mice example code

Marta Pineda-Moncusi

2022-11-11

rm(list=ls())

library(haven) data <- read_dta('C:/Users/martapm/Documents/GitHub/Practical_Imputation/multiple-imputation_with_mice/hip.dta')

Tutorial for multiple imputation using mice package

Introduction text: seven steps

#**0. Load and evaluate the data** We are going to use a data set call boys included in the mice package: https://rdrr.io/cran/mice/man/boys.html age: Decimal age (0-21 years) hgt: Height (cm) wgt: Weight (kg) bmi: Body mass index hc: Head circumference (cm) gen: Genital Tanner stage (G1-G5) phb: Pubic hair (Tanner P1-P6) tv: Testicular volume (ml) reg: Region (north, east, west, south, city)

```
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.1.3
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
dim(boys)
```

```
## [1] 748   9
```

```
summary(boys)
```

```
##       age              hgt              wgt              bmi
##  Min.   : 0.035   Min.   : 50.00   Min.   :  3.14   Min.   :11.77
##  1st Qu.: 1.581   1st Qu.: 84.88   1st Qu.: 11.70   1st Qu.:15.90
```

```
##  Median :10.505   Median :147.30   Median : 34.65   Median :17.45
##  Mean   : 9.159   Mean   :132.15   Mean   : 37.15   Mean   :18.07
##  3rd Qu.:15.267   3rd Qu.:175.22   3rd Qu.: 59.58   3rd Qu.:19.53
##  Max.   :21.177   Max.   :198.00   Max.   :117.40   Max.   :31.74
##                   NA's   :20       NA's   :4        NA's   :21
##        hc          gen        phb           tv          reg
##  Min.   :33.70   G1  : 56   P1  : 63   Min.   : 1.00   north: 81
##  1st Qu.:48.12   G2  : 50   P2  : 40   1st Qu.: 4.00   east :161
##  Median :53.00   G3  : 22   P3  : 19   Median :12.00   west :239
##  Mean   :51.51   G4  : 42   P4  : 32   Mean   :11.89   south:191
##  3rd Qu.:56.00   G5  : 75   P5  : 50   3rd Qu.:20.00   city : 73
##  Max.   :65.00   NA's:503   P6  : 41   Max.   :25.00   NA's :  3
##  NA's   :46                 NA's:503   NA's   :522
```

```
str(boys)
```

```
## 'data.frame':    748 obs. of  9 variables:
##  $ age: num  0.035 0.038 0.057 0.06 0.062 0.068 0.068 0.071 0.071 0.073 ...
##  $ hgt: num  50.1 53.5 50 54.5 57.5 55.5 52.5 53 55.1 54.5 ...
##  $ wgt: num  3.65 3.37 3.14 4.27 5.03 ...
##  $ bmi: num  14.5 11.8 12.6 14.4 15.2 ...
##  $ hc : num  33.7 35 35.2 36.7 37.3 37 34.9 35.8 36.8 38 ...
##  $ gen: Ord.factor w/ 5 levels "G1"<"G2"<"G3"<..: NA NA NA NA NA NA NA NA NA NA ...
##  $ phb: Ord.factor w/ 6 levels "P1"<"P2"<"P3"<..: NA NA NA NA NA NA NA NA NA NA ...
##  $ tv : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ reg: Factor w/ 5 levels "north","east",..: 4 4 4 4 4 4 4 3 3 2 ...
```

Check the structure of the data: - How many cases (individuals) are in the data? 748 - How many variables do we have? 10 - Which are numeric? age, hgt, wgt, bmi, hc, tv - Which are factor? gen, phb, reg - Is there any variable with missing? All except age

#**1. Check that variables with missing data are associated with variables that are complete** To do so, crate a dummy variable were missing values are 1 and complete values are 0, and run a logistic regression.

```
#list the variables you want to test
test_var_list = names(boys)[-1]

for (v in 1:length(test_var_list)){
  test_var = ifelse(is.na(boys[[test_var_list[v]]]), 1,0 )
  test = glm(test_var ~ boys$age)
  ci = confint.default(test)
 print(paste(test_var_list[v]," ~ AGE 95%CI: Low ", round(ci[2,1],4),"  High ", round(ci[2,2],4), sep="")
 print(paste(test_var_list[v]," ~ AGE p value: ",   round((summary(test))$coefficients[2,4],5), sep=""))
}
```

