A simple example of multiple imputation using the mice package

Brice Ozenne

October 22, 2019

This document gather 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

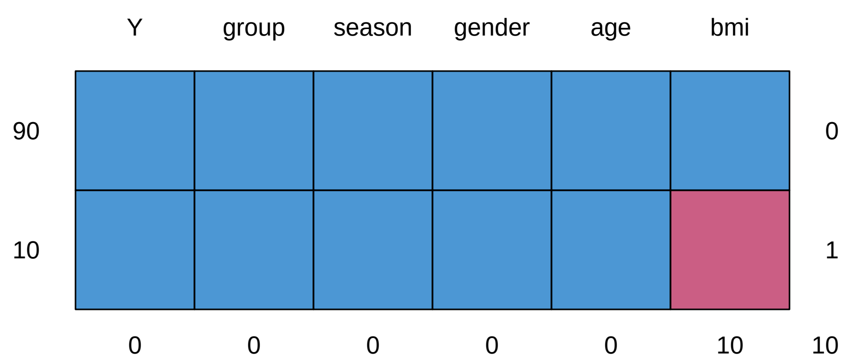
Check the number of missing values in the dataset:

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

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

Missing data patterns:

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



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[c("group","season","gender","age"),"bmi"] <- 1
Mlink
```

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

A value 1 indicates that the column variable was used to impute the row variable, e.g. group was used to impute bmi.

2.3 Step 3: Generate imputed datasets

Generate imputed values

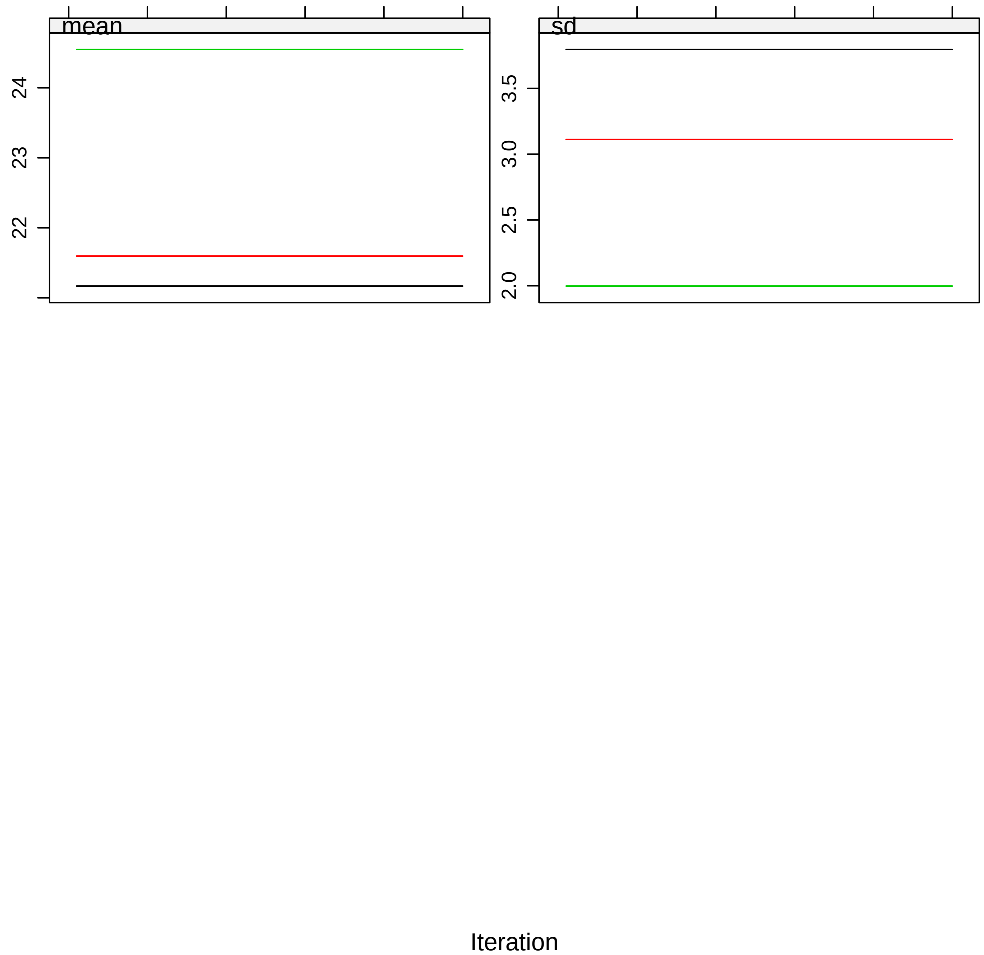
```
n.imputed <- 3 ## number of imputed datasets
dt.mice <- mice(dt.data,
                m=n.imputed,
                maxit = 50, # number of iterations to obtain the
    imputed dataset
                predictorMatrix = Mlink,
                method = 'pmm', # Predictive mean matching, only ok for
     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   1      0   0
season 0     0      0   1      0   0
bmi    0     0      0   0      0   0
gender 0     0      0   1      0   0
age    0     0      0   1      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 20.15124 21.82216 20.90548
2 25.74547 24.76365 25.98147
3 21.27058 15.52519 26.41881
4 14.44499 23.83614 24.66090
```

