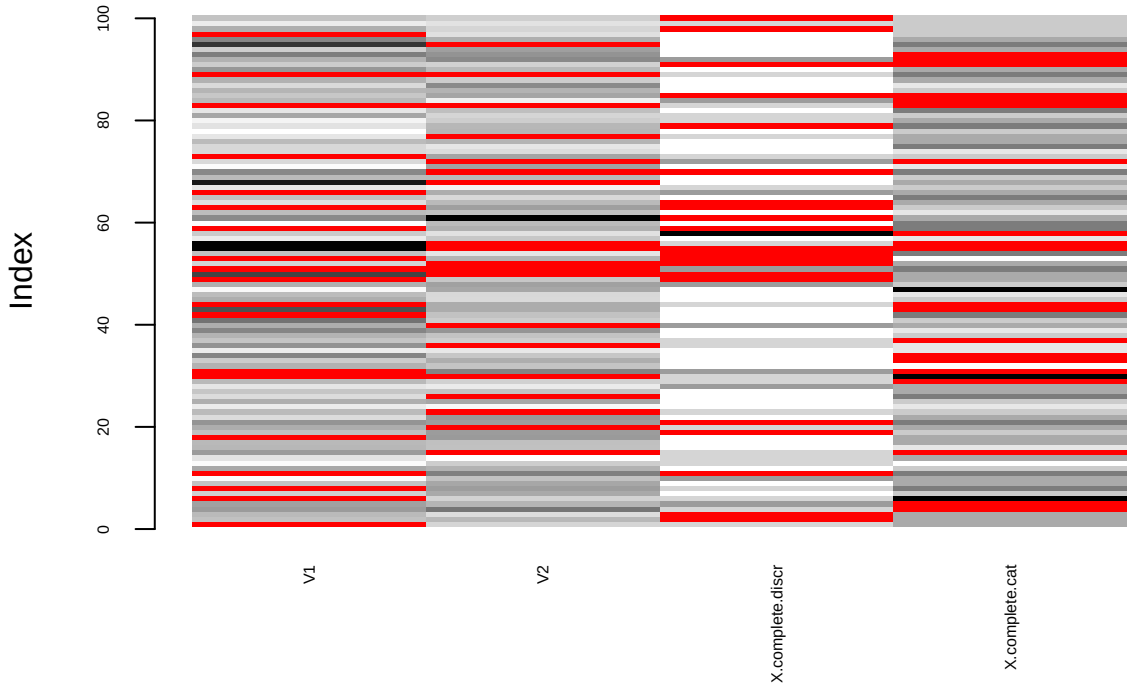
There are four possible logistic distribution functions implemented in the `mice::ampute` function: left-tailed (`"LEFT"`), right-tailed (`"RIGHT"`), centered (`"MID"`), both-tailed (`"TAIL"`).

From the mice vignette: "[These] functions are applied to the weighted sum scores. For instance, in the situation of RIGHT missingness, cases with high weighted sum scores will have a higher probability to have missing values, compared to cases with low weighted sum scores."

```
# Sample mar missing data with centered logistic distribution funciton ---------------
miss <- produce_NA(X.complete, mechanism="MAR", perc.missing = 0.2, logit.model = "MID")

X.miss <- miss$data.incomp
R.miss <- miss$idx_newNA

writeLines(paste0("Percentage of newly generated missing values: ", sum(R.miss)/prod(dim(R.miss))))
```

```
## Percentage of newly generated missing values: 0.22
```

```
matrixplot(X.mar,  cex.axis = 0.5, interactive = F)
```



Other options

- `seed`: specify a seed for the random values generator, useful to obtain reproducible examples.

Full list of arguments

```
#' @param data [data.frame, matrix] (mixed) data table (n x p)
#' @param mechanism [string] either one of "MCAR", "MAR", "MNAR"; default is "MCAR"
#' @param self.mask [string] either NULL or one of "sym", "upper", "lower"; default is NULL
#' @param perc.missing [positive double] proportion of missing values, between 0 and 1; default is 0.5
#' @param idx.incomplete [array] indices of variables to generate missing values for; if NULL then miss
#' @param idx.covariates [matrix] binary matrix such that entries in row i that are equal to 1 indicate
#' @param weights.covariates [matrix] matrix of same size as idx.covariates with weights in row i for c
#' @param by.patterns [boolean] generate missing values according to (pre-specified) patterns; default
#' @param patterns [matrix] binary matrix with 1=observed, 0=missing (n_pattern x p); default is NULL
#' @param freq.patterns [array] array of size n_pattern containing desired proportion of each pattern;
#' @param weights.patterns [matrix] weights used to calculate weighted sum scores (n_pattern x p); if N
#' @param logit.model [string] either one of "RIGHT","LEFT","MID","TAIL"; default is "RIGHT"
#' @param seed [natural integer] seed for random numbers generator; default is NULL
#'
#' @return A list with the following elements
#' \item{data.init}{original data.frame}
#' \item{data.incomp}{data.frame with the newly generated missing values, observed values correspond to
#' \item{idx_newNA}{a boolean data.frame indicating the indices of the newly generated missing values}
```

References

Rubin, Donald B. 1976. "Inference and Missing Data." *Biometrika* 63 (3). [Oxford University Press, Biometrika Trust]: 581–92. http://www.jstor.org/stable/2335739.

Schouten, Rianne Margaretha, Peter Lugtig, and Gerko Vink. 2018. "Generating Missing Values for Simulation Purposes: A Multivariate Amputation Procedure." *Journal of Statistical Computation and Simulation* 88 (15). Taylor & Francis: 2909–30.