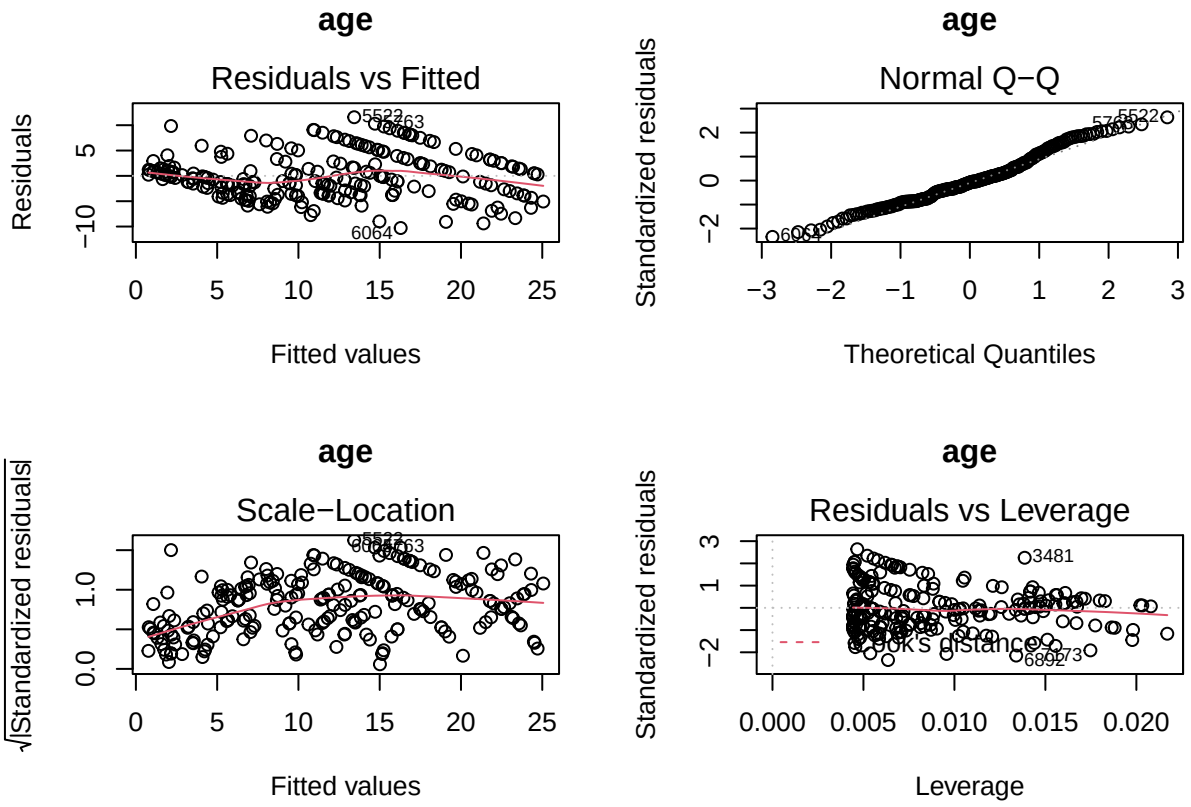
```
## [1] "hgt ~ AGE 95%CI: Low -0.005  High -0.0016"
## [1] "hgt ~ AGE p value: 0.00011"
## [1] "wgt ~ AGE 95%CI: Low -7e-04  High 8e-04"
## [1] "wgt ~ AGE p value: 0.96619"
## [1] "bmi ~ AGE 95%CI: Low -0.0053  High -0.0019"
## [1] "bmi ~ AGE p value: 5e-05"
## [1] "hc ~ AGE 95%CI: Low 0.002  High 0.007"
```

