Visualization of imputed values using the R-package VIM

Bernd Prantner

October, 2011

The package **VIM** (visualization and imputation of missing values) [Templ et al., 2011a] is developed to explore and analyze the structure of missing or imputed values in data using graphical methods. Getting knowledge about the structure is helpful to identify the mechanism, which is generating the missings, respectively errors, which may have happened in the imputation process. Furthermore, it is also necessary for selecting an appropriate imputation method in order to reliably estimate the missing values. Package **VIM** offers different built-in imputation methods to impute the missing values. Moreover, it can also be used to produce high-quality graphics for publications. A simple graphical user interface allows an easy handling of the included methods.

This paper describes the appliance of the package **VIM** and the included methods to produce graphics for imputed data. It can be seen as addition to Templ et al. [2012] and the vignette of the package **VIM** [Templ and Alfons, 2009], which are focused on the visualization of missing values. At first, the graphical user interface and the differences of the built-in imputation methods are explained. Also the concept, how the package **VIM** distinguishes imputed and normal values, is illustrated. Afterwards, a detailed description of each graphical method and example plots are given. Additionally, important customizable parameters are noted and explained. In order to reproduce all plots, only the example data sets from package **VIM** are used and the R-commands are supplied.

Contents

List of Figures

1 Description of the graphical user interface of VIM

After the package **VIM** is loaded, it can be used in two different ways, by directly using the R console, or by taking advantage of the developed graphical user interface (GUI).

```
> library(VIM)
```

The R package **tcltk** [R Development Core Team, 2009] has been used to develop the GUI. A GUI is helpful for inexperienced users and for users who aren't familiar with the package **VIM**. It allows easy handling of the included functions. The GUI can be loaded with the following command:

```
> vmGUImenu()
```

If the GUI has been closed and is reopened during a session, all selections and settings will be recovered. [Templ and Alfons, 2009] Figure 1 shows the initial GUI.

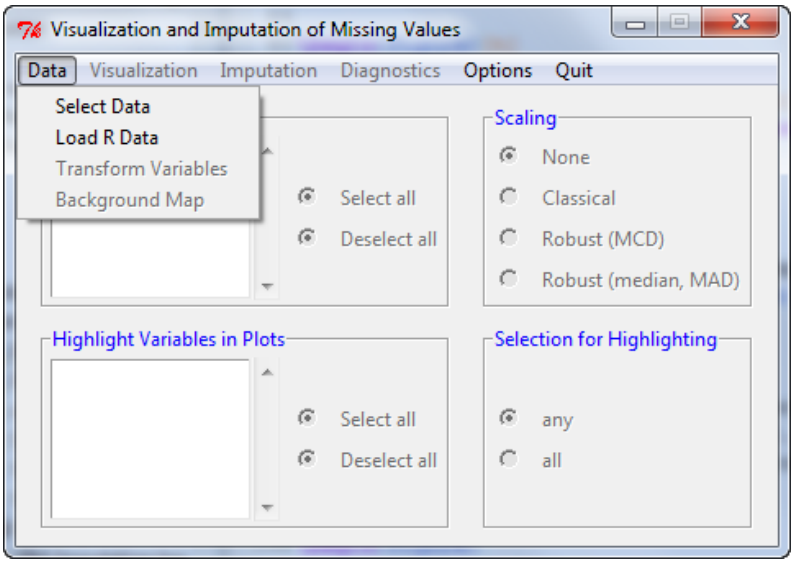


Figure 1: The **VIM** GUI and it's menu for importing data

Since this paper is focused on the imputation process, the important menus are the *Data*, the *Imputation*, the *Diagnostics* and the *Options* menu.

1.1 Data menu and data handling

In the *Data* menu, one can select a data frame from the R workspace (see Figure 2). Three data sets are provided by the package **VIM**, *chorizonDL* [Reimann et al., 2008], *sleep* [Allison and Cicchetti, 1976] and *tao* (derived from the GGOBI project [Swayne et al., 2003]).

