Multiple Imputation in R

Alexander McLain

Contents

1 Multiple Imputation using `mice`

The `mice` package implements a method to deal with missing data. The package creates multiple imputations (replacement values) for multivariate missing data. The method is based on Fully Conditional Specification, where each incomplete variable is imputed by a separate model.

The Multivariate Imputation by Chained Equations (MICE) algorithm can impute mixes of continuous, binary, unordered categorical and ordered categorical data. In addition, MICE can impute continuous two-level data, and maintain consistency between imputations by means of passive imputation. Many diagnostic plots are implemented to inspect the quality of the imputations.

```
library(mice)
```

Here, we're going to use the `nhanes` dataset in `mice` to demonstrate how imputation works. This dataset has the following variables:

- `age`: Age group (1=20-39, 2=40-59, 3=60+)
- `bmi`: Body mass index (kg/m**2)
- `hyp`: Hypertensive (1=no,2=yes)
- `chl`: Total serum cholesterol (mg/dL)

Let's look at the data and examine the missingness patterns.

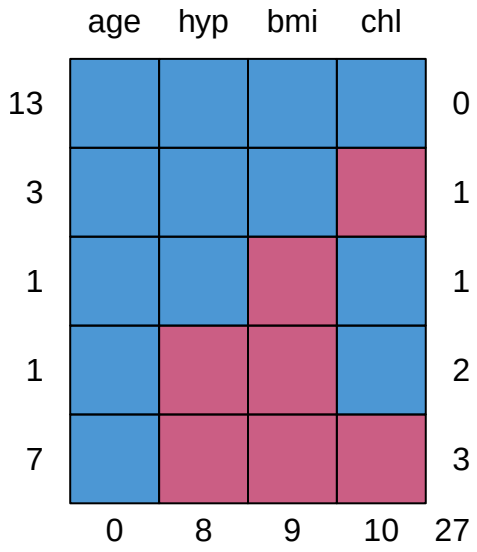
```
head(nhanes)
```

```
##   age  bmi hyp chl
## 1   1   NA  NA  NA
## 2   2 22.7   1 187
## 3   1   NA   1 187
## 4   3   NA  NA  NA
## 5   1 20.4   1 113
## 6   3   NA  NA 184
```

```
summary(nhanes)
```

```
##       age            bmi             hyp             chl       
##  Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0  
##  1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0  
##  Median :2.00   Median :26.75   Median :1.000   Median :187.0  
##  Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4  
##  3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0  
##  Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0  
##                 NA's   :9       NA's   :8       NA's   :10
```

```
md.pattern(nhanes)
```




```
##    age hyp bmi chl    
## 13   1   1   1   1  0
## 3    1   1   1   0  1
## 1    1   1   0   1  1
## 1    1   0   0   1  2
## 7    1   0   0   0  3
##      0   8   9  10 27
```

1.1 Basic Imputation

```
imp <- mice(nhanes)
```

```
##
##  iter imp variable
##   1   1  bmi  hyp  chl
##   1   2  bmi  hyp  chl
##   1   3  bmi  hyp  chl
##   1   4  bmi  hyp  chl
##   1   5  bmi  hyp  chl
##   2   1  bmi  hyp  chl
##   2   2  bmi  hyp  chl
##   2   3  bmi  hyp  chl
##   2   4  bmi  hyp  chl
##   2   5  bmi  hyp  chl
```

```
##   3   1  bmi  hyp  chl
##   3   2  bmi  hyp  chl
##   3   3  bmi  hyp  chl
##   3   4  bmi  hyp  chl
##   3   5  bmi  hyp  chl
##   4   1  bmi  hyp  chl
##   4   2  bmi  hyp  chl
##   4   3  bmi  hyp  chl
##   4   4  bmi  hyp  chl
##   4   5  bmi  hyp  chl
##   5   1  bmi  hyp  chl
##   5   2  bmi  hyp  chl
##   5   3  bmi  hyp  chl
##   5   4  bmi  hyp  chl
##   5   5  bmi  hyp  chl
```

```
imp
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##   age   bmi   hyp   chl
##    "" "pmm" "pmm" "pmm"
## PredictorMatrix:
##     age bmi hyp chl
## age   0   1   1   1
## bmi   1   0   1   1
## hyp   1   1   0   1
## chl   1   1   1   0
```

To look at the imputated data we use

```
imp$imp
```

```
## $age
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $bmi
##       1    2    3    4    5
## 1  29.6 20.4 30.1 27.2 22.0
## 3  27.2 30.1 27.2 27.2 27.2
## 4  24.9 20.4 27.4 24.9 22.0
## 6  24.9 25.5 25.5 25.5 27.4
## 10 22.5 27.5 22.0 28.7 22.5
## 11 29.6 22.0 22.0 35.3 29.6
## 12 24.9 22.7 22.7 26.3 29.6
## 16 25.5 27.2 30.1 30.1 22.7
## 21 22.0 22.0 27.4 28.7 27.5
##
## $hyp
##    1 2 3 4 5
## 1  1 1 1 1 1
## 4  2 1 1 2 1
## 6  2 2 2 2 2
## 10 1 1 1 2 2
```

```
## 11 1 1 1 1 2
## 12 1 1 1 2 2
## 16 1 1 1 1 1
## 21 1 1 1 1 1
##
## $chl
##      1   2   3   4   5
## 1  199 113 186 187 131
## 4  284 184 206 204 206
## 10 238 229 199 238 238
## 11 199 113 131 218 187
## 12 187 187 187 187 199
## 15 131 187 206 229 204
## 16 187 187 204 204 187
## 20 206 186 204 206 204
## 21 131 187 229 238 131
## 24 186 184 206 206 186
```

To extract the complete data we use the `complete` function:

```
c.long <- complete(imp,"long")
head(c.long)
```

```
##   .imp .id age  bmi hyp chl
## 1    1   1   1 29.6   1 199
## 2    1   2   2 22.7   1 187
## 3    1   3   1 27.2   1 187
## 4    1   4   3 24.9   2 284
## 5    1   5   1 20.4   1 113
## 6    1   6   3 24.9   2 184
```

```
tail(c.long)
```

```
##     .imp .id age  bmi hyp chl
## 120    5  20   3 25.5   2 204
## 121    5  21   1 27.5   1 131
## 122    5  22   1 33.2   1 229
## 123    5  23   1 27.5   1 131
## 124    5  24   3 24.9   1 186
## 125    5  25   2 27.4   1 186
```

