Visualisation and Imputation of Missing Values

Alexander Kowarik (Statistics Austria), Matthias Templ (ZHAW Winterthur)

July 2017

Outline / R Package

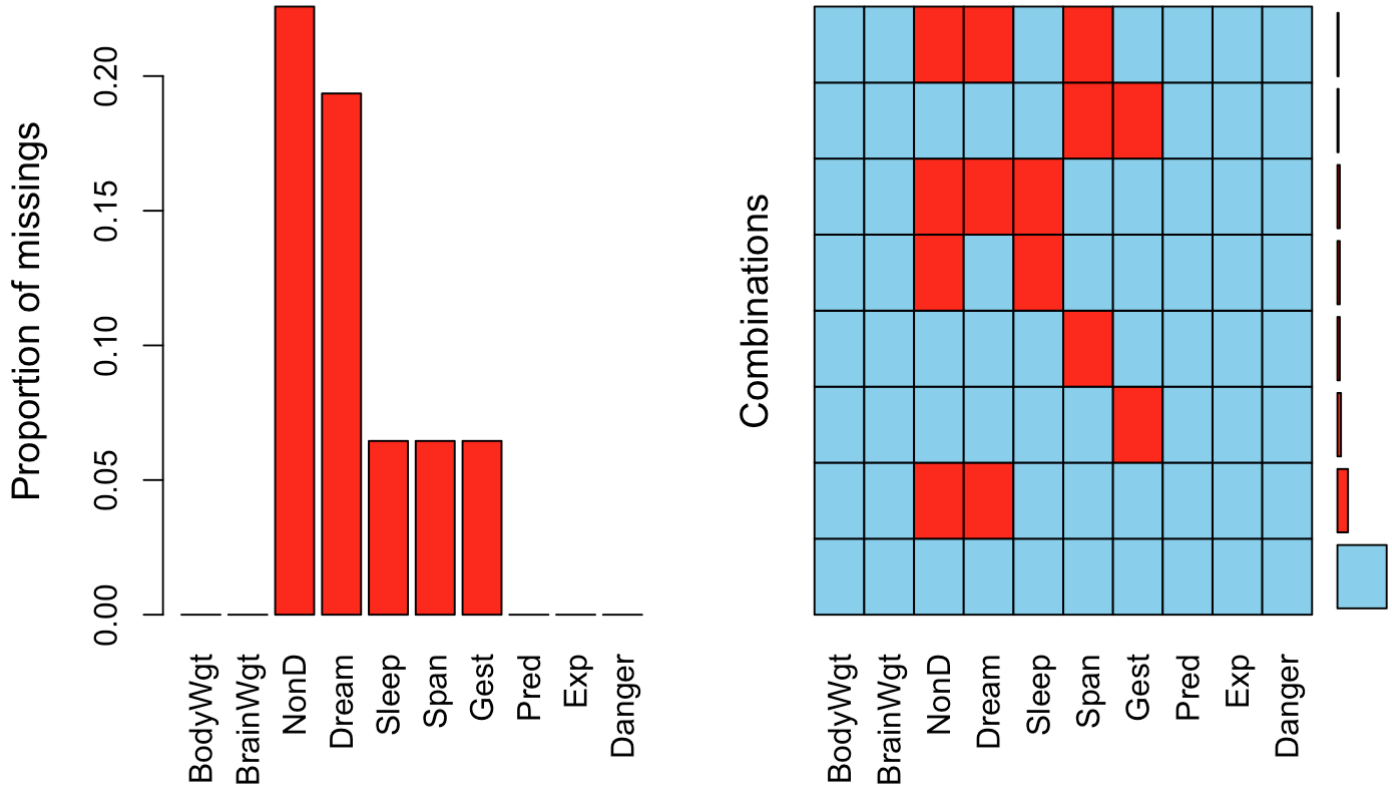
- Content:
  - Tools for visualization of missing data structures (and imputed values)
  - Tools for imputation
- Current CRAN version 4.7.0
- Development version and issue tracking on github https://github.com/statistikat/VIM
- This presentation and the R code https://github.com/alexkowa/VIM_ISI2017
- JSS paper on imputation of missing values with VIM, Kowarik, Templ
- Advances in Data Analysis and Classification paper on visualization with VIM, Templ, Alfons, Filzmoser

Visualisation of Missing Data

- Always important: knowledge about the structure of missing values. Visualisation vs statistical tests.
- literature with focus on visualization of missing data is sparse
- only a few visualization tools missing data
- R package VIM supports the visualization (also with a GUI).