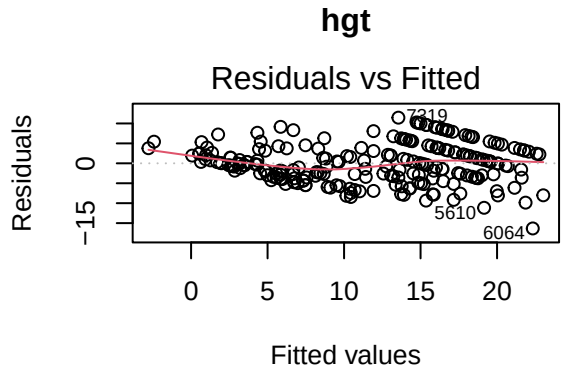
```
## [1] "hc ~ AGE p value: 0.00038"
## [1] "gen ~ AGE 95%CI: Low -0.0379  High -0.0294"
## [1] "gen ~ AGE p value: 0"
## [1] "phb ~ AGE 95%CI: Low -0.0378  High -0.0293"
## [1] "phb ~ AGE p value: 0"
## [1] "tv ~ AGE 95%CI: Low -0.0356  High -0.0272"
## [1] "tv ~ AGE p value: 0"
## [1] "reg ~ AGE 95%CI: Low -0.0013  High 0"
## [1] "reg ~ AGE p value: 0.04873"
```

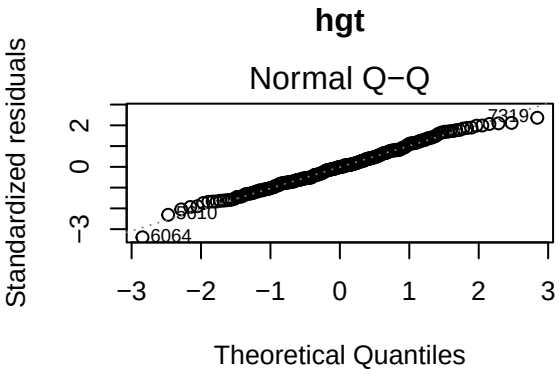
2. Check that your numeric variables are linear with your outcome

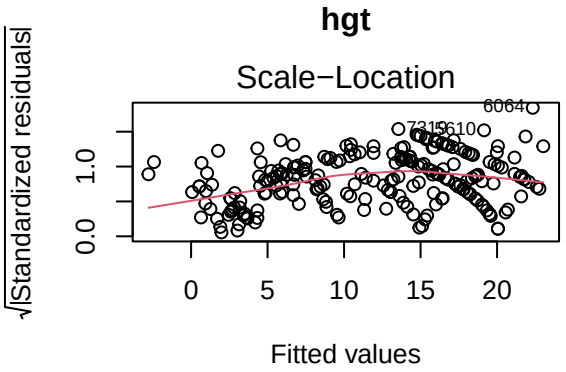
For this exercise, we will consider the variable 'bmi' as the outcome

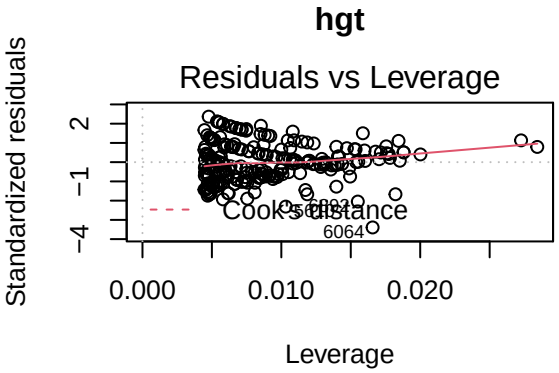
```
#list the variables you want to test
test_var_list = c("age","hgt","wgt","bmi","hc")
outcome = "tv"

for (v in 1:length(test_var_list)){
  df = cbind(boys[test_var_list[v]],outcome = boys[[outcome]])
  test = lm(outcome ~ ., data = df)
  par(mfrow = c(2, 2))
  plot(test,main=paste(test_var_list[v],"\n",sep=""))
}
```