Alternatively, a data set in `.RData` format can be loaded from the file system into the R workspace and into the GUI.

1.1.1 Selecting variables

After a data set has been chosen, or loaded, it's variables are shown in the *Select Variables* and *Highlight Variables in Plots* dialog (see Figure 3). With the distinction of plot and highlight variables, one can plot a certain variable and highlight values which are missing or imputed values in other variables.

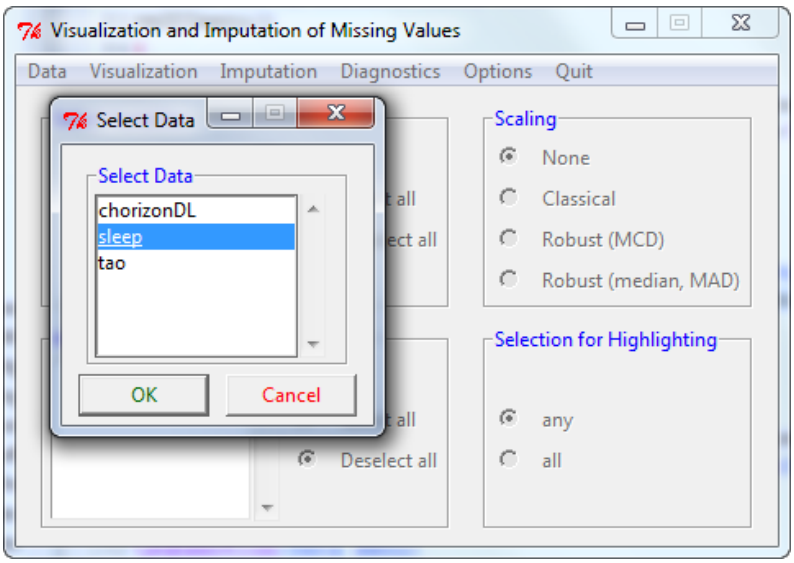


Figure 2: The dialog for data selection.

If more than one variable is chosen for highlighting, it is possible to select whether observations with missing, respectively imputed values in any or in all of these variables should be highlighted in the *Selection for Highlighting* dialog (see the lower right frame in Figure 3).

An important feature is that the variables will be used in the same order as they were selected, which is especially useful for parallel coordinate plots. [Templ and Alfons, 2009]

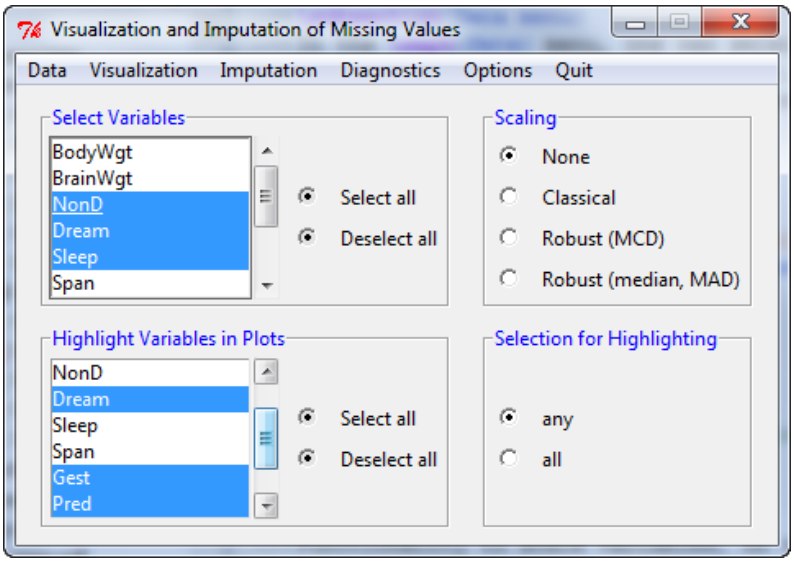


Figure 3: Variable selection with the **VIM** GUI.