If we wanted to imput more data we use the `m=` arguement. Here we'll supress the output.

```
imp <- mice(nhanes, m = 20, print = F)
```

1.2 Changing the predictor matrix

The predictor matrix is a square matrix that specifies the variables that are used to impute each incomplete variable. Let us have a look at the predictor matrix that was used

```
imp$pred
```

```
##     age bmi hyp chl
## age   0   1   1   1
## bmi   1   0   1   1
## hyp   1   1   0   1
## chl   1   1   1   0
```

Each variable in the data has a row and a column in the predictor matrix. A value 1 indicates that the column variable was used to impute the row variable. For example, the 1 at entry [bmi, age] indicates that variable age was used to impute the incomplete variable bmi. Note that the diagonal is zero because a variable is not allowed to impute itself. The row of age contains all zeros because there were no missing values in age. mice gives you complete control over the predictor matrix, enabling you to choose your own predictor relations. This can be very useful, for example, when you have many variables or when you have clear ideas or prior knowledge about relations in the data at hand. You can use mice() to give you the initial predictor matrix, and change it afterwards, without running the algorithm. This can be done by typing

```
ini <- mice(nhanes, maxit=0, print=F)
pred <- ini$pred
pred
```

```
##     age bmi hyp chl
## age   0   1   1   1
## bmi   1   0   1   1
## hyp   1   1   0   1
## chl   1   1   1   0
```

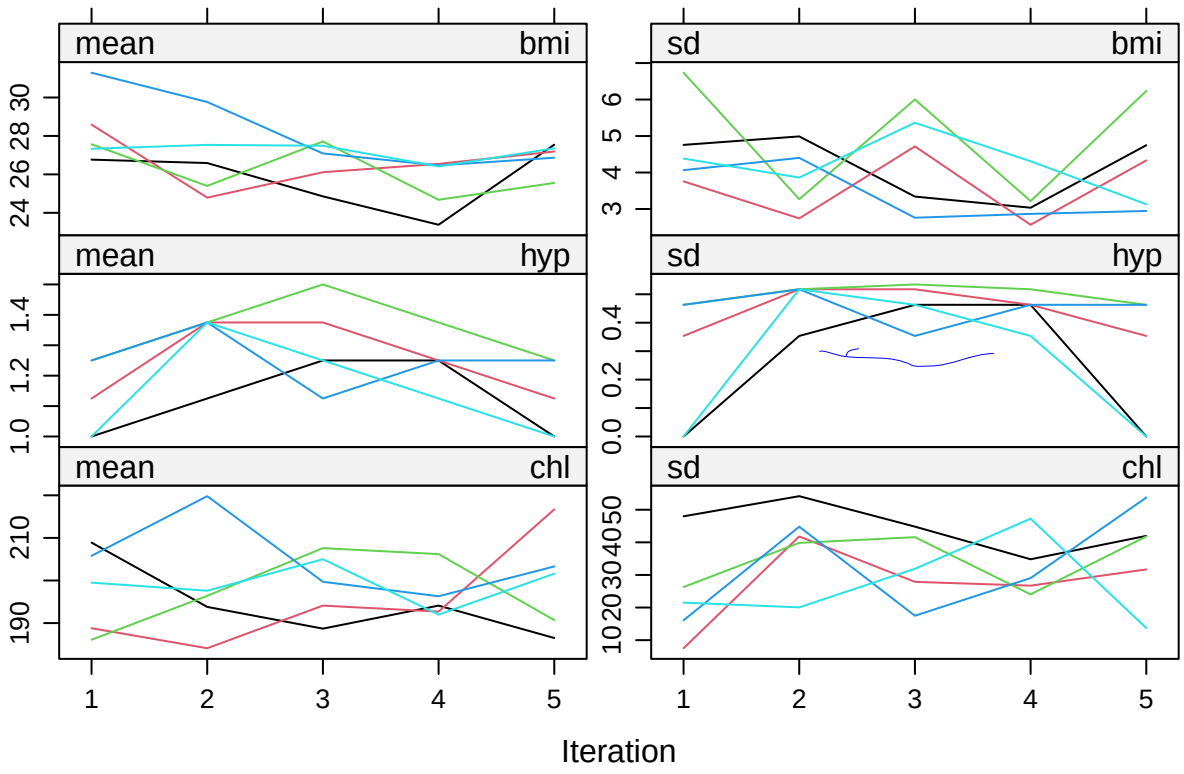
The object pred contains the predictor matrix from an initial run of mice with zero iterations, specified by maxit = 0. Altering the predictor matrix and returning it to the mice algorithm is very simple. For example, the following code removes the variable hyp from the set of predictors, but still leaves it to be predicted by the other variables.

```
pred[ ,"hyp"] <- 0
pred
```

```
##     age bmi hyp chl
## age   0   1   0   1
## bmi   1   0   0   1
## hyp   1   1   0   1
## chl   1   1   0   0
```

1.3 Checking convergenge

```
imp <- mice(nhanes, m = 5, print = F)
plot(imp)
```



The plot shows the mean (left) and standard deviation (right) of the imputed values only. In general, we would like the streams to intermingle and be free of any trends at the later iterations.

1.4 Setting the seed

The algorithm uses random sampling, and therefore, the results will be (perhaps slightly) different if we repeat the imputations with different seeds. In order to get exactly the same result, use the seed argument

```
imp <- mice(nhanes, seed=123, print=F)
```

where 123 is some arbitrary number that you can choose yourself. Rerunning this command will always yields the same imputed values.

1.5 Changing the Method

```
imp$method
```

```
##   age   bmi   hyp   chl
##    "" "pmm" "pmm" "pmm"
```

pmm stands for "Predictive mean matching", which is the default method of imputation.