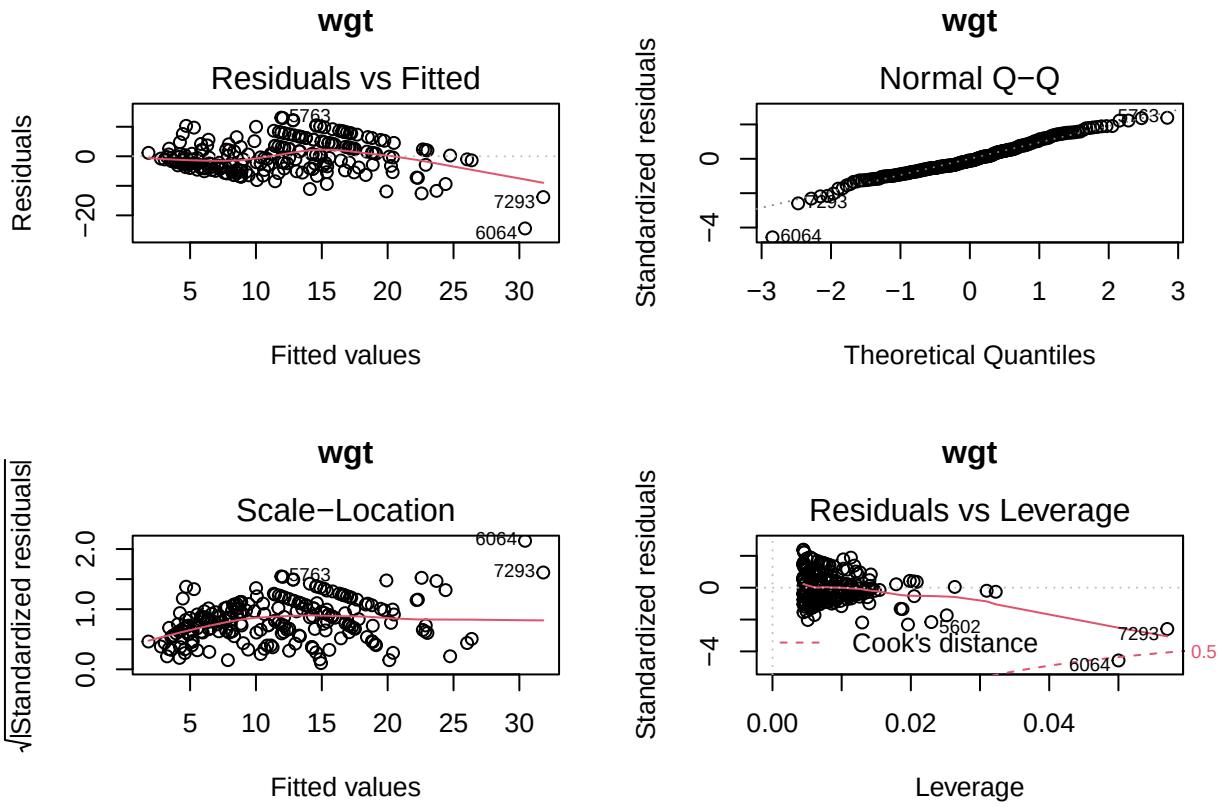


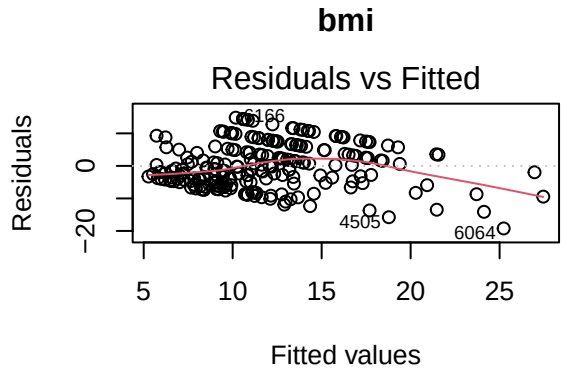

hgt
Residuals vs Fitted
Residuals
Fitted values
7319
5610
6064

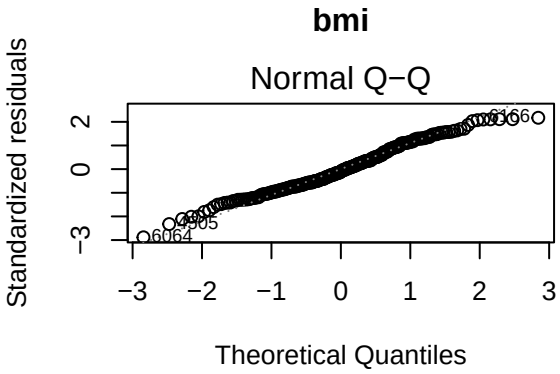

hgt
Normal Q−Q
Standardized residuals
Theoretical Quantiles
7319
5610
6064

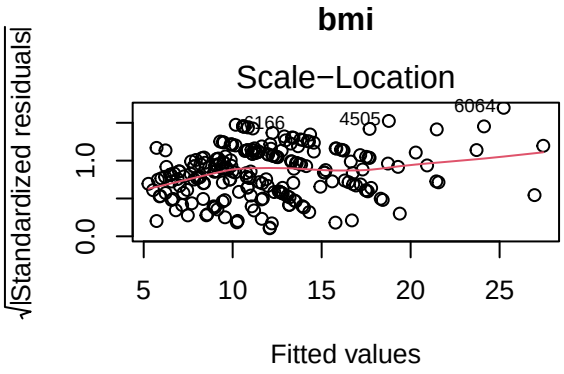

hgt
Scale−Location
√|Standardized residuals|
Fitted values
6064
7319
5610