```
5  24.51607 25.74547 21.08652
6  26.55555 22.05849 24.51607
7  20.97076 20.69408 25.02349
8  22.17178 17.51962 24.76365
9  17.60500 21.94247 26.41881
10 18.25130 22.05849 25.69917
```

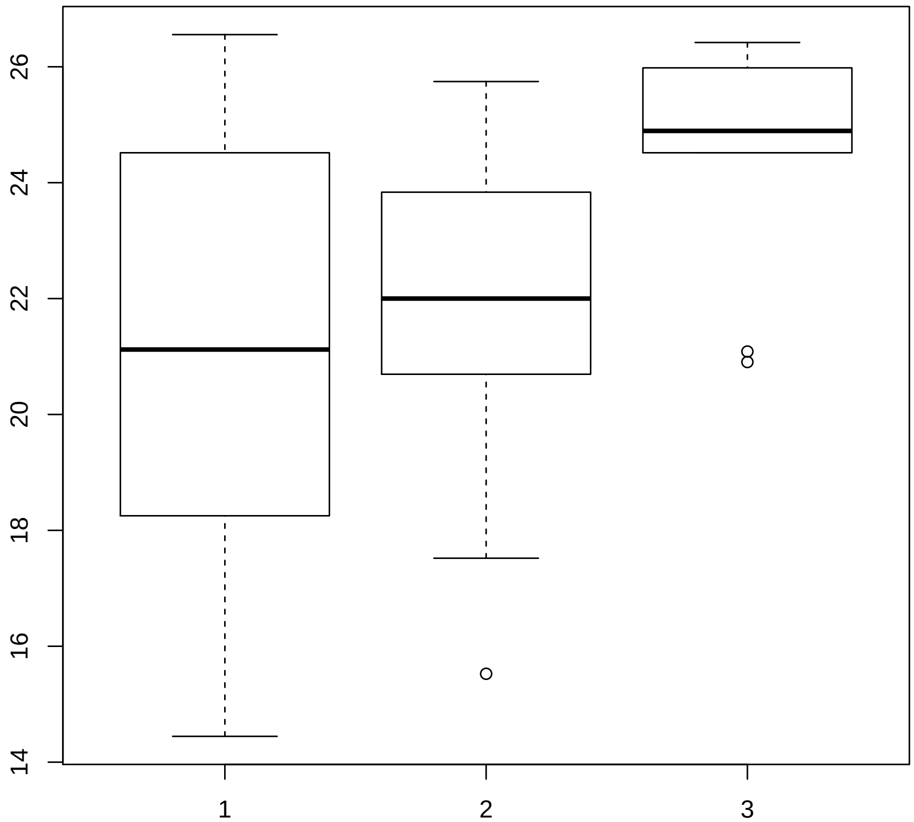
The rows correspond to the 3 different imputed datasets and the columns to 10 imputed values per dataset. One can also summarizes the imputed values computing their quantiles:

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