To do a better job of matching the data, we'll use a dataset that has some data features built in:

```
str(nhanes)
```

```
## 'data.frame':    25 obs. of  4 variables:
##  $ age: num  1 2 1 3 1 3 1 1 2 2 ...
##  $ bmi: num  NA 22.7 NA NA 20.4 NA 22.5 30.1 22 NA ...
##  $ hyp: num  NA 1 1 NA 1 NA 1 1 1 NA ...
##  $ chl: num  NA 187 187 NA 113 184 118 187 238 NA ...
```

```
str(nhanes2)
```

```
## 'data.frame':    25 obs. of  4 variables:
##  $ age: Factor w/ 3 levels "20-39","40-59",..: 1 2 1 3 1 3 1 1 2 2 ...
##  $ bmi: num  NA 22.7 NA NA 20.4 NA 22.5 30.1 22 NA ...
##  $ hyp: Factor w/ 2 levels "no","yes": NA 1 1 NA 1 NA 1 1 1 NA ...
##  $ chl: num  NA 187 187 NA 113 184 118 187 238 NA ...
```

If we do the imputation on `nhanes2` the method will change based on which variable we are imputing:

```
imp <- mice(nhanes2, print=F)
imp$meth
```

```
##       age       bmi       hyp       chl
##        ""    "pmm" "logreg"    "pmm"
```

There are many imputation methods that can be used. See `?mice` for more. To change a method we use the `meth=` arguement:

```
meth <- imp$meth
meth
```

```
##       age       bmi       hyp       chl
##        ""    "pmm" "logreg"    "pmm"
```

```
# "bmi" stands for Bayesian normal linear regression imputation
meth["bmi"] <- "norm"
meth
```

```
##       age       bmi       hyp       chl
##        ""    "norm" "logreg"    "pmm"
```

```
imp <- mice(nhanes2, meth = meth, print=F)
imp$method
```

```
##       age       bmi       hyp       chl
##        ""    "norm" "logreg"    "pmm"
```

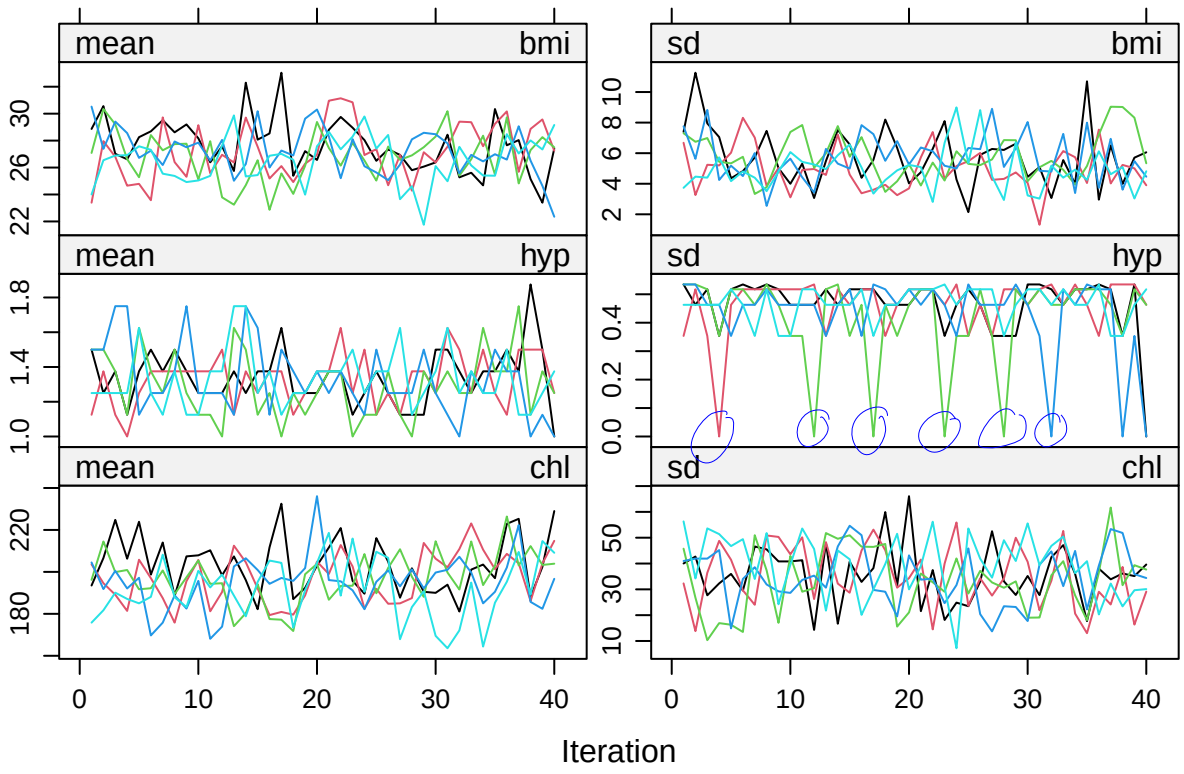
```
head(imp$imp$bmi)
```

```
##          1        2        3        4        5
## 1 37.13370 22.24783 23.94452 22.53471 25.17086
## 3 30.02802 32.73826 33.01491 25.03214 28.83450
## 4 27.05339 24.26349 16.47030 30.31736 21.64611
## 6 20.40314 14.44807 26.14748 32.53472 21.22275
## 10 27.37041 19.56158 21.78650 33.46433 30.12461
## 11 27.79964 27.43660 24.14224 19.06437 28.39525
```

1.6 Running additional iterations

Though using just five iterations (the default) often works well in practice, we need to extend the number of iterations of the mice algorithm to confirm that there is no trend and that the trace lines intermingle well. We can increase the number of iterations to 40 by running 35 additional iterations using the `mice.mids()` function.

```
imp40 <- mice.mids(imp, maxit=35, print=F)
plot(imp40)
```



Now let's do an analysis of the data.