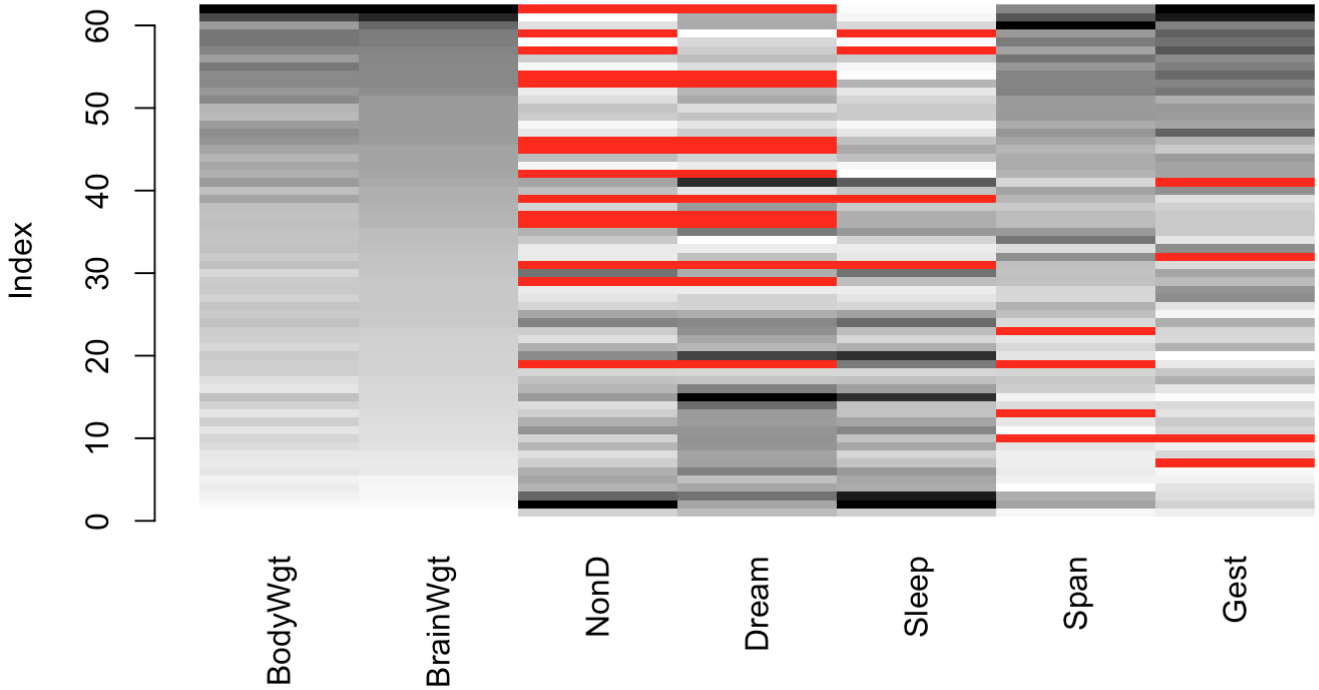
Aggregation Plots

```
aggr(sleep)
```



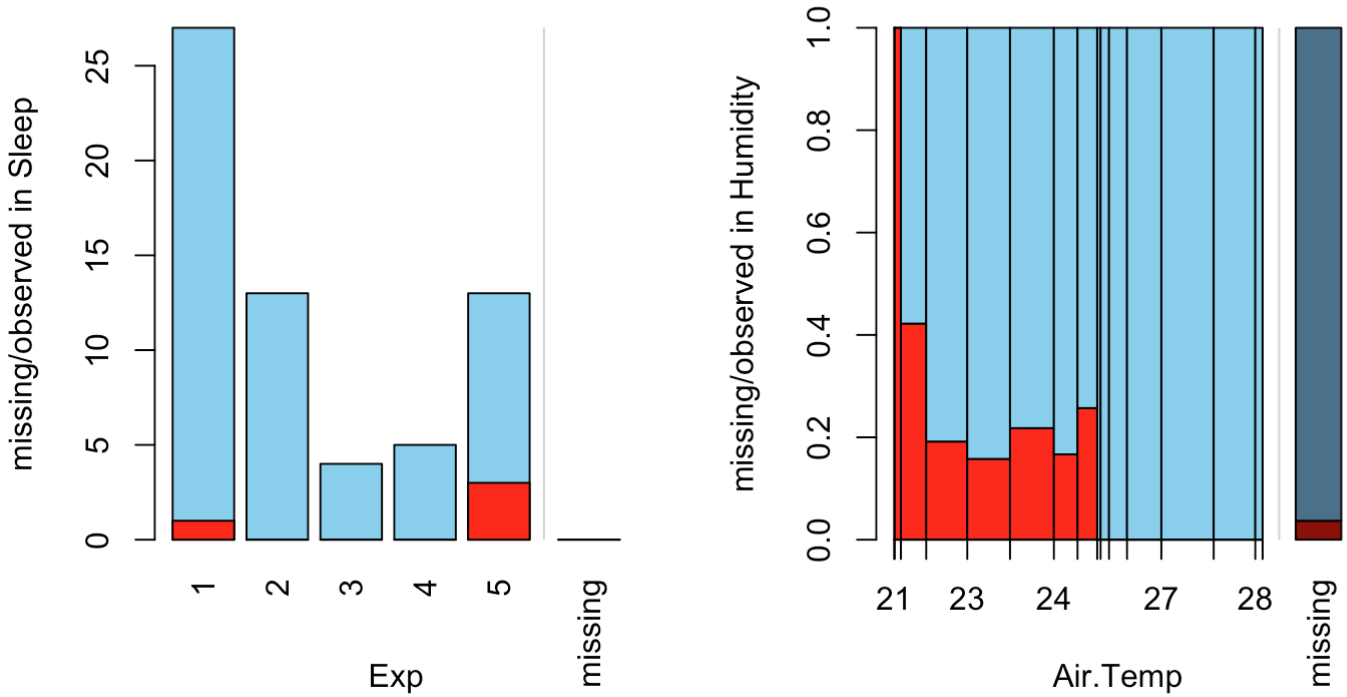
Missing Values in Matrix Form

```