```
            1        2        3
0%   14.44499 15.52519 20.90548
25%  18.72629 20.97610 24.55228
50%  21.12067 22.00048 24.89357
75%  23.93000 23.39173 25.91090
100% 26.55555 25.74547 26.41881
```

Boxplot of the imputed values:

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



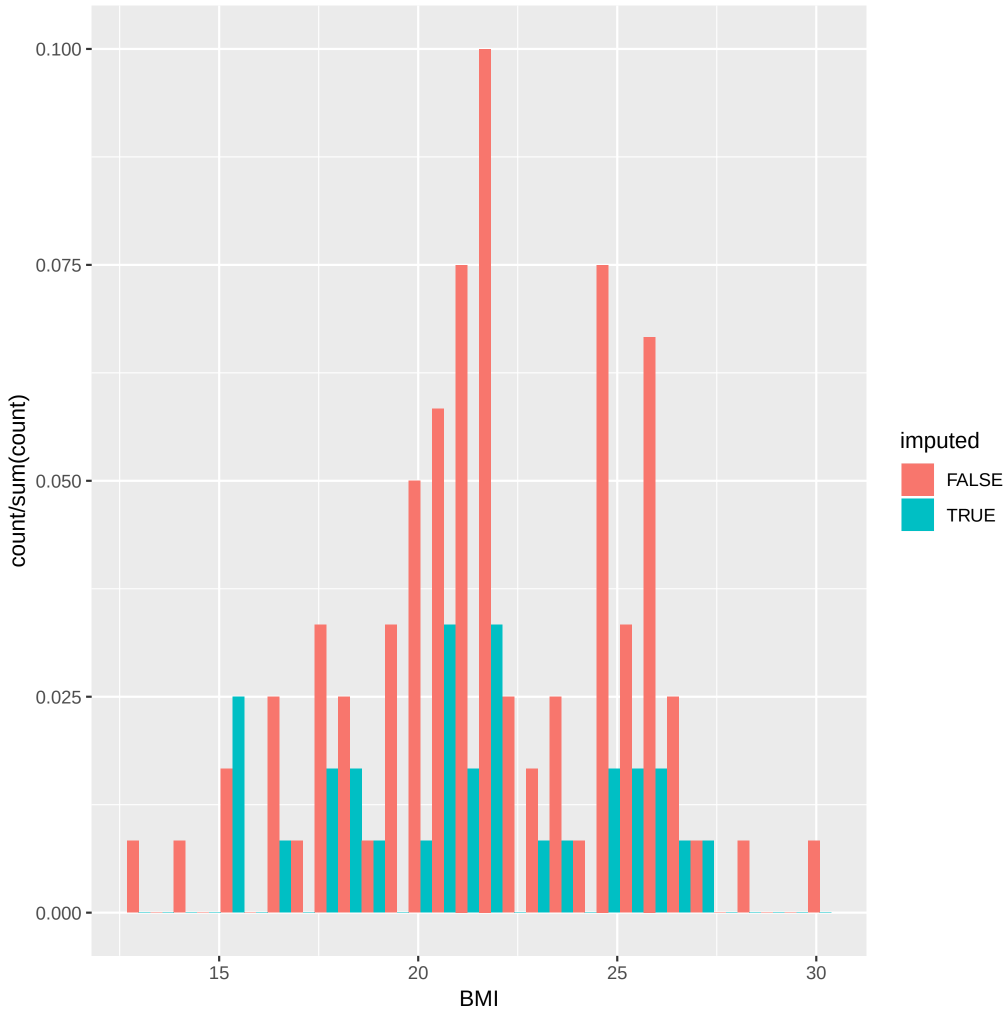
Imputed values vs. observed values

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

Histogram

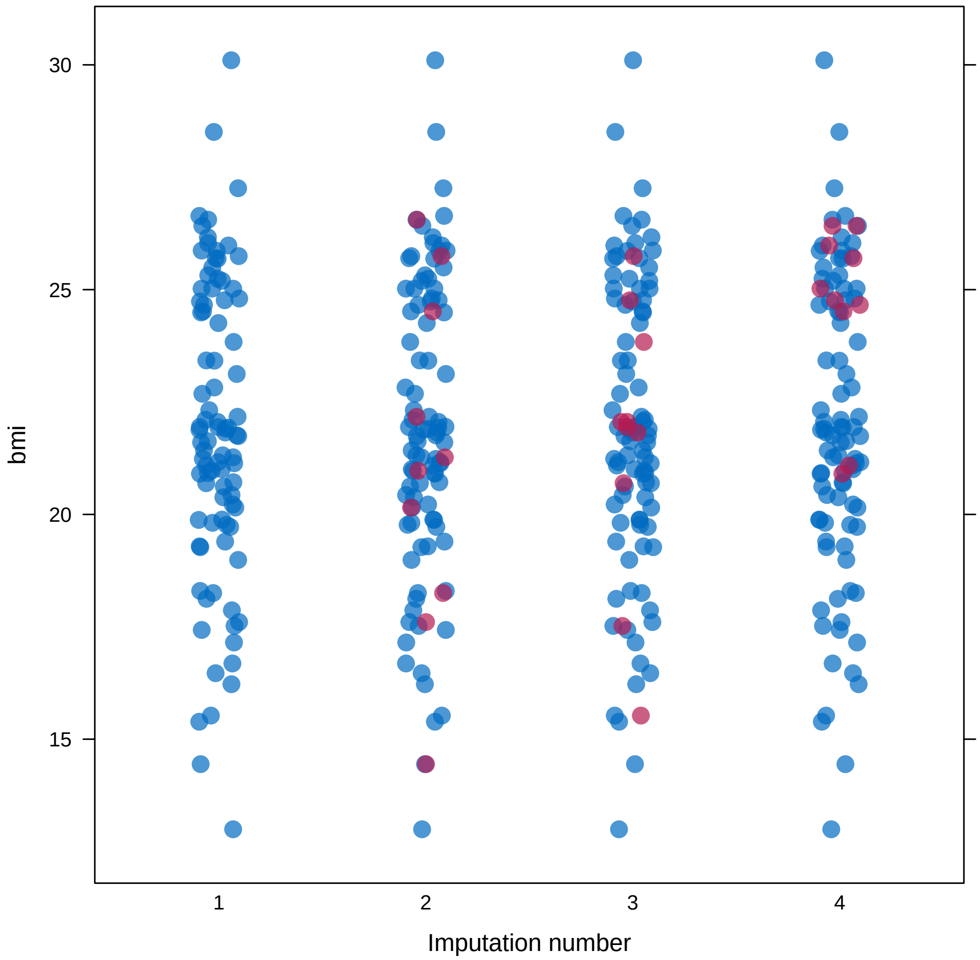
```
gg1.bmi <- ggplot(dt.bmi, aes(bmi, group = imputed, fill = imputed))
gg1.bmi <- gg1.bmi + geom_histogram(aes(y=..count../sum(..count..)),
    position = "dodge")
gg1.bmi
```