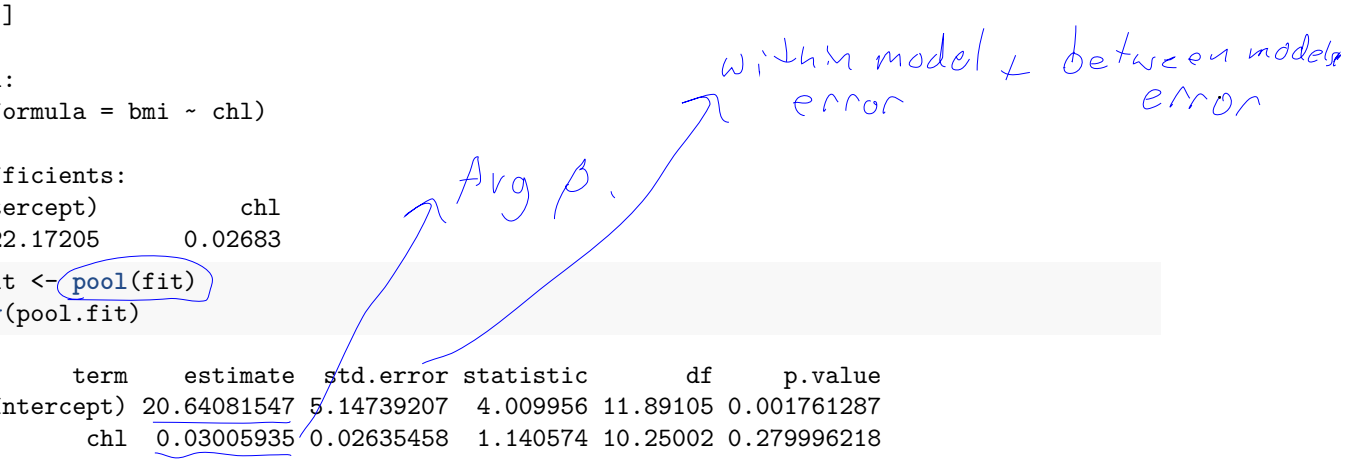
```
fit <- with(imp40, lm(bmi ~ chl))
fit
```

```
## call :
## with.mids(data = imp40, expr = lm(bmi ~ chl))
##
## call1 :
## mice.mids(obj = imp, maxit = 35, printFlag = F)
##
## nmis :
## age bmi hyp chl
##   0   9   8  10
##
## analyses :
## [[1]]
##
## Call:
## lm(formula = bmi ~ chl)
##
## Coefficients:
## (Intercept)          chl
##    20.62578      0.03024
##
##
## [[2]]
##
## Call:
## lm(formula = bmi ~ chl)
```

```
##
## Coefficients:
## (Intercept)          chl
##    22.50259      0.02145
##
##
## [[3]]
##
## Call:
## lm(formula = bmi ~ chl)
##
## Coefficients:
## (Intercept)          chl
##    16.41630      0.05338
##
##
## [[4]]
##
## Call:
## lm(formula = bmi ~ chl)
##
## Coefficients:
## (Intercept)          chl
##    21.48737      0.01839
##
##
## [[5]]
##
## Call:
## lm(formula = bmi ~ chl)
##
## Coefficients:
## (Intercept)          chl
##    22.17205      0.02683
```

```
pool.fit <- pool(fit)
summary(pool.fit)
```

```
##          term    estimate  std.error statistic       df     p.value
## 1 (Intercept) 20.64081547 5.14739207  4.009956 11.89105 0.001761287
## 2         chl  0.03005935 0.02635458  1.140574 10.25002 0.279996218
```



2 Imputing Clustered Data

```
con <- url("https://www.gerkovink.com/mimp/popular.RData")
load(con)
```

This data contains several datasets and functions that, when loaded, are available to you in R. The data we'll use are:

```
head(popNCR)
```

```
##   pupil class extrav  sex texp popular popteach
## 1     1     1      5    1   NA     6.3       NA
## 2     2     1     NA    0   24     4.9       NA
```

```
## 3     3     1      4    1   NA     5.3        6
## 4     4     1      3 <NA>   NA     4.7        5
## 5     5     1      5    1   24      NA        6
## 6     6     1     NA    0   NA     4.7        5
```

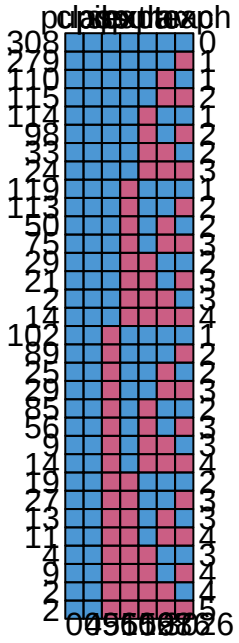
```
dim(popNCR)
```

```
## [1] 2000    7
```

```
summary(popNCR)
```

```
##      pupil           class          extrav         sex           texp
##  Min.   : 1.00   17     :  26   Min.   : 1.000   0   :661   Min.   : 2.0
##  1st Qu.: 6.00   63     :  25   1st Qu.: 4.000   1   :843   1st Qu.: 7.0
##  Median :11.00   10     :  24   Median : 5.000   NA's:496   Median :12.0
##  Mean   :10.65   15     :  24   Mean   : 5.313              Mean   :11.8
##  3rd Qu.:16.00   4      :  23   3rd Qu.: 6.000              3rd Qu.:16.0
##  Max.   :26.00   21     :  23   Max.   :10.000              Max.   :25.0
##                  (Other):1855   NA's   :516                 NA's   :976
##     popular         popteach
##  Min.   :0.000   Min.   : 1.000
##  1st Qu.:3.900   1st Qu.: 4.000
##  Median :4.800   Median : 5.000
##  Mean   :4.829   Mean   : 4.834
##  3rd Qu.:5.800   3rd Qu.: 6.000
##  Max.   :9.100   Max.   :10.000
##  NA's   :510     NA's   :528
```

```
md.pattern(popNCR)
```