matrixplot(x, sortby = "BrainWgt")
```



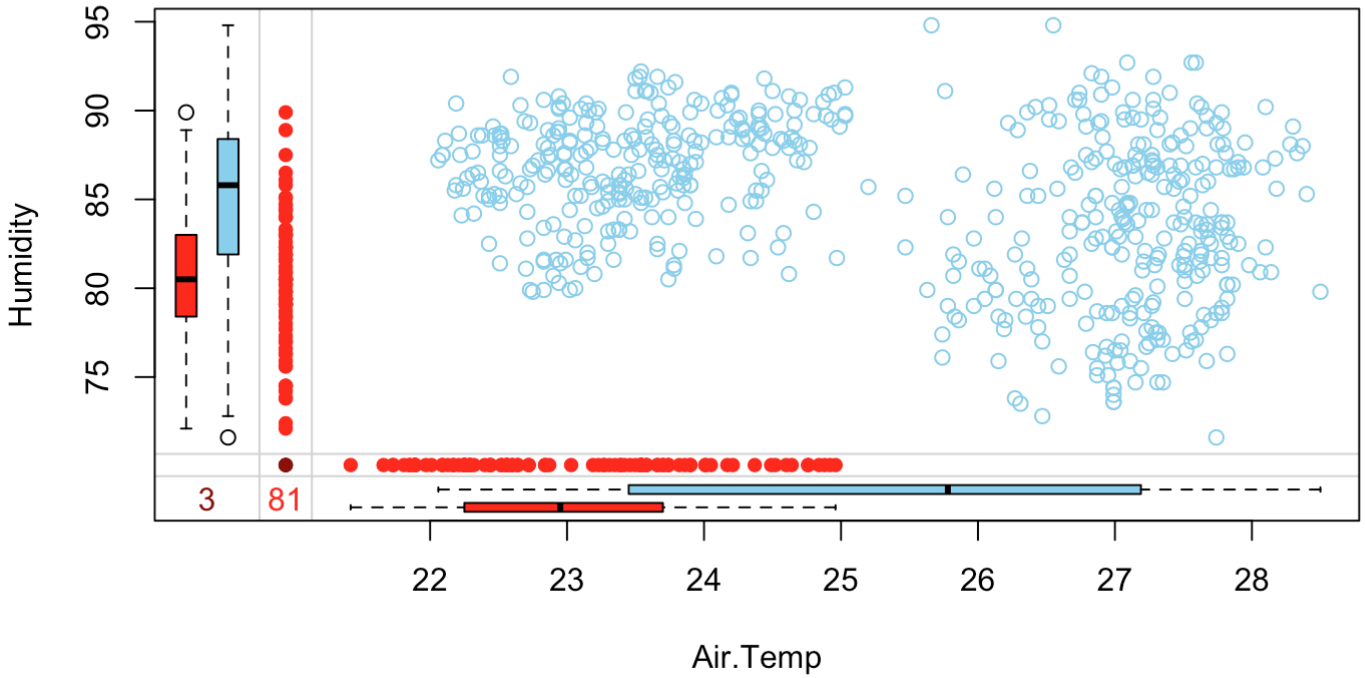
Univariate Plots

```
par(mfrow=c(1,2)); histMiss(x2); spineMiss(x3)
```