1.2 Imputation menu

In the *Imputation* menu one can choose between the built-in methods for the imputation of missing values (see Figure 4). Currently, the implemented methods are:

- k-Nearest Neighbor imputation
- Hotdeck imputation

- IRMI (iterative robust model-based imputation), see [Templ et al., 2011b]

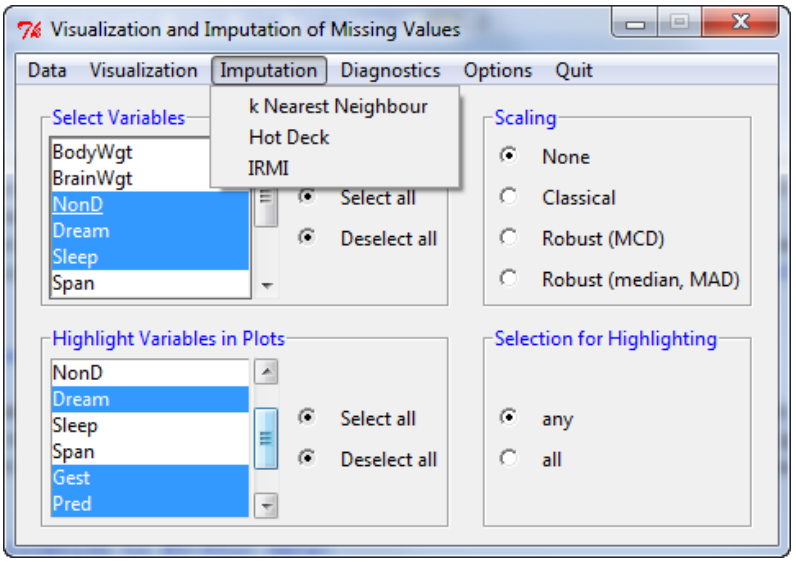


Figure 4: Built-in imputation methods in the **VIM** GUI.

However, the built-in methods are especially adapted for the package **VIM**. Unlike different imputation methods, which simply impute the variables of choice, an additional variable for each imputed variable is added to the data set. Each created variable is related to an imputed variable and represents a logical vector indicating which values of the variable have been imputed. To distinguish all imputed variables from the index variables, and to connect a particular index variable to an imputed variable at the same time, the name of the index variable is composed of the name of the imputed variable and a given suffix defined by the *delimiter*-Parameter in the *Options menu* (see section 1.4 on page 8).

For example, let the name of the variable that should be imputed be **net_income** and the delimiter is set to **_imp** , the resulting index variable would be **net_income_imp**.

Please be aware that the delimiter must be unique in the names of the variables in the data set. The indices are needed for the graphical methods to determine the imputed values that should be highlighted.

NOTE: *The indices variables are NOT shown in the GUI, since this is no gain in information for the users. Instead, they are automatically added to the data set for the graphical methods that belongs to the* Diagnostics *menu.*

Be sure that these variables are included in the data set, which is passed over to the graphical methods for imputed values, as well as the delimiter of these variables, when using the package **VIM** *with the R console.*

1.3 Diagnostics menu

1.3.1 Difference between visualization and diagnostics menu

The GUI comes with two menus for the graphical methods of the package **VIM**, the *Visualization* and the *Diagnostics* menu. The first one is intended to analyze the structure of missing values with graphical methods *before* the imputation process, whereas the *Diagnostics* menu is designed to analyze the outcome *after* the imputation process. Both menus have an identical structure of their commands, but the graphical methods behave differently.

To understand the difference, the internal processing of the commands is explained:

Every command for graphical methods has the same basic functionality. As long as the delimiter is set to `NULL` *(which is the default value)*, the graphical methods of the *Visualization* menu are applied, which means methods to analyze the structure of missing values are executed.