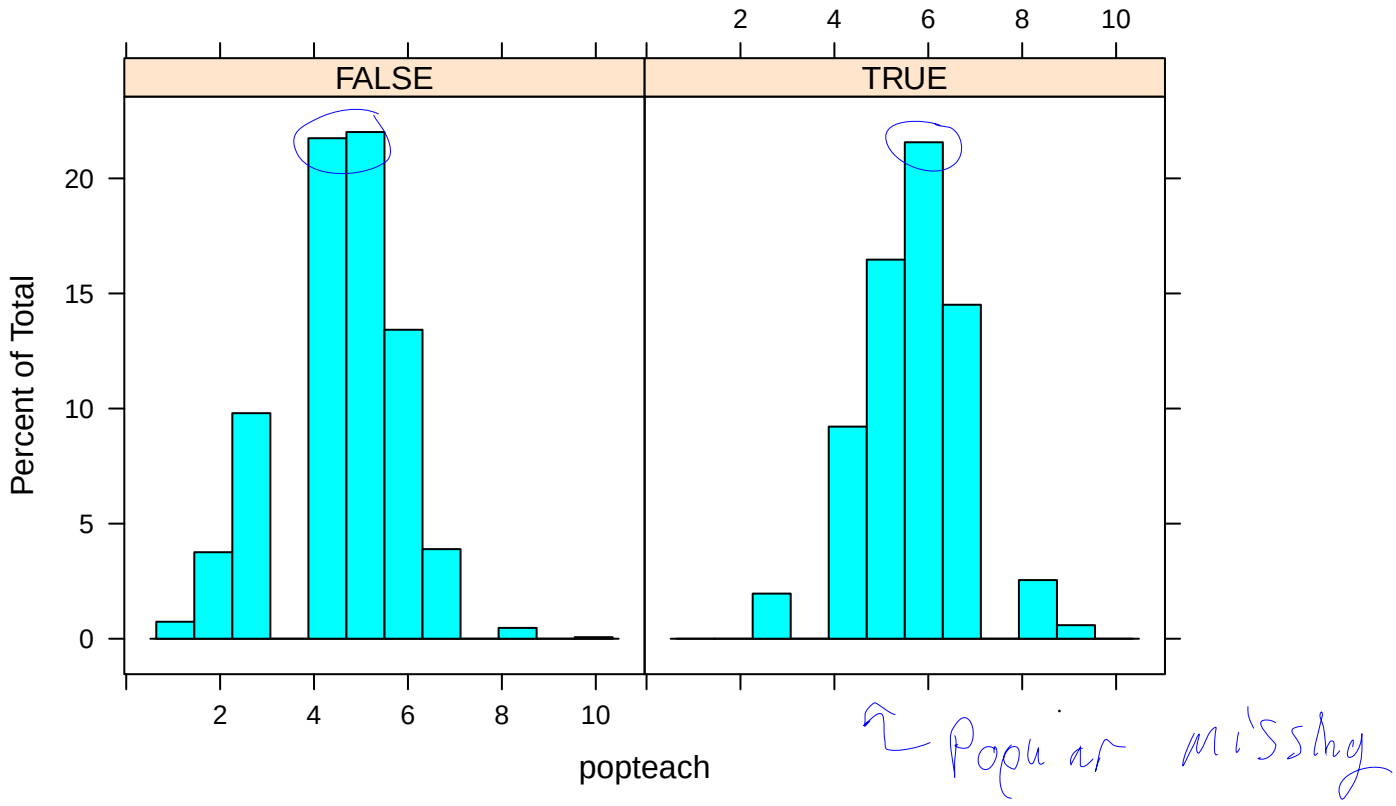
```
##     pupil class sex popular extrav popteach texp
## 308     1     1   1       1      1        1    1    0
## 279     1     1   1       1      1        1    0    1
## 110     1     1   1       1      1        0    1    1
## 115     1     1   1       1      1        0    0    2
## 114     1     1   1       1      0        1    1    1
```

```
## 98      1     1   1       1      0       1    0    2
## 33      1     1   1       1      0       0    1    2
## 24      1     1   1       1      0       0    0    3
## 119     1     1   1       0      1       1    1    1
## 113     1     1   1       0      1       1    0    2
## 50      1     1   1       0      1       0    1    2
## 75      1     1   1       0      1       0    0    3
## 29      1     1   1       0      0       1    1    2
## 21      1     1   1       0      0       1    0    3
## 2       1     1   1       0      0       0    1    3
## 14      1     1   1       0      0       0    0    4
## 102     1     1   0       1      1       1    1    1
## 89      1     1   0       1      1       1    0    2
## 25      1     1   0       1      1       0    1    2
## 29      1     1   0       1      1       0    0    3
## 85      1     1   0       1      0       1    1    2
## 56      1     1   0       1      0       1    0    3
## 9       1     1   0       1      0       0    1    3
## 14      1     1   0       1      0       0    0    4
## 19      1     1   0       0      1       1    1    2
## 27      1     1   0       0      1       1    0    3
## 13      1     1   0       0      1       0    1    3
## 11      1     1   0       0      1       0    0    4
## 4       1     1   0       0      0       1    1    3
## 9       1     1   0       0      0       1    0    4
## 2       1     1   0       0      0       0    1    4
## 2       1     1   0       0      0       0    0    5
##         0     0 496     510    516     528  976 3026
```

