A simple example of multiple imputation using the mice package

Brice Ozenne

October 28, 2019

This document gathers code from the documentation of the mice package. See https://stefvanbuuren.name/mice/.

Load packages

```
library(lava)
library(mice)
library(data.table)
library(ggplot2)
```

1 Simulate data (just to have an example to work with)

Generative model

```
mSim <- lvm(Y~group+season+bmi+gender+age)
categorical(mSim, labels = c("winter","summer")) <- ~season
categorical(mSim, labels = c("SAD","HC")) <- ~group
categorical(mSim, labels = c("Male","Female")) <- ~gender
distribution(mSim,~bmi) <- lava::gaussian.lvm(mean = 22, sd = 3)
distribution(mSim,~age) <- lava::uniform.lvm(20,80)
```

Sampling

```
n <- 1e2
set.seed(10)
dt.data <- as.data.table(sim(mSim,n))
```

Add missing values

```
dt.data[1:10, bmi:=NA]
```

2 Working with mice

2.1 Step 1: Inspect the missing data pattern

Check the number of missing values in the dataset:

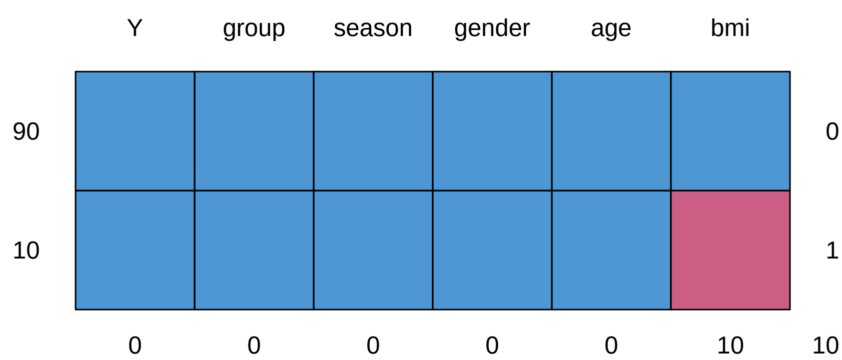
```
colSums(is.na(dt.data))
```

```
Y  group season    bmi gender    age
0      0      0     10      0      0
```

Missing data patterns:

```
md.pattern(dt.data)
```

```
   Y group season gender age bmi
90 1     1      1      1   1   1  0
10 1     1      1      1   1   0  1
   0     0      0      0   0  10 10
```



2.2 Step 2: Define imputation model

```
all.variables <- c("Y","group","season","bmi","gender","age")
n.variables <- length(all.variables)
Mlink <- matrix(0, n.variables, n.variables,
                dimnames = list(all.variables,all.variables))
Mlink["bmi",c("group","season","gender","age")] <- 1
Mlink
```

```
       Y group season bmi gender age
Y      0     0      0   0      0   0
group  0     0      0   0      0   0
season 0     0      0   0      0   0
bmi    0     1      1   0      1   1
gender 0     0      0   0      0   0
age    0     0      0   0      0   0
```

A value of 1 means that the column variable is used as a predictor for the target block (in the rows).