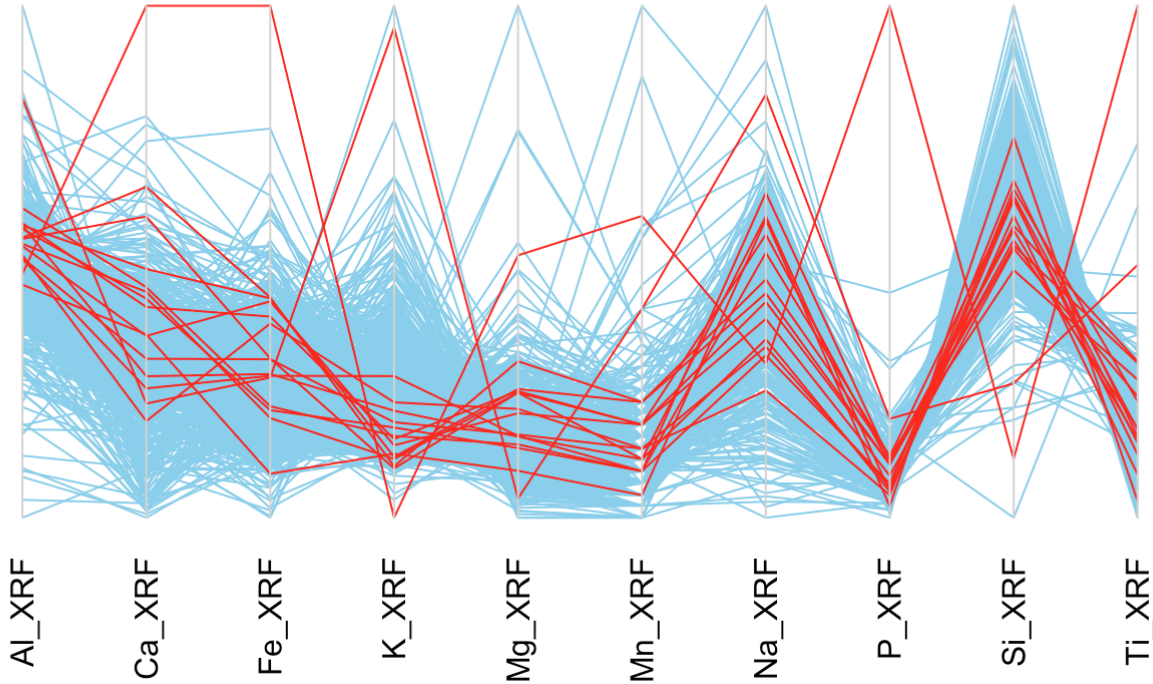
Bivariate Plots

```
marginplot(x3)
```



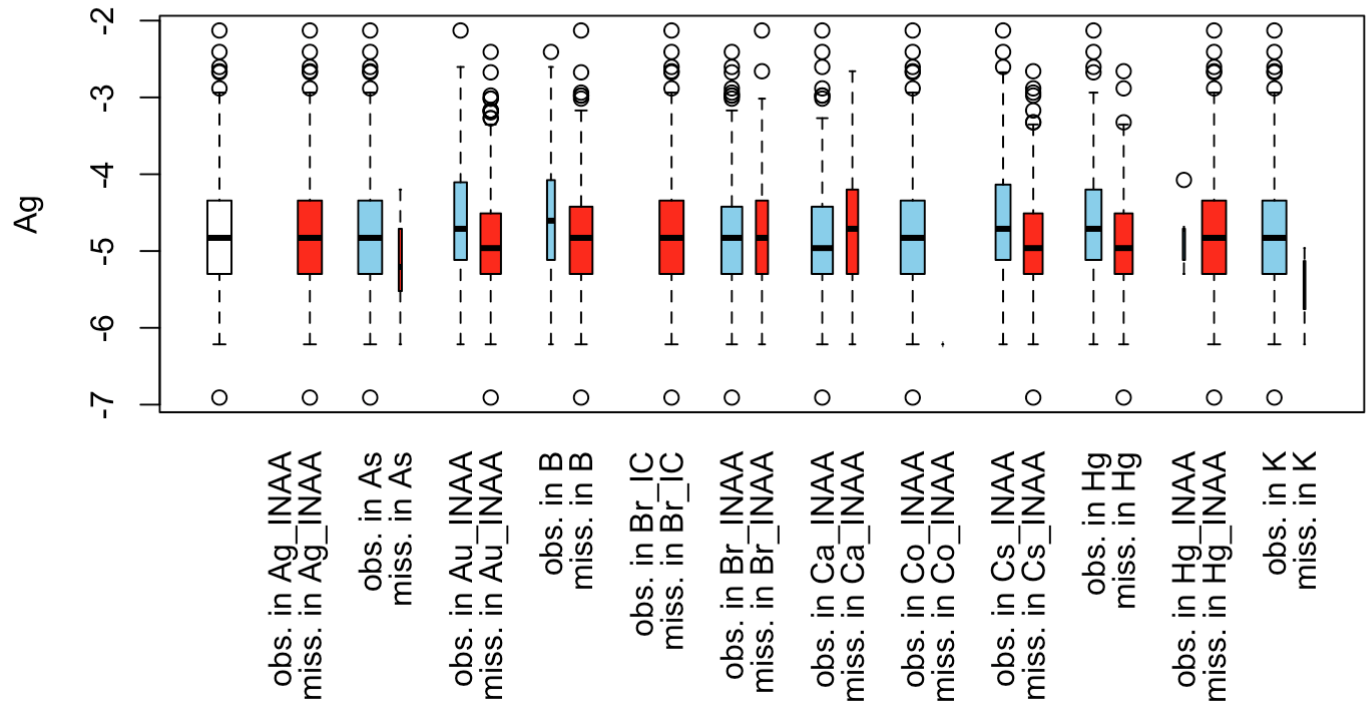
Multivariate Plots

```
parcoordMiss(x4,plotvars=2:11, interactive = FALSE)
```



Multiple Plots

```
pbox(x5)
```



Donor Imputation - hotdeck

```
hotdeck(data, variable = NULL, ord_var = NULL,
  domain_var = NULL, makeNA = NULL, NAcond = NULL,
  impNA = TRUE, donorcond = NULL, imp_var = TRUE,
  imp_suffix = "imp")
```

- *data* - data.frame
- *variable* - variables to be imputed
- *ord_var* - variables to sort by
- *domain_var* - variables to build imputation classes
- a random sort variable is always be added

Donor Imputation - kNN

```
kNN(data, variable = colnames(data), metric = NULL,
  k = 5, dist_var = colnames(data), weights = NULL,
  numFun = median , catFun = maxCat , makeNA = NULL,
  NAcond = NULL, impNA = TRUE, donorcond = NULL,
  mixed = vector(), mixed.constant = NULL, trace = FALSE ,
  imp_var = TRUE, imp_suffix = "imp", addRandom = FALSE)
```

- *dist_var* - variables used for distance combination
- *weights* - weights for distance computation
- *numFun*, *catFun* - aggregation function for numerical or categorical target variables (*sampleCat*, *maxCat*).
- *addRandom* - add a random variable to the distance computation (very low weight)

Donor Imputation - matchImpute

Random within groups imputation, grouping variables are dropped sequentially in case all values are missing in a group.

```
matchImpute(data,
  variable = colnames(data)[!colnames(data) %in% match_var],
  match_var, imp_var = TRUE, imp_suffix = "imp")
```

