WP.XX
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Paris, France, 28-30 April 2014)

Topic (v): International Collaboration and Software & Tools

NEW FEATURES OF VIM - VISUALIZATION AND IMPUTATION OF MISSING VALUES

Submitted by STATISTICS AUSTRIA& TU WIEN [1]

I. INTRODUCTION

The first version of the The package `VIM` [Templ et al., 2013, 2012] have been developed to explore and analyze the structure of missing values in data using graphical methods, to impute these missing values with the built-in imputation methods and to verify the imputation process using visualization tools, as well as to produce high-quality graphics for publications.
A graphical user interface have been newly developed to give access to these methods and tools to users with limited R skills. It is available in the package `VIMGUI` [Schopfhauser et al., 2013]. All important methods are supported by the flexible point- and click-interface.
This paper describes the application of the methods available in the package `VIM` and demonstrates the usage of the graphical user interface in `VIMGUI`. Special attention is also given to the new imputation functionality and the VIM integration of survey objects that are output from R's survey package [Lumley, 2012].

II. THE R PACKAGE VIM

A. OVERVIEW

The package VIM includes visualization techniques to explore the structure of incomplete and imputed data. Therefore it is not just possible to analyze the structure and relations of missing and non-missing data parts, but also to analyze imputed data. The visualization techniques are described for missing values are described in detail in [Templ et al., 2012].
In addition, in the package `VIM` various kind of imputation methods are included. The most common imputation techniques are described in Section B. The range of available methods is quite extensive from old-fashioned methods like hot deck imputation to quite sophisticated methods like iterative step-wise robust regression imputation [Templ et al., 2011].

[1] Prepared by Alexander Kowarik (alexander.kowarik@statistik.gv.at), Matthias Templ (matthias.templ@gmail.com), Daniel Schopfhauser (e0925704@student.tuwien.ac.at).

B. IMPUTATION METHODS

Three kinds of imputation methods are currently implemented in `VIM`, namely hot-deck, k-nearest neighbor and iterative robust model-based imputation. The `VIMGUI` supports also individual regression imputation where users can specify (the formula of) a model for regression imputation by point and click. All of the above methods are implemented in a flexible manner with many options for customization.

1. **Hot-deck Imputation:** The implementation of the popular sequential and random hot-deck algorithm with the option to use it within a domain. The most important arguments are:

- `data` - a data frame or matrix, which contains the data set with missing values in some variables
- `variable` - a vector of variable names for which missing values should be imputed
- `ord_var` - a vector of variable names for sorting
- `domain_var` - a vector of variable names for building domains and to impute within these

The full call of the function is (sensible defaults are given for the available function parameters):

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

From the point of computational speed, this method is faster than any other methods but the quality of imputations might be increased with the following methods.

2. k **Nearest Neighbor Imputation:** The implementation of the k-NN algorithm is base on an extension of the Gower distance, which can now handle distance variables of the type binary, categorical, ordered, continuous and semi-continuous. The most important arguments are:

- `data` and `variable` - see above
- `dist_var` - a vector of variable names to be used for calculating the distances
- `weights` - a numeric vector containing a weight for each distance variable
- `numFun` - a function for aggregating the k nearest neighbors in the case of a numerical variable, defaults to the median.
- `catFun` - a function for aggregating the k nearest neighbors in the case of a categorical variable. The function `maxCat` (which is the default) chooses the level with the most occurrences and random if the maximum is not unique. The function `sampleCat` samples with probabilities corresponding to the occurrence of the level in the nearest neighbors.
- `addRandom` - a boolean variable if a additional variable containing only random numbers should be added to avoid multiple selection of the same donor.

The full call (including sensible defaults) of the function is:

```
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)
```

The method is implemented in a sophisticated manner so that not the whole distance matrix must be calculated. Thus the implementation in `VIM` is also applicable for large data sets.

3. **Iterative Robust Model-Based Imputation:** This iterative regression imputation method is described in detail in [Templ et al., 2011]. In each step of the iteration (inner loop), one variable is used as a response variable and the remaining variables serve as the regressors. The procedure is

repeated until the algorithm converges (outer loop). The data can consist of a mix of binary, categorical, count, continuous and semi-continuous variables, appropriate regression methods are selected internally by the algorithm. Robust regression using MM-estimation [Maronna et al., 2006] is used by default to get reliable results even if the data contains outliers. The most important arguments are:

- `robust` - a boolean variable to enable or disable robust regression
- `step` - a boolean variable to enable or disable a step-wise (`stepAIC`) selection of regressors in each iteration.
- `mixed` - column index of the semi-continuous variables
- `count` - column index of the count variables

The later two arguments are important for selecting the correct regression method, which is done within the algorithm in an automatized manner hidden from the user. If the data contains semi-continuous or count variables, this must be specified while the algorithm detects the correct distribution for all other types (continuous, binary, categorical).
The full call of the function is:

```
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")
```

This method is most useful to get reliable imputations in an automated manner. However, if the users are trained in regression modelling, the following method can be useful.