2.3 Step 3: Generate imputed datasets

Generate imputed values

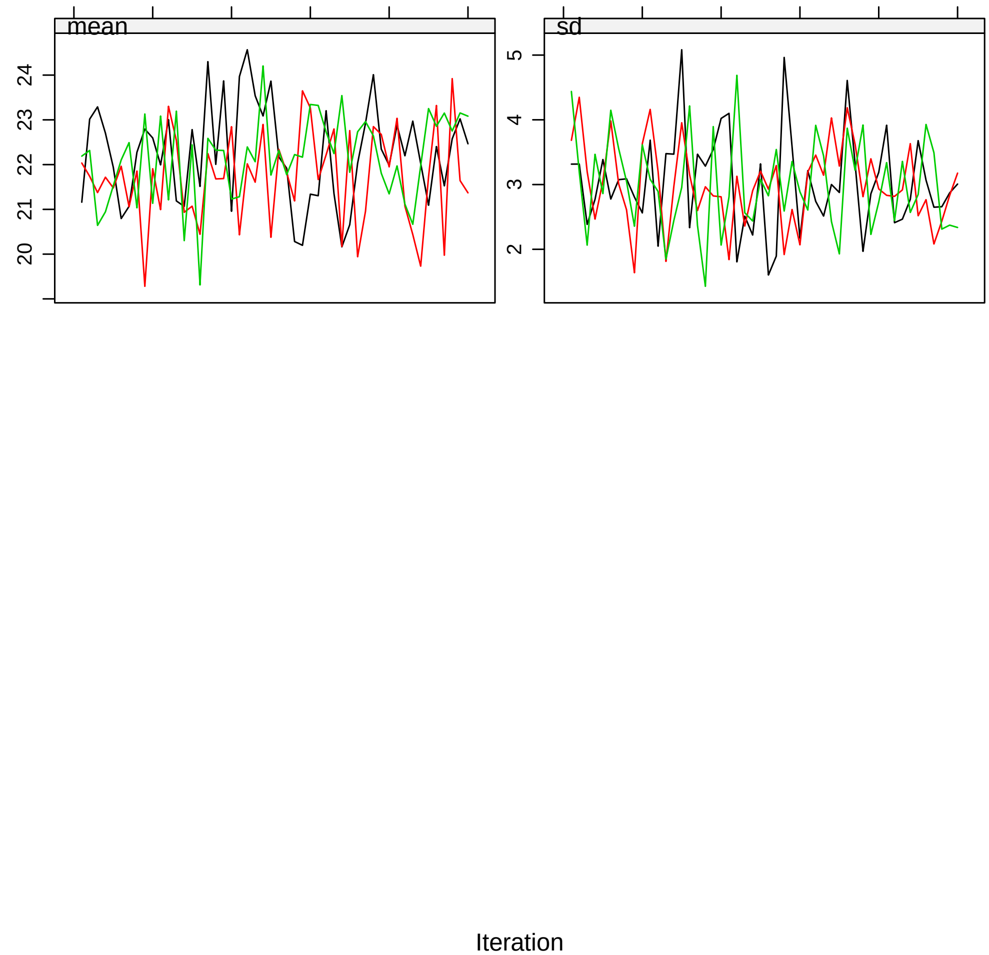
```
n.imputed <- 3 ## number of imputed datasets
dt.mice <- mice(dt.data,
                m=n.imputed,
                maxit = 50, # number of iterations to obtain the
   imputed dataset
                predictorMatrix = Mlink,
                method = 'pmm', # Predictive mean matching, only ok for
    continuous variables, it is possible to set constrains for positive
    variables
                seed = 500, printFlag = FALSE)
summary(dt.mice)
```

```
Class: mids
Number of multiple imputations:  3
Imputation methods:
     Y  group season    bmi gender    age
    ""     ""     ""  "pmm"     ""     ""
PredictorMatrix:
       Y group season bmi gender age
Y      0     0      0   0      0   0
group  0     0      0   0      0   0
season 0     0      0   0      0   0
bmi    0     1      1   0      1   1
gender 0     0      0   0      0   0
age    0     0      0   0      0   0
```

2.4 Step 4: Check the imputed datasets

2.4.1 Convergence of the imputation algorithm

```
plot(dt.mice)
```



2.4.2 Visualizing the imputed values

Visualize imputed value values and check they are plausible (e.g. mice is not imputed a BMI of 75):

```
dt.mice$imp$bmi
```

```
         1        2        3
1 25.68855 25.31909 21.60139
2 27.25524 15.38820 19.28934
3 25.31909 22.82264 21.60139
4 21.94247 25.98147 24.80171
```

```
5  17.42985 21.94247 25.68855
6  22.68303 18.98739 20.97076
7  21.82216 21.93016 22.82264
8  19.81314 21.13770 26.03528
9  22.82264 21.88207 25.68855
10 19.87741 18.29777 22.31832
```