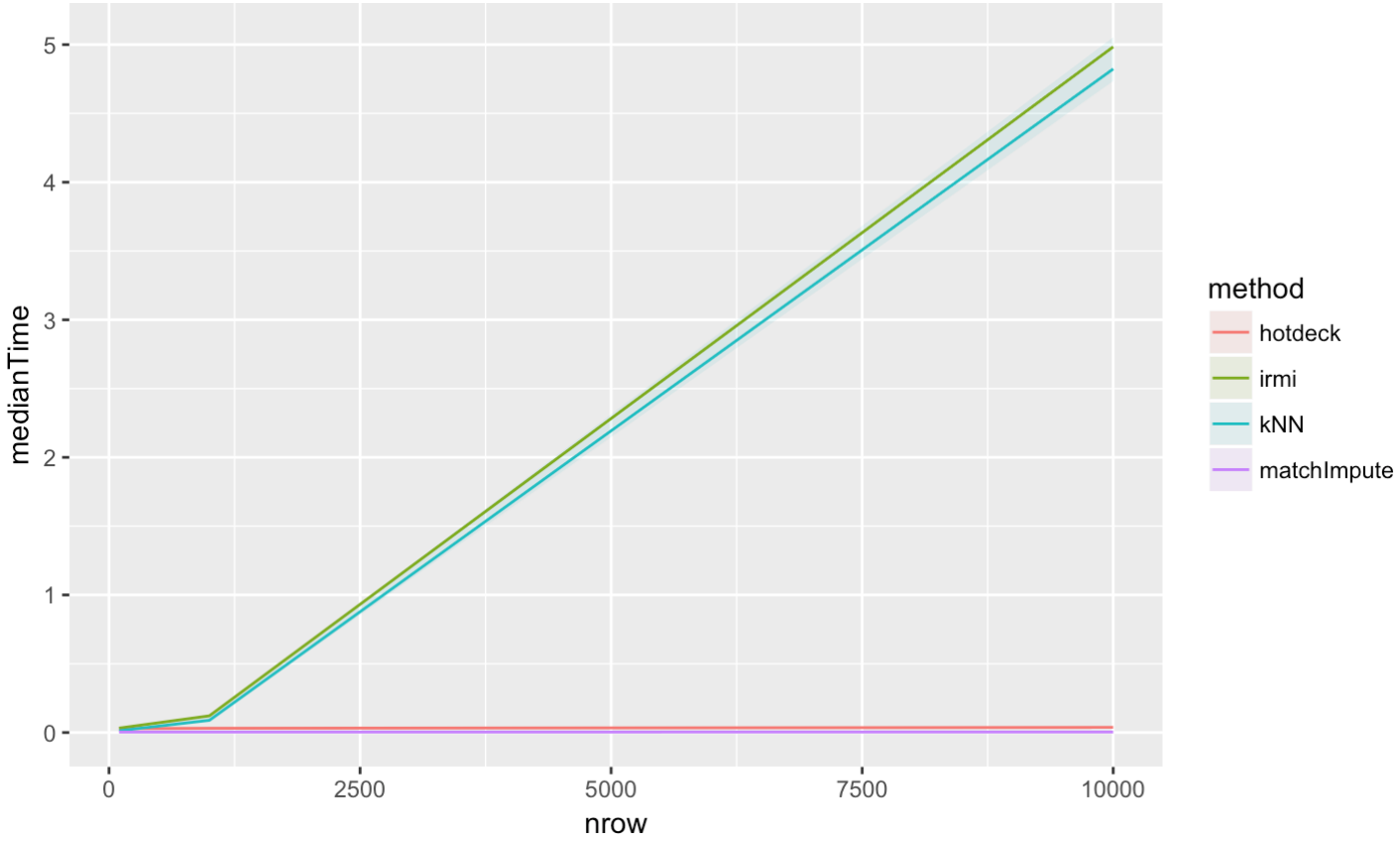
- *match_var* variables to build groups

Iterative (Robust) Regression Imputation (1)