4. **Individual Regression Imputation:** Through the `VIMGUI` a formula can be specified for defining a model that describes a single variable using a combination of explanatory variables for regression imputation. The `VIMGUI` guides the user by point and click to formulated a certain model.

C. VISUALIZATION OF MISSING VALES AND IMPUTED VALUES

Simple aggregation plots but also specialized univariate, bivariate, multiple and mulitvariate plots are available and ready to use in `VIM` (and `VIMGUI`).

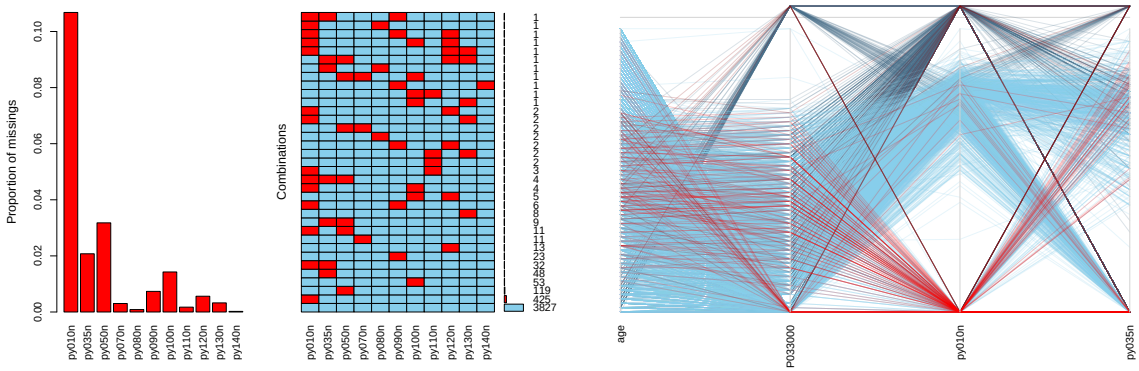

FIGURE 1. Two examples for visualization of missing values. The left plot shows the proportion of missing values, the middle one the combination of variables corresponding to missingness and the right plot shows a parallel coordinate plot of few variables. The EU-SILC data was used for demonstration.

The most simplest two visualization methods are displayed in Figure 1 together with a more advanced method. The left plot simple shows the proportion of missing values in the income components of the EU-SILC data from Austria 2006. The plot in the middle shows the combination of missingness in the rows of the EU-SILC data. 3827 observations are complete, 425 observations have only missings in variable *py010n*, 32 observations include missings in *py10n* and *py035n*, for example. This plot is perfect to see monotone missing values. The right plot should show one of many advanced methods implemented in `VIM`. The parallel coordinate plot displays with blue lines those observations which include observed values in the income components, while the red lines displays observations with missing values. Note that this plot can highly be customized, it is interactive and variables for missing values highlighting can be interactively selected.
We do not show further functionality of the visualization methods in the package but refer to **??**.

III. THE R PACKAGE VIMGUI

The R package `VIMGUI` implements a graphical user interface for the R package `VIM`. The visualization and the imputation functionality is accessible via an easy-to-use point and click user interface.
The following features are supported by the GUI and clickable at the menu of the GUI:

Import/Export: Import of CSV files with interactive selection of parameters for importing including a preview of the data whereas parameters for the data import (e.g., the separator, the symbol for the decimal, ...) can be selected interactively. The import and export of SPSS, SAS (XPORT format), Stata and R binary files is supported. Objects of class `survey`, from the R package `survey` for analyzing data from sample surveys, can be used as input for the imputation process - over the menu entry *Survey* where such survey objects can be imported to *VIMGUI* and exported. Facilities to create a survey object are included as well.

Scaling/Transformation: Scaling and transformation of continuous and semi-continuous variables.

Script: The results produced within the GUI are saved as commands in a separated file. This is useful to provide reproducibility.

Colours/Shading: Colors and alpha-transparency values can be set for the plots.