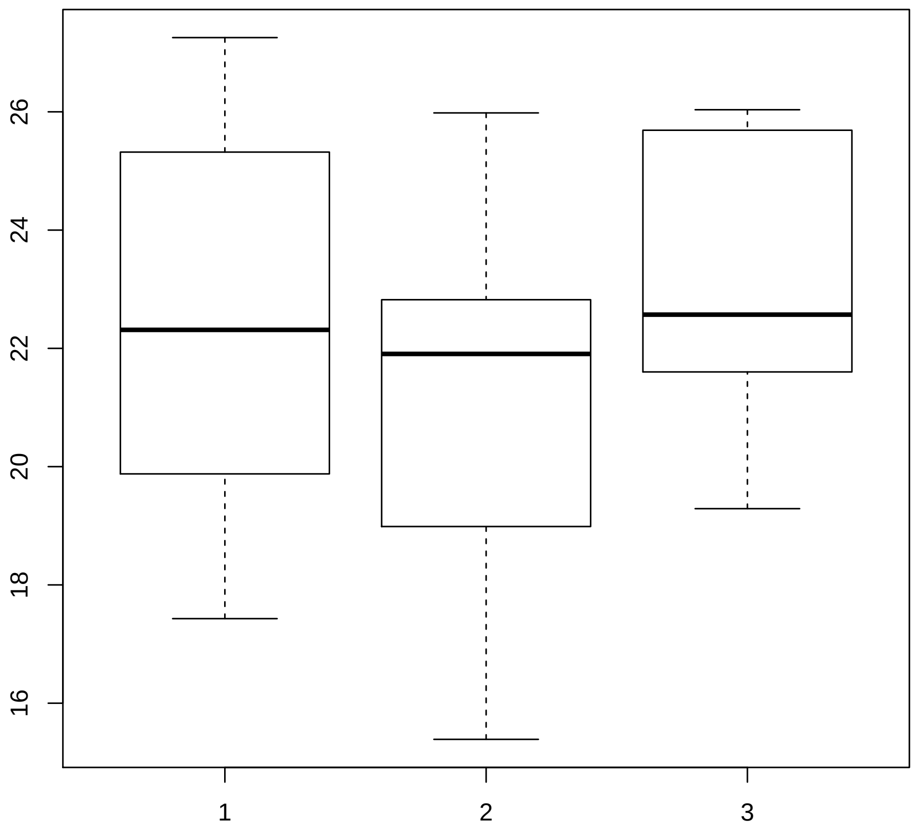
The rows correspond to the 3 different imputed datasets and the columns to 10 imputed values per dataset. One can also summarizes the imputed values computing their quantiles:

```
apply(dt.mice$imp$bmi,2,quantile)
```

```
            1        2        3
0%   17.42985 15.38820 19.28934
25%  20.36360 19.52497 21.60139
50%  22.31275 21.90611 22.57048
75%  24.69498 22.60260 25.46684
100% 27.25524 25.98147 26.03528
```