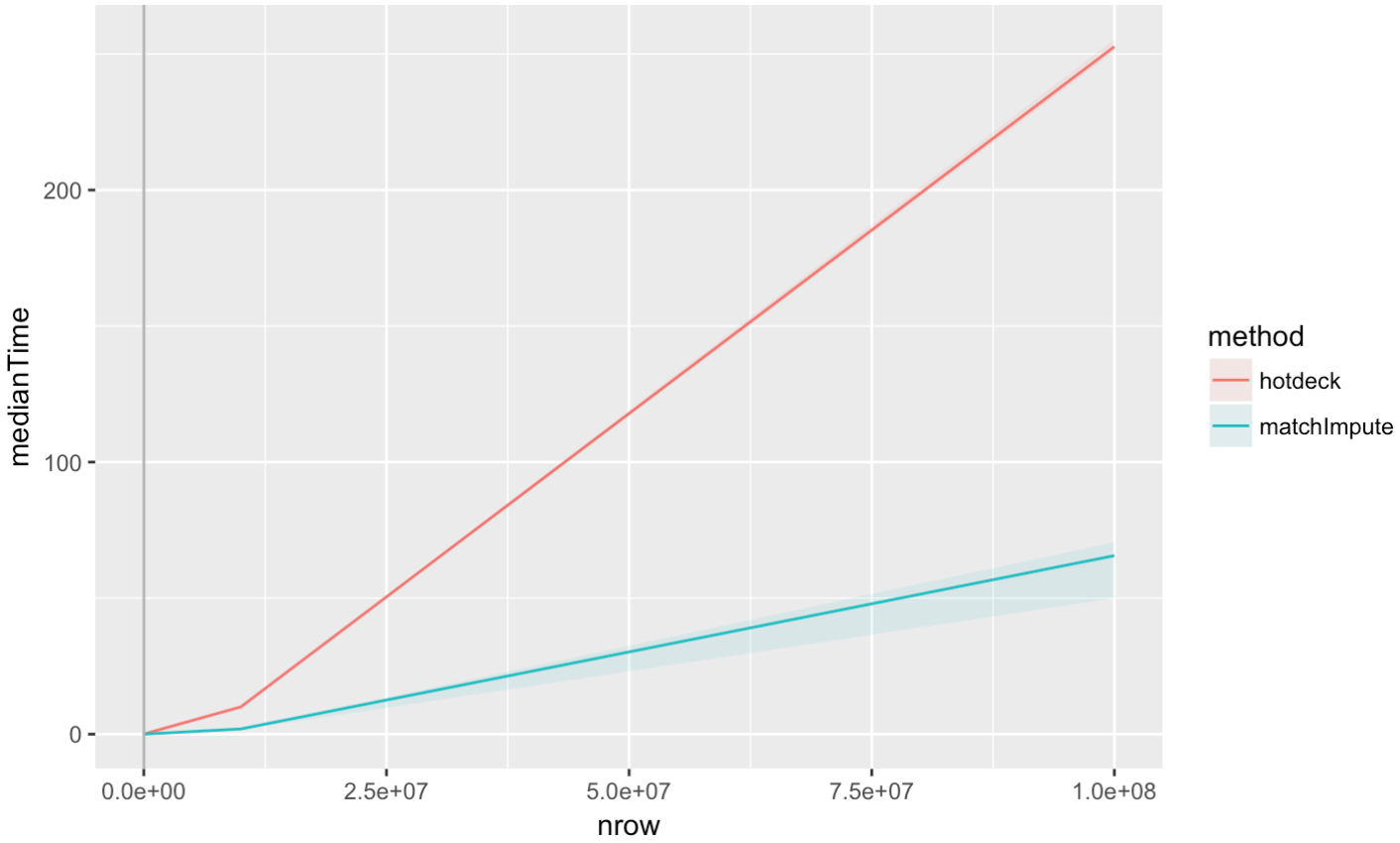
```
irmi(x, eps = 5, maxit = 100, mixed = NULL,
  mixed.constant = NULL, count = NULL, step = FALSE ,
  robust = FALSE , takeAll = TRUE, noise = TRUE,
  noise.factor = 1, force = FALSE , robMethod = "MM",
  force.mixed = TRUE, mi = 1, addMixedFactors = FALSE ,
  trace = FALSE , init.method = "kNN")
```

- *robust* - robust or non-robust
- *step* - *stepAIC* in every iteration
- *mixed* - column indices of semi-continuous variables
- *count* - column indices of count variables (Poisson)
- *noise* - add a random error to the imputed value
- *mi* - number of imputations ⇒ multiple imputation

Imputation Benchmarking (1)



Imputation Benchmarking (2)



Iterative Robust Regression Imputation (2)