In the following some impressions of the GUI are shown. We want to show the functionality of the GUI but do not get into details about a specific data set. Thus, a very simple data set is used for demonstration, whereas the interpretation of the data set is not in focus.
In Figure 2 the variable configuration is shown. For some algorithms, like the model-based imputation, it is important to specify the type of distribution of the variables. The second column should guide the users to make the correct classification, it simple shows some values of each variable.

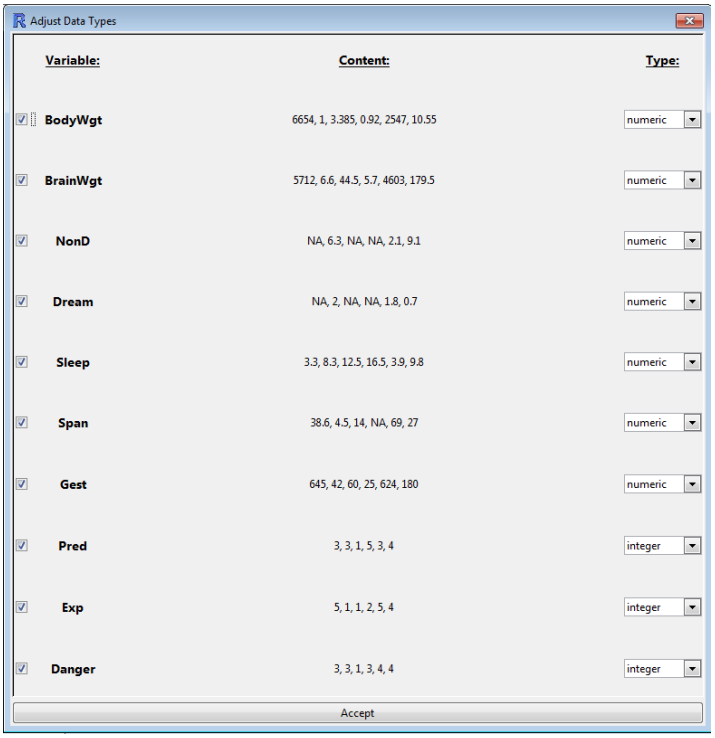

FIGURE 2. Menu for Variable Type Configuration

The next figure, Figure 3, the first tab of `VIMGUI` is shown - the *Data* menu. It simple gives an overview of variables and the most basic (point and variance) estimations can be displayed also by domain. The estimations are done with package survey if a survey object is used.

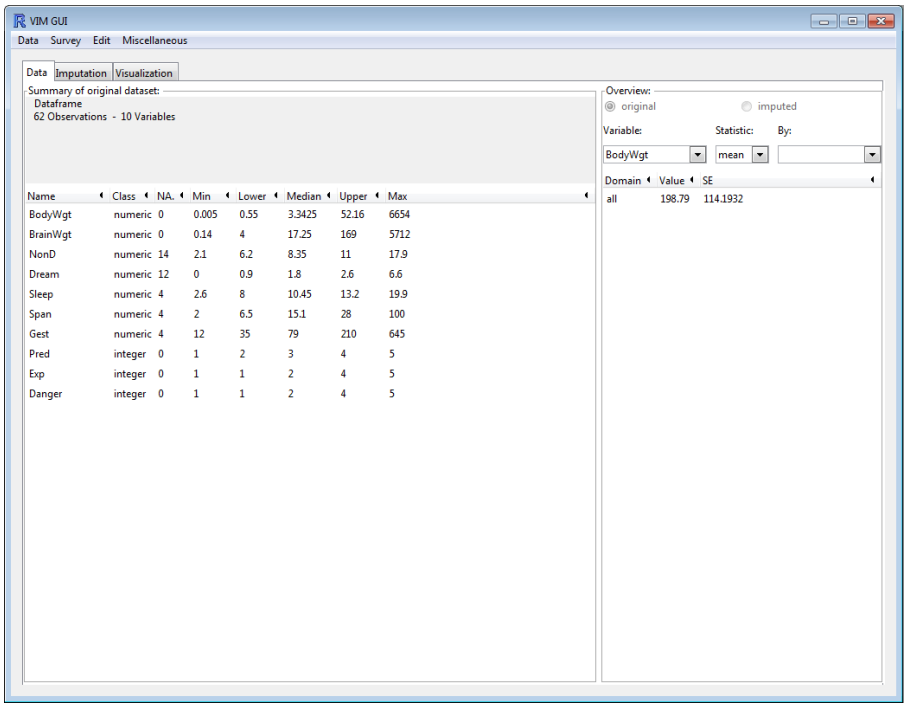

FIGURE 3. The Data Tab

In the imputation tab (Figure 4) the described three different imputation methods plus simple regression imputation, where users can specify their own imputation regression model, can be selected. Depending on the method chosen in the sidebar tab, parameters can be selected.