Boxplot of the imputed values:

```
boxplot(dt.mice$imp$bmi)
```



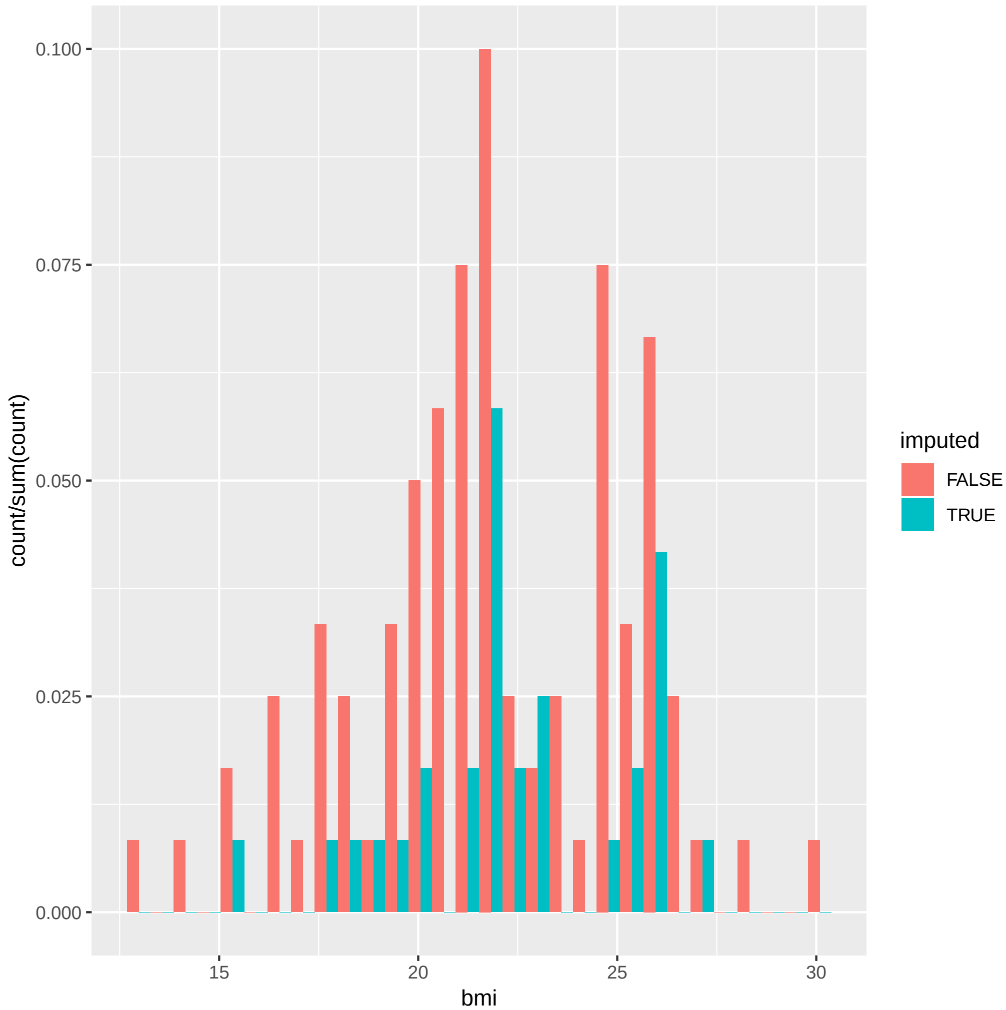
Imputed values vs. observed values

```
dt.bmi <- rbind(data.table(bmi = unlist(dt.mice$imp$bmi), imputed =
    TRUE),
                data.table(bmi = na.omit(dt.data$bmi), imputed = FALSE)
    )
```

Histogram

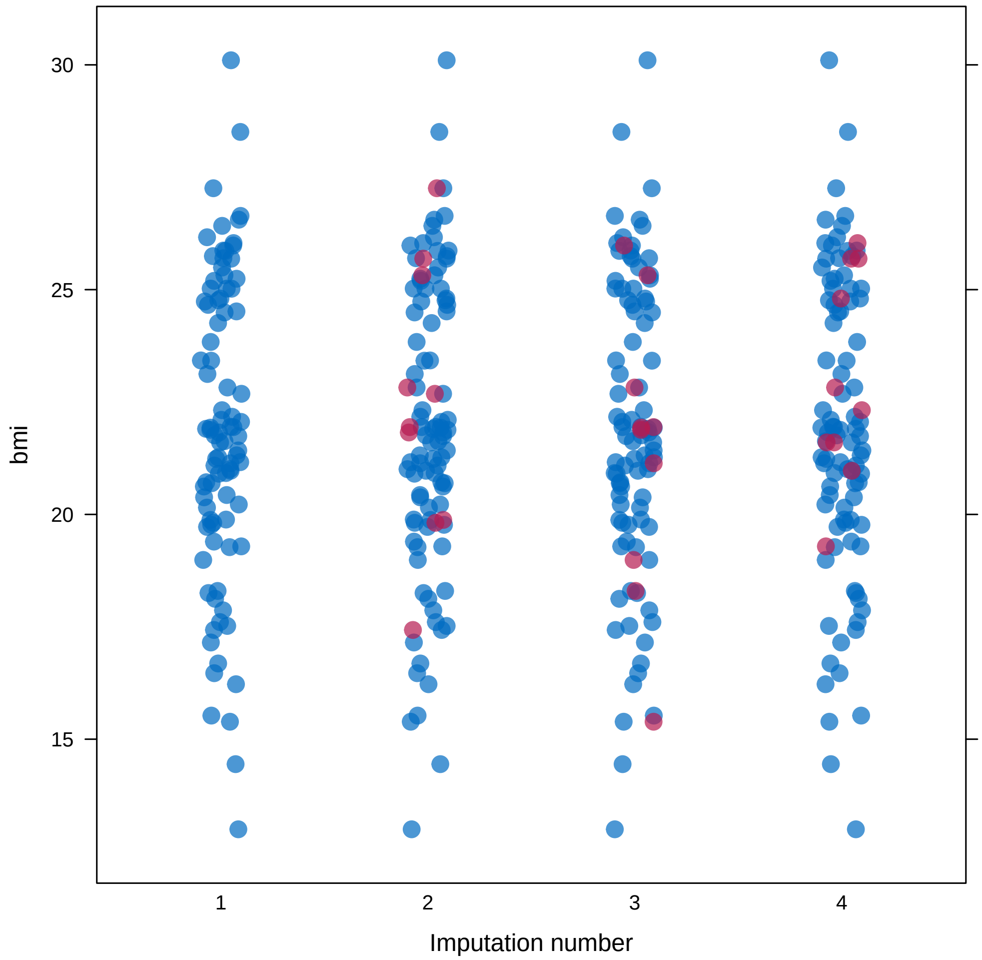
```
gg1.bmi <- ggplot(dt.bmi, aes(bmi, group = imputed, fill = imputed))
gg1.bmi <- gg1.bmi + geom_histogram(aes(y=..count../sum(..count..)),
    position = "dodge")
gg1.bmi
```

```
‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



One more plot:

```
stripplot(dt.mice, bmi∼.imp, pch=20, cex=2)
```


30
25
20
15
bmi
1
2
3
4
Imputation number

2.5 Step 3: Fit the statical model on each imputed dataset

```
e.mice <- with(data = dt.mice,
               lm(Y~group+season+bmi+gender+age)
               )
e.mice
```

```
call :
with.mids(data = dt.mice, expr = lm(Y ~ group + season + bmi +
    gender + age))

call1 :
mice(data = dt.data, m = n.imputed, method = "pmm", predictorMatrix = Mlink,
    maxit = 50, printFlag = FALSE, seed = 500)

nmis :
     Y  group season    bmi gender    age
     0      0      0     10      0      0

analyses :
[[1]]

Call:
lm(formula = Y ~ group + season + bmi + gender + age)

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      0.5208        0.5992        0.7517        0.9735        0.7954        1.0058


[[2]]

Call:
lm(formula = Y ~ group + season + bmi + gender + age)

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      1.2661        0.8914        1.1338        0.9197        0.8447        1.0088


[[3]]

Call:
lm(formula = Y ~ group + season + bmi + gender + age)
```

```
Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      1.1214        0.7458        1.4506        0.9159        0.8573        1.0081
```

Check that using `with`:

```
e.mice$analyses[[1]]
```

```

Call:
lm(formula = Y ~ group + season + bmi + gender + age)

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      0.5208        0.5992        0.7517        0.9735        0.7954        1.0058
```

is equivalent to run the linear regression on the imputed dataset:

```
dt.tempo <- copy(dt.data)
dt.tempo[is.na(bmi), bmi := dt.mice$imp$bmi[,1]]
lm(Y ~ group + season + bmi + gender + age, data  = dt.tempo)
```

```

Call:
lm(formula = Y ~ group + season + bmi + gender + age, data = dt.tempo)

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      0.5208        0.5992        0.7517        0.9735        0.7954        1.0058
```

2.6 Step 4: Pool the results over the imputed datasets

```
ePool.mice <- pool(e.mice)
summary(ePool.mice)
```

```
              estimate   std.error  statistic        df    p.value
(Intercept)  0.9694266 1.332790683   0.727366 52.148057 0.46888374
groupHC      0.7454997 0.379770099   1.963029 30.298012 0.05272075
seasonsummer 1.1120467 0.527089351   2.109788  5.024377 0.03764349
bmi          0.9363468 0.063722009  14.694245 13.428456 0.00000000
genderFemale 0.8324731 0.338852458   2.456742 90.285942 0.01593243
age          1.0075630 0.009578818 105.186567 84.307182 0.00000000
```

The (pooled) estimate is the average of the estimates relative to each imputed dataset:

```
Q.coef <- colMeans(do.call(rbind,lapply(e.mice$analyses, coef)))
Q.coef
```

```
(Intercept)      groupHC seasonsummer          bmi genderFemale          age
  0.9694266    0.7454997    1.1120467    0.9363468    0.8324731    1.0075630
```

The variance is a bit more complex and involves:

- the within-imputation variance (depends on the sample size)

```
covW <- Reduce("+",lapply(e.mice$analyses, vcov))/n.imputed
covW
```

```
              (Intercept)      groupHC  seasonsummer           bmi  genderFemale
(Intercept)   1.568091910 -0.093480148 -0.0399097160 -5.728182e-02 -0.0843633775 -3.366141e
groupHC      -0.093480148  0.115763163  0.0094967612  2.269357e-03  0.0048518076 -4.621780e
seasonsummer -0.039909716  0.009496761  0.1145514144 -1.233739e-03  0.0103324967 -1.316344e
bmi          -0.057281821  0.002269357 -0.0012337388  2.677583e-03  0.0001937303 -3.977686e
genderFemale -0.084363377  0.004851808  0.0103324967  1.937303e-04  0.1133952760  1.912624e
age          -0.003366141 -0.000462178 -0.0001316344 -3.977686e-05  0.0001912624  8.855684e
```

- the between-imputation variance (depends on the amount of missing data)

```
ls.diffCoef <- lapply(e.mice$analyses, function(iI){coef(iI)-Q.coef})
covB <- Reduce("+",lapply(ls.diffCoef,tcrossprod))/(n.imputed-1)
covB
```

```
             [,1]         [,2]          [,3]          [,4]          [,5]          [,6]
[1,]  0.156179320  0.054483744  0.1097704140 -1.235176e-02  0.0120121980  6.112650e-04
[2,]  0.054483744  0.021346623  0.0279984041 -3.933280e-03  0.0036072493  2.167560e-04
[3,]  0.109770414  0.027998404  0.1224538273 -1.033447e-02  0.0110091756  4.141649e-04
[4,] -0.012351758 -0.003933280 -0.0103344679  1.037183e-03 -0.0010436570 -4.777893e-05
[5,]  0.012012198  0.003607249  0.0110091756 -1.043657e-03  0.0010692841  4.613820e-05
[6,]  0.000611265  0.000216756  0.0004141649 -4.777893e-05  0.0000461382  2.397687e-06
```

- the simulation error

```
covE <- covB/n.imputed
covE
```

```
             [,1]         [,2]         [,3]          [,4]          [,5]          [,6]
[1,]  0.052059773  0.018161248  0.036590138 -4.117253e-03  0.0040040660  2.037550e-04
[2,]  0.018161248  0.007115541  0.009332801 -1.311093e-03  0.0012024164  7.225200e-05
[3,]  0.036590138  0.009332801  0.040817942 -3.444823e-03  0.0036697252  1.380550e-04
[4,] -0.004117253 -0.001311093 -0.003444823  3.457278e-04 -0.0003478857 -1.592631e-05
[5,]  0.004004066  0.001202416  0.003669725 -3.478857e-04  0.0003564280  1.537940e-05
[6,]  0.000203755  0.000072252  0.000138055 -1.592631e-05  0.0000153794  7.992289e-07
```

The total variance is:

```
covT <- covW + covB + covE
```

leading to the standard errors:

```
sqrt(diag(covT))
```

```
(Intercept)      groupHC seasonsummer          bmi genderFemale          age 
1.332790683  0.379770099  0.527089351  0.063722009  0.338852458  0.009578818 
```

3 Special case: imputation using a specific law and no covariate

Mice can be adapted in order, for instance, to sample from a uniform distribution or a truncated normal distribution. First define a function able to generate data like:

```
mice.impute.SI_unif <- function(y, ry, ...){ ## truncated normal law
    require(truncnorm)
    n.NA <- sum(ry==FALSE)
    sample <- runif(n.NA, min = 0, max = 1)
    return(cbind(sample))
}
```

or