2.1 Inspecting Missingness Pattern

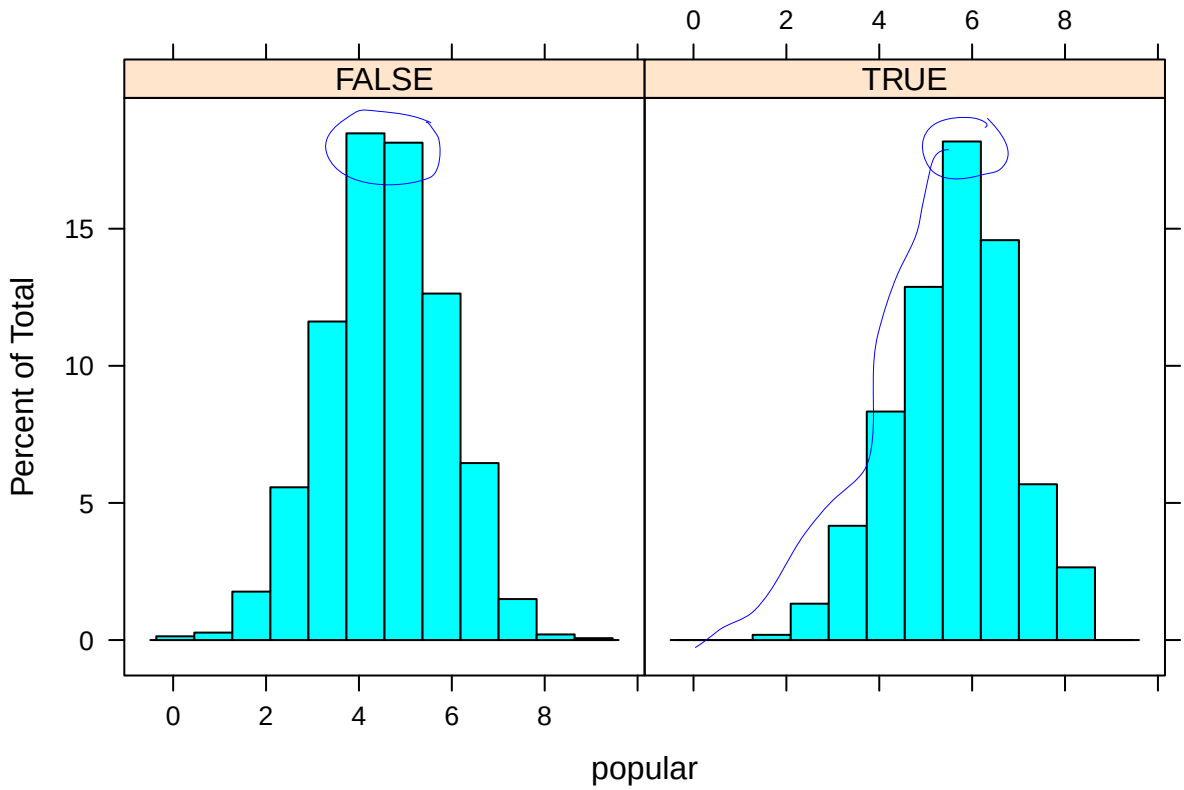
Let's look at if the missing data of popular depend on popteach? One could for example check this by making a histogram of popteach separately for the pupils with known popularity and missing popularity.

```
library(lattice)
histogram(~ popteach | is.na(popular), data=popNCR)
```



Find out if the missingness in teacher popularity depends on pupil popularity.

```
histogram(~ popular | is.na(popteach), data = popNCR)
```



Yes: there is a dependency. The relation seems to be right-tailed.

2.2 Imputing

```
str(popNCR)
```

```
## 'data.frame':    2000 obs. of  7 variables:
##  $ pupil   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ class   : Factor w/ 100 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ extrav  : int  5 NA 4 3 5 NA 5 4 5 5 ...
##  $ sex     : Factor w/ 2 levels "0","1": 2 1 2 NA 2 1 1 1 1 1 ...
##  $ texp    : int  NA 24 NA NA 24 NA 24 NA 24 24 ...
##  $ popular : num  6.3 4.9 5.3 4.7 NA 4.7 5.9 NA NA 3.9 ...
##  $ popteach: int  NA NA 6 5 6 5 5 NA 5 3 ...
```

Impute the `popNCR` dataset with mice using imputation method `norm` for `popular`, `popteach`, `texp` and `extrav`. Exclude pupil as a predictor for all variables.

```
ini <- mice(popNCR, maxit = 0)
meth <- ini$meth
meth
```

```
##    pupil    class   extrav      sex     texp  popular popteach
##       ""       ""    "pmm" "logreg"    "pmm"    "pmm"    "pmm"
```

```
meth[c(3, 5, 6, 7)] <- "norm"
meth
```

```
##    pupil    class   extrav      sex     texp  popular popteach
##       ""       ""   "norm" "logreg"   "norm"   "norm"   "norm"
```

```
pred <- ini$pred
pred
```

```
##          pupil class extrav sex texp popular popteach
## pupil        0     1      1   1    1       1        1
## class        1     0      1   1    1       1        1
## extrav       1     1      0   1    1       1        1
## sex          1     1      1   0    1       1        1
## texp         1     1      1   1    0       1        1
## popular      1     1      1   1    1       0        1
## popteach     1     1      1   1    1       1        0
```

```
pred[, "pupil"] <- 0
pred[, "class"] <- 0
pred
```

```
##          pupil class extrav sex texp popular popteach
## pupil        0     0      1   1    1       1        1
## class        0     0      1   1    1       1        1
## extrav       0     0      0   1    1       1        1
## sex          0     0      1   0    1       1        1
## texp         0     0      1   1    0       1        1
## popular      0     0      1   1    1       0        1
## popteach     0     0      1   1    1       1        0
```

```
imp1 <- mice(popNCR, meth = meth, pred = pred, print = FALSE)
```