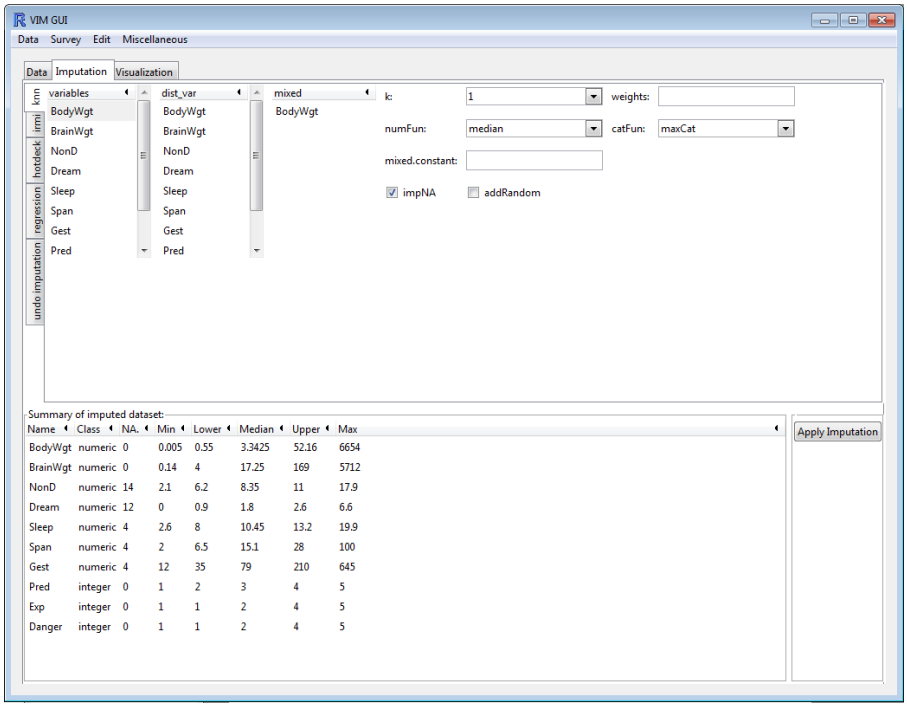

FIGURE 4. The Imputation Tab

If the regression model is chosen as imputation method, a model formula can be created with the GUI.