```
mice.impute.SI_tnorm <- function(y, ry, ...){ ## truncated normal law
    require(truncnorm)
    n.NA <- sum(ry==FALSE)
    sample <- rtruncnorm(n.NA, a = 0, b = 1, mean = 1, sd = 0.1)
    return(cbind(sample))
}
```

Then prepare the matrix indicating which variable should be used during the imputation:

```
impute.var <- c("bmi","group")
Mlink2 <- matrix(0,
                 nrow = length(impute.var),
                 ncol = length(impute.var),
                 dimnames = list(impute.var,impute.var))
Mlink2["bmi","group"] <- 1
Mlink2
```

```
      bmi group
bmi     0     1
group   0     0
```

Then run mice as usual except that the method should correspond to one of the previous functions:

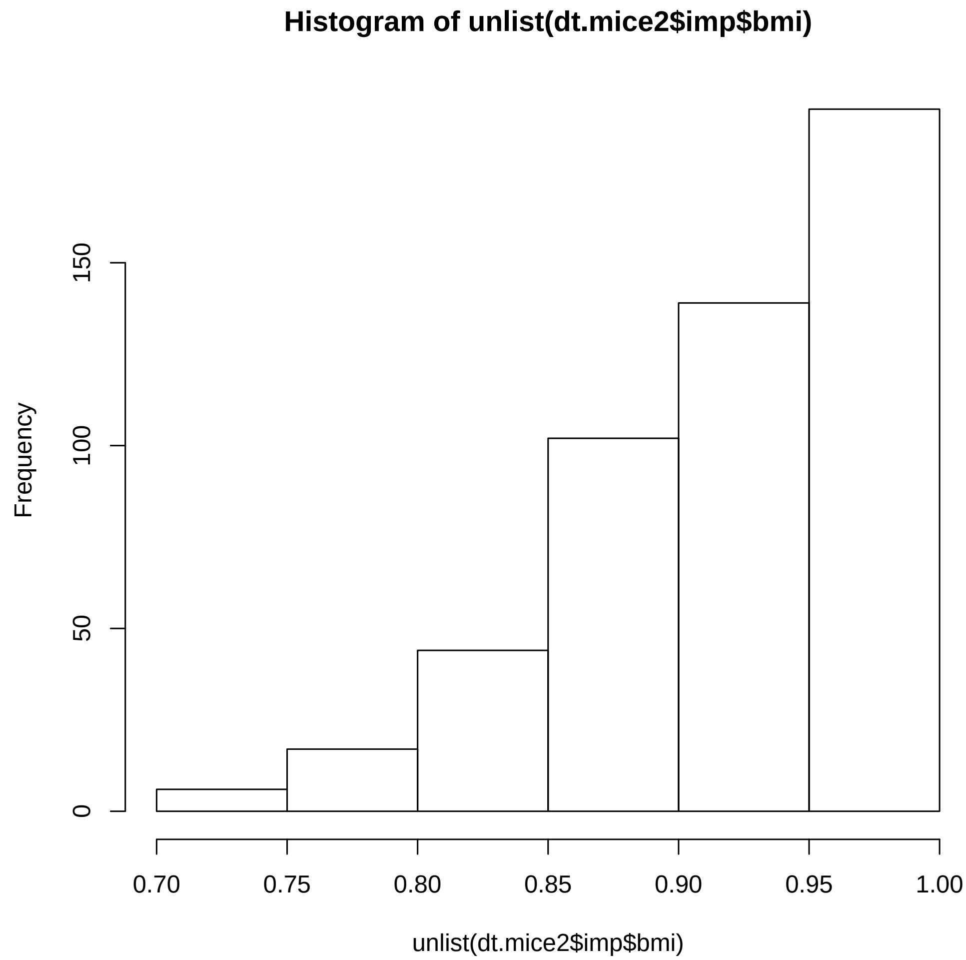
```
n.imputed <- 50 ## number of imputed datasets
set.seed(1)
dt.mice2 <- mice(dt.data,
                 m=n.imputed,
                 maxit = 1, # not relevant
                 predictorMatrix = Mlink2, # not relevant
                 method = 'SI_tnorm', # function previous define (
    without "mice.impute.")
                 seed = 500, printFlag = FALSE)
```

Then as usual one should check that the imputed values are satisfying:

```
quantile(unlist(dt.mice2$imp$bmi))
```

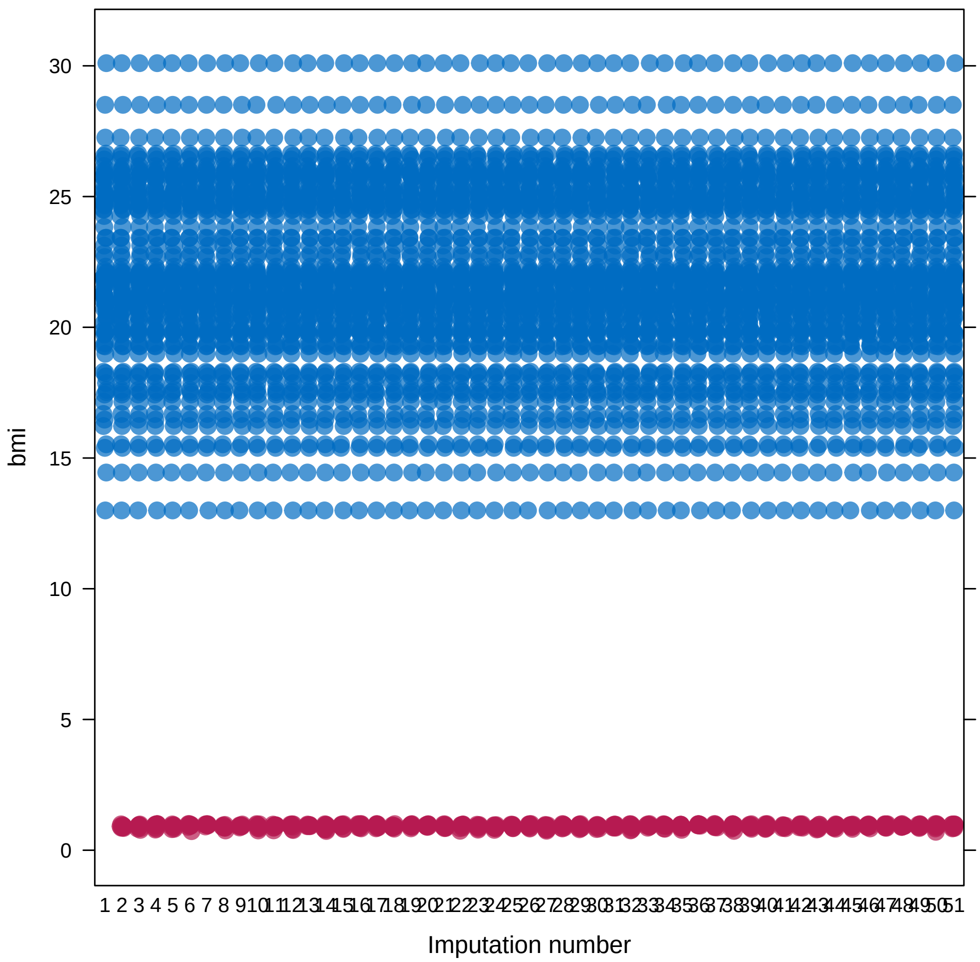
```
       0%       25%       50%       75%      100%
0.7041556 0.8790477 0.9317021 0.9687630 0.9997288
```

```
hist(unlist(dt.mice2$imp$bmi))
```


Histogram of unlist(dt.mice2impbmi)
Frequency
0
50
100
150
0.70
0.75
0.80
0.85
0.90
0.95
1.00
unlist(dt.mice2impbmi)

One more plot:

```
stripplot(dt.mice2, bmi~.imp, pch=20, cex=2)
```



Here for instance the imputed values does not overlap the observed one so something (i.e. the parameters of the distribution used for the imputation) is wrong.

4 Reporting guideline

From https://stefvanbuuren.name/Winnipeg/Lectures/Winnipeg.pdf:

- Amount of missing data
- Reasons for missingness
- Differences between complete and incomplete data
- Method used to account for missing data

- Software
- Number of imputed datasets
- Imputation model
- Derived variables
- Diagnostics
- Pooling
- Listwise deletion
- Sensitivity analysis