Let's compare the means of the imputed and complete case datasets:

```
summary(complete(imp1))
```

```
##      pupil           class          extrav       sex          texp
##  Min.   : 1.00   17     :  26   Min.   : 1.000   0: 985   Min.   :-6.465
##  1st Qu.: 6.00   63     :  25   1st Qu.: 4.139   1:1015   1st Qu.: 8.000
##  Median :11.00   10     :  24   Median : 5.000            Median :12.253
##  Mean   :10.65   15     :  24   Mean   : 5.269            Mean   :12.509
##  3rd Qu.:16.00   4      :  23   3rd Qu.: 6.000            3rd Qu.:16.698
##  Max.   :26.00   21     :  23   Max.   :10.000            Max.   :35.745
##                  (Other):1855
##     popular          popteach
##  Min.   : 0.000   Min.   : 1.000
##  1st Qu.: 4.100   1st Qu.: 4.000
##  Median : 5.000   Median : 5.000
##  Mean   : 5.006   Mean   : 5.021
##  3rd Qu.: 5.971   3rd Qu.: 6.000
##  Max.   :10.547   Max.   :10.000
##
```

```
summary(complete(imp1))
```

```
##      pupil           class          extrav       sex          texp
##  Min.   : 1.00   17     :  26   Min.   : 1.000   0: 985   Min.   :-6.465
##  1st Qu.: 6.00   63     :  25   1st Qu.: 4.139   1:1015   1st Qu.: 8.000
##  Median :11.00   10     :  24   Median : 5.000            Median :12.253
##  Mean   :10.65   15     :  24   Mean   : 5.269            Mean   :12.509
##  3rd Qu.:16.00   4      :  23   3rd Qu.: 6.000            3rd Qu.:16.698
##  Max.   :26.00   21     :  23   Max.   :10.000            Max.   :35.745
##                  (Other):1855
##     popular          popteach
##  Min.   : 0.000   Min.   : 1.000
##  1st Qu.: 4.100   1st Qu.: 4.000
##  Median : 5.000   Median : 5.000
##  Mean   : 5.006   Mean   : 5.021
##  3rd Qu.: 5.971   3rd Qu.: 6.000
##  Max.   :10.547   Max.   :10.000
##
```

```
summary(popNCR[complete.cases(popNCR),])
```

```
##      pupil           class          extrav       sex          texp
##  Min.   : 1.00   87     :  9   Min.   : 3.000   0:154   Min.   : 2.00
##  1st Qu.: 6.00   45     :  8   1st Qu.: 5.000   1:154   1st Qu.: 7.00
##  Median :11.00   73     :  8   Median : 5.000            Median :12.00
##  Mean   :10.72   5      :  7   Mean   : 5.471            Mean   :11.82
##  3rd Qu.:16.00   32     :  7   3rd Qu.: 6.000            3rd Qu.:16.00
##  Max.   :23.00   52     :  7   Max.   :10.000            Max.   :25.00
##                  (Other):262
##     popular         popteach
##  Min.   :0.000   Min.   :1.000
##  1st Qu.:4.200   1st Qu.:4.000
##  Median :5.000   Median :5.000
##  Mean   :4.973   Mean   :5.071
##  3rd Qu.:5.800   3rd Qu.:6.000
##  Max.   :8.000   Max.   :8.000
```

##

Now impute the `popNCR` dataset again with mice using imputation method `norm` for `popular`, `texp`, and `extrav`, but now include `class` as a predictor for all variables.

```
pred <- ini$pred
pred[, "pupil"] <- 0
pred
```

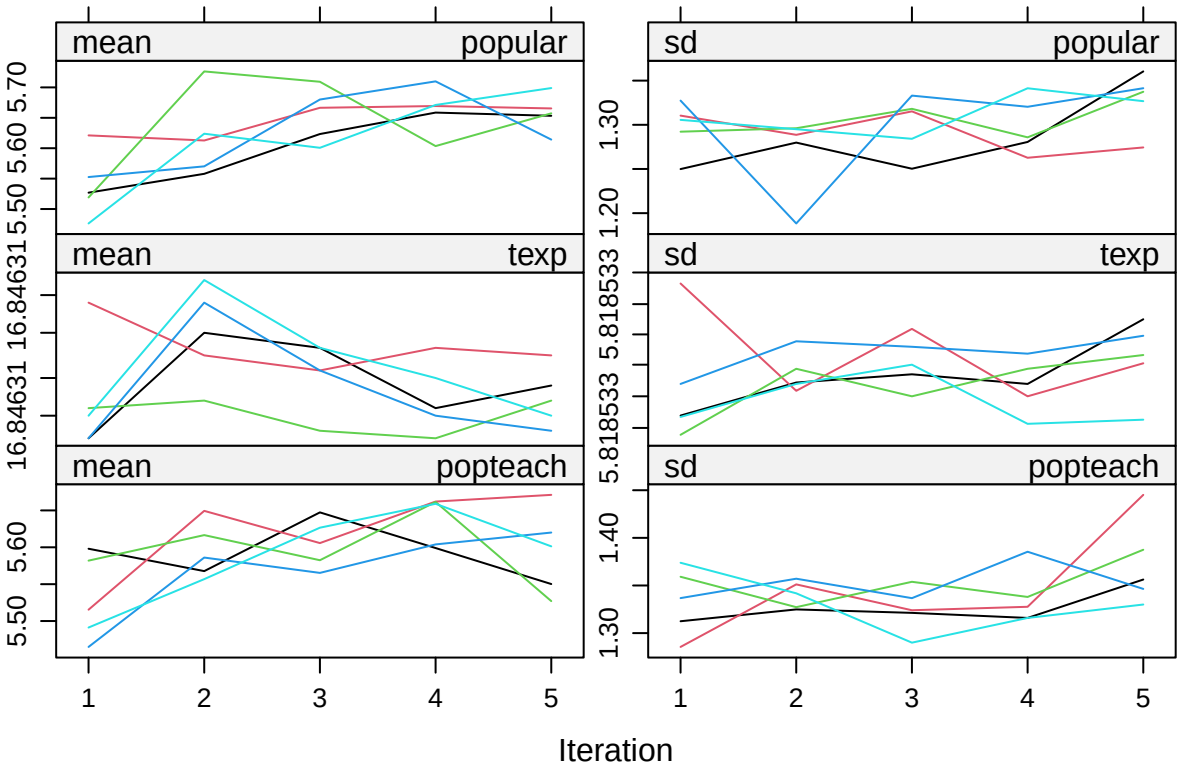
```
##          pupil class extrav sex texp popular popteach
## pupil        0     1      1   1    1       1        1
## class        0     0      1   1    1       1        1
## extrav       0     1      0   1    1       1        1
## sex          0     1      1   0    1       1        1
## texp         0     1      1   1    0       1        1
## popular      0     1      1   1    1       0        1
## popteach     0     1      1   1    1       1        0
```

```
imp2 <- mice(popNCR, meth = meth, pred = pred, print = FALSE, seed = 1234)
```

```
## Warning: Number of logged events: 90
```

Inspect the trace lines for the variables `popular`, `texp`, and `extrav`.