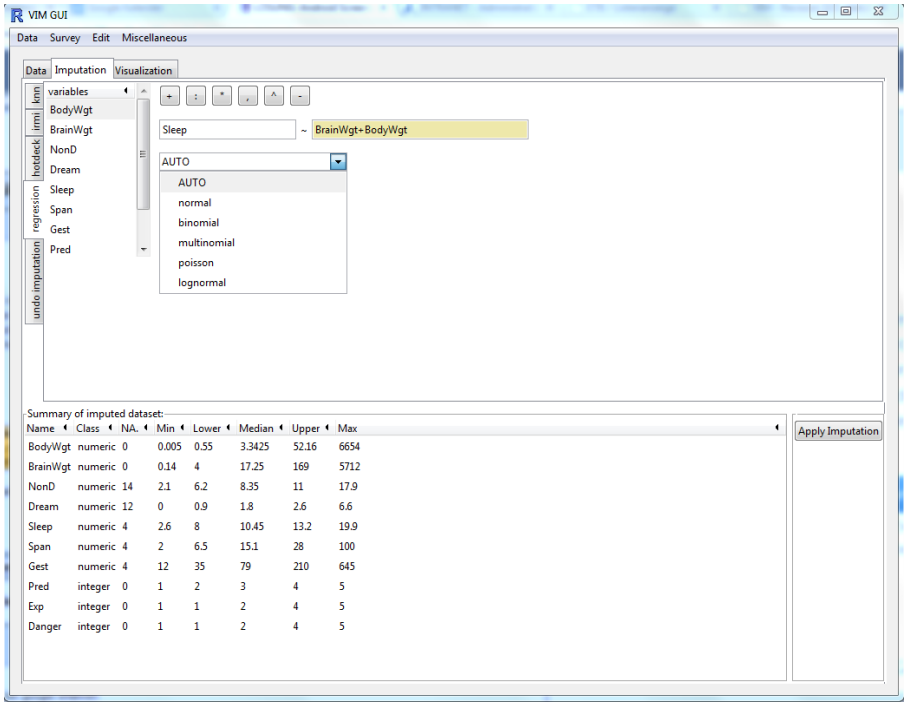

FIGURE 5. The Regression Imputation Method

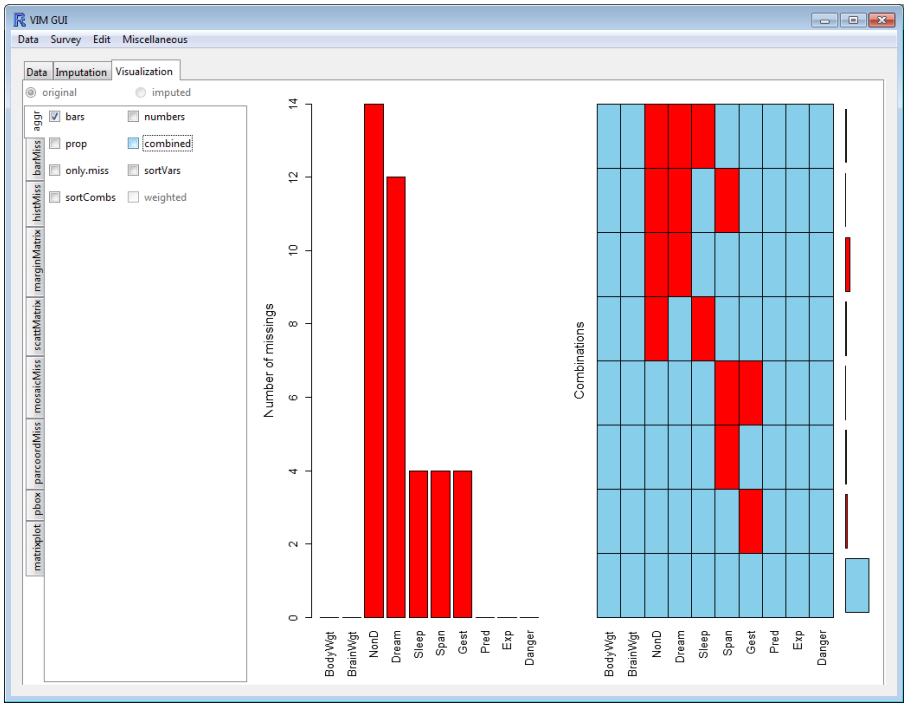

FIGURE 6. The Visualization Tab

In the visualization tab, the visualization methods can be selected in the left vertical tab. If the data are not imputed, the distribution of missing values related to observed values are shown in the plots. If the data are already imputed, the distribution of imputed data is displayed.

IV. CONCLUSION

The *VIM* and *VIMGUI* packages includes a comprehensive collection of imputation and visualization methods. Before imputation, the structure of missing values can be explored using the build-in visualization tools. These tools are easily clickable and parameters for the methods can be changed by point and click.
The imputation methods can either be applied on data frames but also on objects resulting from the survey package. Note that more functionality is included in the package as shown in this contribution, most of these functionality can be easily accessed by point and click.
The *VIM* and *VIMGUI* packages are currently widely used over the world. The download statistics for the R-package downloads over one of many download mirrors – the RStudio server – shows that the package downloaded more than 150 times a week over the last months (package updates also contributes to this number).

References

T. Lumley. *survey: analysis of complex survey samples*, 2012. R package version 3.28-2.

R.A. Maronna, R.D. Martin, and V.J. Yohai. *Robust Statistics: Theory and Methods*. John Wiley & Sons, New York, 2006. ISBN 978-0-470-01092-1.

S. Schopfhauser, M. Templ, A. Alfons, A. Kowarik, and B. Prantner. *VIMGUI: Visualization and Imputation of Missing Values*, 2013. URL http://CRAN.R-project.org/package=VIMGUI. R package version 0.9.0.

M. Templ, A. Kowarik, and P. Filzmoser. Iterative stepwise regression imputation using standard and robust methods. *Comput Stat Data Anal*, 55(10):2793–2806, 2011.

M. Templ, A. Alfons, A. Kowarik, and B. Prantner. *VIM: Visualization and Imputation of Missing Values*, 2013. URL http://CRAN.R-project.org/package=VIM. R package version 4.0.0.

Matthias Templ, Andreas Alfons, and Peter Filzmoser. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*, 6:29–47, 2012.