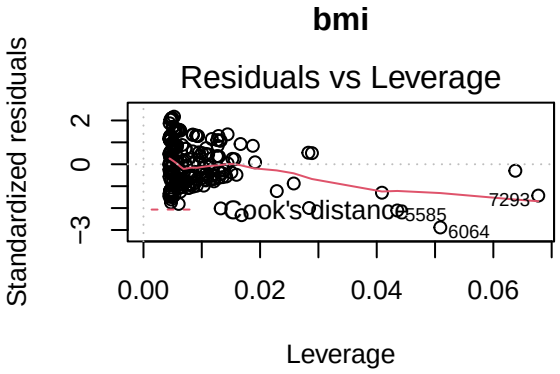

hgt
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
6064

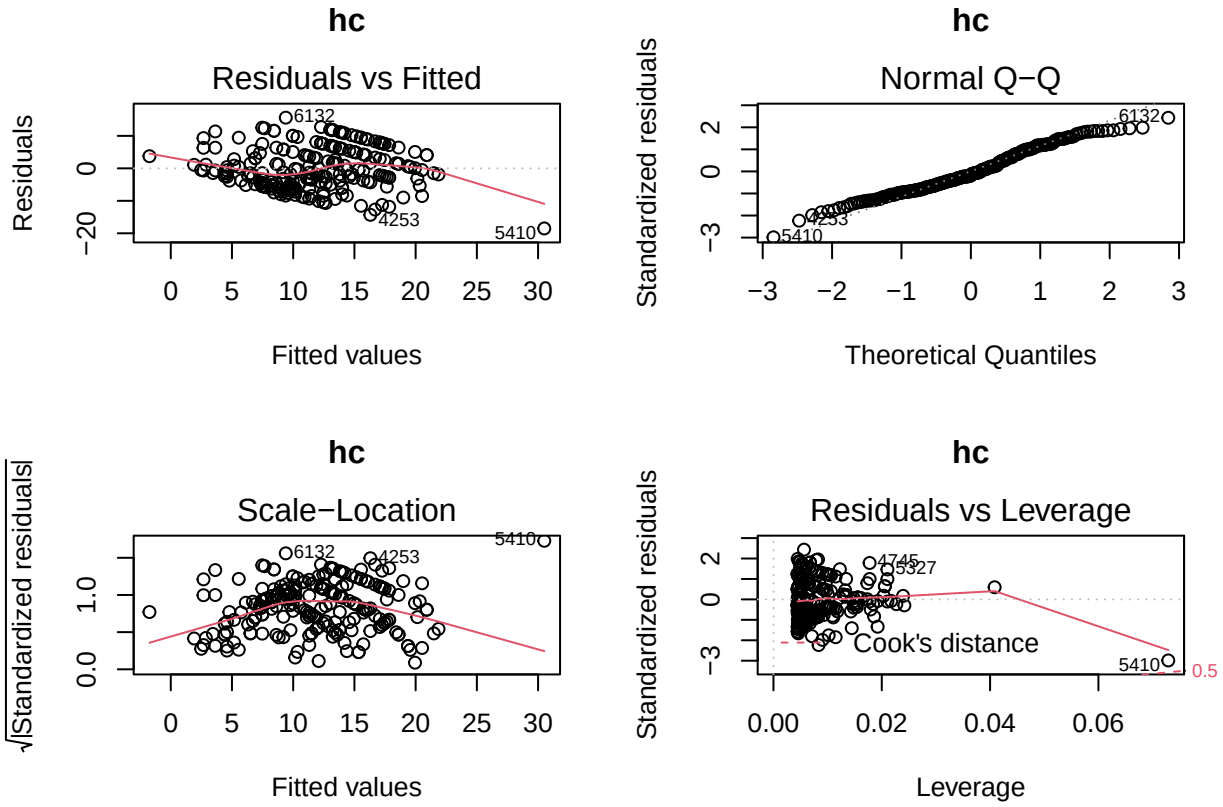

wgt
Residuals vs Fitted
Residuals
Fitted values
5763
7293
6064
wgt
Normal Q−Q
Standardized residuals
Theoretical Quantiles
wgt
Scale−Location
√|Standardized residuals|
Fitted values
wgt
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
5602
0.5