```
`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
      2.6716        0.7320        1.1417        0.8566        0.6350        1.0124


[[2]]

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

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age
      2.1738        0.4821        1.2057        0.9066        0.6046        1.0020


[[3]]

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

```
Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age  
      1.4529        0.3860        1.1272        0.9050        0.9807        1.0094  
```

Check that using `with`:

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

```

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

Coefficients:
 (Intercept)       groupHC  seasonsummer           bmi  genderFemale           age  
      2.6716        0.7320        1.1417        0.8566        0.6350        1.0124  
```

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  
      2.6716        0.7320        1.1417        0.8566        0.6350        1.0124  
```

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)  2.0994178 1.53720448  1.365737 27.60999 0.175443034
groupHC      0.5333769 0.42990642  1.240681 24.64629 0.217965151
seasonsummer 1.1581844 0.37942876  3.052442 89.52004 0.002988004
bmi          0.8894300 0.06542205 13.595263 21.64939 0.000000000
genderFemale 0.7401163 0.44568379  1.660631 17.16642 0.100286406
age          1.0079300 0.01217062 82.816644 20.68587 0.000000000
```

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
  2.0994178    0.5333769    1.1581844    0.8894300    0.7401163    1.0079300
```

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)   1.862431644 -0.0994440642 -0.0477903745 -6.786768e-02 -0.0961541467 -4.196538
groupHC      -0.099444064  0.1422982153  0.0119430572  2.070599e-03  0.0056222025 -5.605698
seasonsummer -0.047790374  0.0119430572  0.1416426832 -1.604466e-03  0.0129427682 -1.624194
bmi          -0.067867684  0.0020705988 -0.0016044660  3.202784e-03 -0.0001538781 -4.848575
genderFemale -0.096154147  0.0056222025  0.0129427682 -1.538781e-04  0.1404564390  2.433642
age          -0.004196538 -0.0005605698 -0.0001624194 -4.848575e-05  0.0002433642  1.096841
```

- the between-imputation variance (depends on the amount of missin 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,]  0.3754244681  0.1025439091  0.0070478503 -0.0137914086 -0.1128631208  5.705877e-04
[2,]  0.1025439091  0.0318909867 -0.0005751962 -0.0048485267 -0.0246900921  4.853229e-04
[3,]  0.0070478503 -0.0005751962  0.0017426263  0.0004376612 -0.0060716948 -2.014883e-04
[4,] -0.0137914086 -0.0048485267  0.0004376612  0.0008079455  0.0024361901 -1.127512e-04
[5,] -0.1128631208 -0.0246900921 -0.0060716948  0.0024361901  0.0436332030  3.493617e-04
[6,]  0.0005705877  0.0004853229 -0.0002014883 -0.0001127512  0.0003493617  2.882992e-05
```

- the simulation error

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

```
              [,1]          [,2]          [,3]          [,4]          [,5]          [,6]
[1,]  0.1251414894  0.0341813030  2.349283e-03 -4.597136e-03 -0.0376210403  1.901959e-04
[2,]  0.0341813030  0.0106303289 -1.917321e-04 -1.616176e-03 -0.0082300307  1.617743e-04
[3,]  0.0023492834 -0.0001917321  5.808754e-04  1.458871e-04 -0.0020238983 -6.716278e-05
[4,] -0.0045971362 -0.0016161756  1.458871e-04  2.693152e-04  0.0008120634 -3.758374e-05
[5,] -0.0376210403 -0.0082300307 -2.023898e-03  8.120634e-04  0.0145444010  1.164539e-04
[6,]  0.0001901959  0.0001617743 -6.716278e-05 -3.758374e-05  0.0001164539  9.609975e-06
```

The total variance is:

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

leading to the standard errors:

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

```
(Intercept)      groupHC seasonsummer          bmi genderFemale          age
 1.53720448   0.42990642   0.37942876   0.06542205   0.44568379   0.01217062
```

3 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