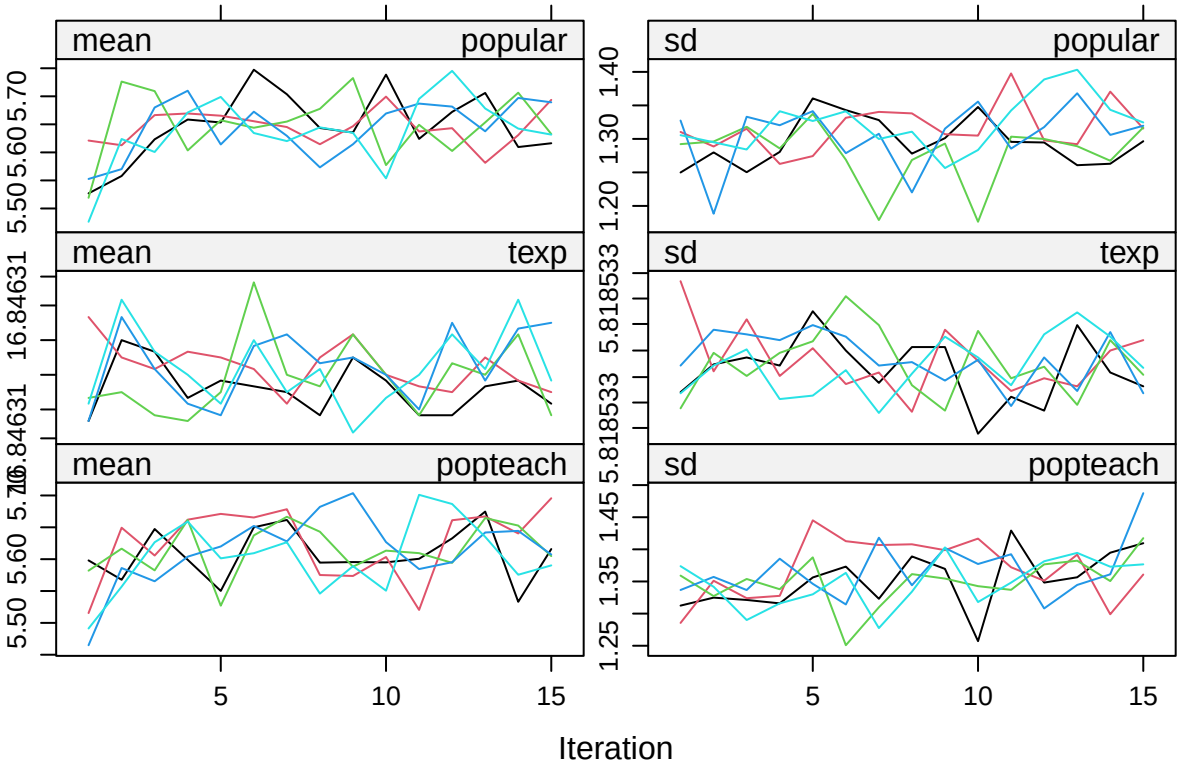
```
plot(imp2, c("popular", "texp", "popteach"))
```



Add another 10 iterations and inspect the trace lines again.

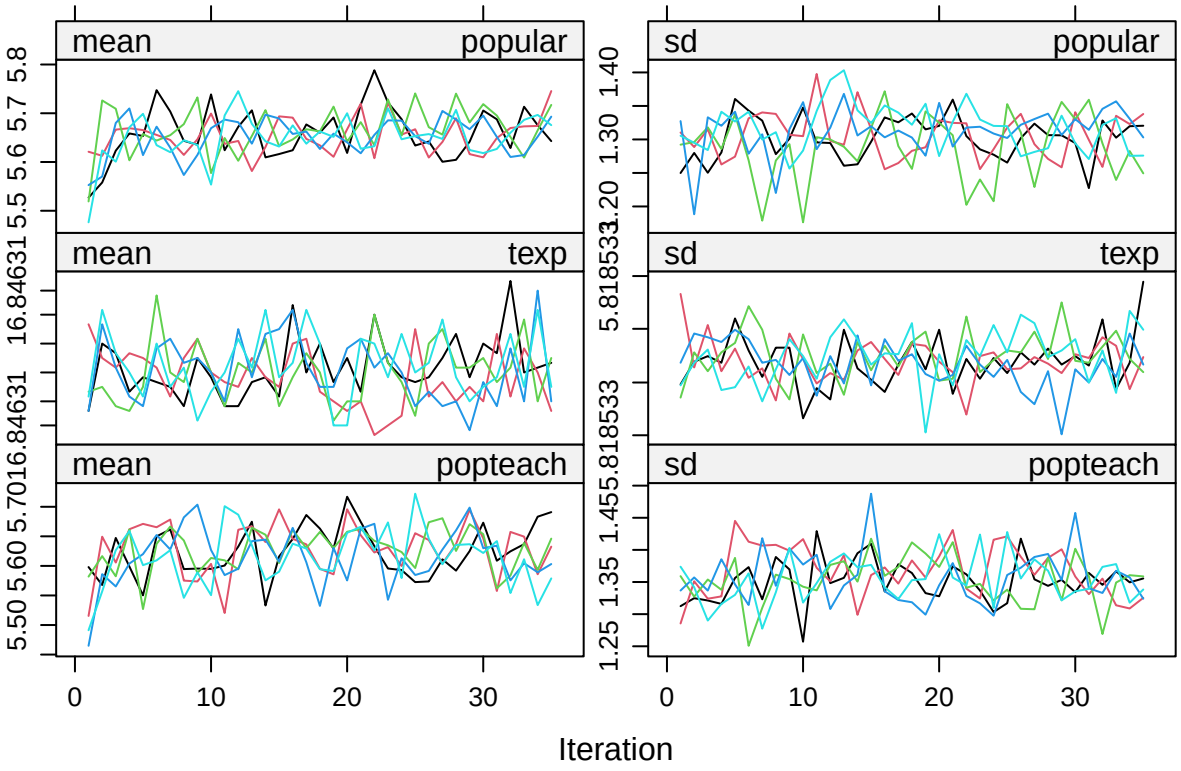
```
imp3 <- mice.mids(imp2, maxit = 10, print = F)
```

```
plot(imp3, c("popular", "texp", "popteach"))
```



It seems OK. Adding another 20 iterations confirms this.

```
imp3b <- mice.mids(imp3, maxit = 20, print = FALSE)
plot(imp3b, c("popular", "texp", "popteach"))
```



Now let's look at the distribution of the variables

```
densityplot(imp3)
```