bmi
Residuals vs Fitted
Residuals
Fitted values
6166
4505
6064


bmi
Normal Q−Q
Standardized residuals
Theoretical Quantiles
6166
4505
6064


bmi
Scale−Location
√|Standardized residuals|
Fitted values
6166
4505
6064


bmi
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
7293
5585
6064



How to interpret the plots: The diagnostic plots show residuals in four different ways > **Residuals vs Fitted: Used to check the linear relationship assumptions.** A horizontal line, without distinct patterns is an indication for a linear relationship, what is good. > Normal Q-Q: Used to examine whether the residuals are normally distributed. It's good if residuals points follow the straight dashed line. > Scale-Location (or Spread-Location): Used to check the homogeneity of variance of the residuals (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity. > Residuals vs Leverage: Used to identify influential cases, that is extreme values that might influence the regression results when included or excluded from the analysis.

- Which variables does follow a normal distribution? tv
- Which variables does not follow a normal distribution? age, hgt, wgt, hc
- What should we do? Transform the variables. Simplest e.g. Categorization

#"'{r dplyr} library(dplyr)

#transform to categorical boys %>% mutate(age_cat = ifelse(age>=18,">=18","<18")

) #Include the NA boys %>% mutate(age_cat = ifelse(is.na(age),"NA",age_cat))

ifelse(boys$bmi[!is.na(boys$bmi)]>=18,">=18","<18"), useNA = "always")

#"'

#== 7 Questions structure [empty] ==

#**1. Check the assumption that data is missing at random (MAR)** MICE can handle both MAR and missing not at random (MNAR). Multiple imputation under MNAR requires additional modeling assumptions that influence the generated imputations. [Not covered in this practical]

#"'{r Sensitivity analysis to test MAR or NMAR}

ini <- mice(boys,maxit=0,print=FALSE) post <- ini$post k <- seq(1,1.5,0.1) est <- vector("list",length(k)) for (i in 1:length(k)) { post["bmi"] <- paste("imp[[j]][,i] <-",k[i],"* imp[[j]][,i]") imp <- mice(boys, post=post, seed=2022, print=FALSE, maxit=20) fit <- with(imp, lm(hc~age+bmi+gen+phb+tv+reg)) est[[i]] <- summary(pool(fit)) } pint(est[[1]]) #"'

#**2. Form of the imputation model**

#**3. Select the variables that will be used for imputing the missing values (i.e., set of variables to include as predictors)** # General advice: to include as many relevant variables as possible including their interactions. # Warning: This may however lead to unwieldy model specifications that could easily get out of hand.

#**4. Incorporating variables that are functions of other (incomplete) variables to the imputation.** # Many data sets contain transformed variables, sum scores, interaction variables, ratio's, and so on. It can be useful to incorporate the transformed variables into the multiple imputation algorithm.

#**5. Order in which variables should be imputed.** |—|—|—|—| | Method | Description | Scale type | Default | |—|—|—|—| | pmm | Predictive mean matching | numeric | Y | | norm | Bayesian linear regression | numeric | | | norm.nob | Linear regression, non-Bayesian | numeric | | | mean | Unconditional mean imputation | numeric | | | 2L.norm | Two-level linear model | numeric | | | logreg | Logistic regression | factor, 2 levels | Y | | polyreg | Multinomial logit model | factor, >2 levels | Y | | polr | Ordered logit model | ordered, >2 levels | Y | | lda | Linear discriminant analysis | factor | | | sample | Random sample from the observed data | any | | |—|—|—|—|

```
a<- c("a","b")
```

#**6. Setup of the starting imputations and the number of iterations.** The convergence of the MICE algorithm can be monitored in many ways.

#**7. Number of multiply imputed data sets (m):** Thumps up rule: 10% NA = 10 imputed data sets, 20% NA = 20 imputed datasets, etc.

Including Plots

You can also embed plots, for example:

#`{r pressure, echo=FALSE} #plot(pressure) #`

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.