Otherwise, the given data set is searched for variable names matching the given delimiter. They are considered to be index variables for imputed variables. If such variables are found, the graphical methods of the *Diagnostics* menu are applied.

For the users' convenience and to help users unfamiliar with the package **VIM**, if a delimiter is given, but no matching variable names are found, the data set is considered to may have missing values instead and hence the graphical methods of the *Visualization* menu are applied and a warning is printed in the R console.

Only if any graphical method of the *Diagnostics* menu is selected, the index variables, as well as the delimiter, to distinguish between variables with and without imputed values, is added to the data set, which is then passed over to the chosen graphical method.

Consequently, if any graphical method of the *Visualization* menu is selected for already imputed data, no values are highlighted, since the data set is searched for non-existent missing values. However, if a method is selected via the *Diagnostics* menu, but no indices variables are found with the given delimiter, the program automatically executes the suiting graphical method of the *Visualization* menu.

1.3.2 Selecting of the plot methods

Another feature of the **VIM** GUI is, that only applicable plot methods are selectable, depending on the selected variables in the *Select Variables* dialog. That means, if only one variable is selected, only the univariate plot methods are selectable (see Figure 5). Likewise if two variables are chosen in the dialog, only the bivariate graphical methods are applicable.

The graphical Method "Aggregate Missings and imputed Missings" is excluded from this restraint, since it's applicable to all dimension of data, it's always selectable.

1.4 Options menu

Figure 6 shows the *Options* menu, which is used for the adjustment of the global settings of the package **VIM**.

- In the top frame one can alter the colors used in the graphical methods.
- The frame beneath that is intended to set the transparency of the colors. This can be used to prevent overplotting.
- The third frame is an option to embed multivariate plots in `Tcl/Tk` windows.
- Last, but not least, the fourth frame contains the delimiter. This is a suffix, to identify the indexes for imputed variables. It is also used by the imputation methods, which append this suffix to the name of the created indices during the process.

2 Plot methods

In the following, example data sets from package **VIM** are used to explain the diagnostics methods. Since the data comes with the package, the reader can reproduce all plots in this paper easily.

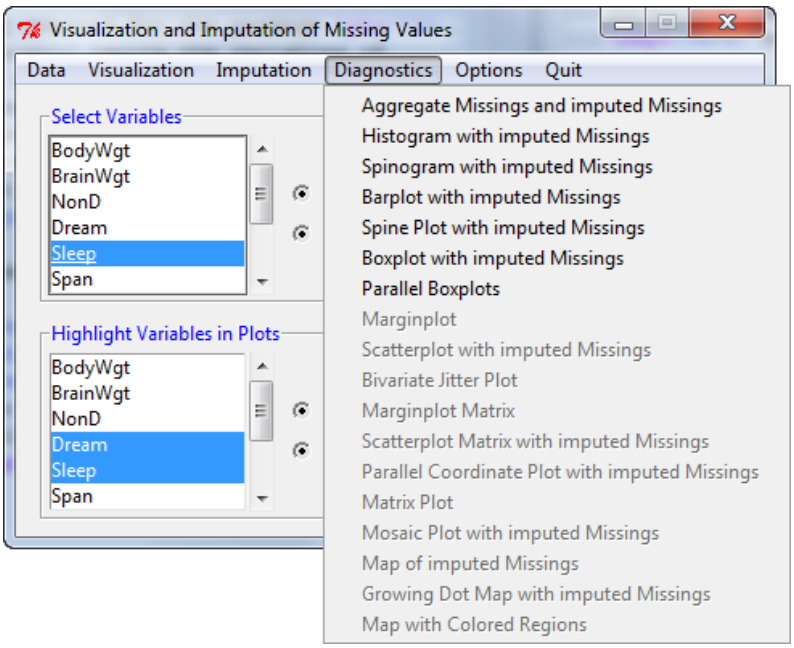


Figure 5: Applicable univariate graphical methods in the diagnostics menu, based on the selection of one variable

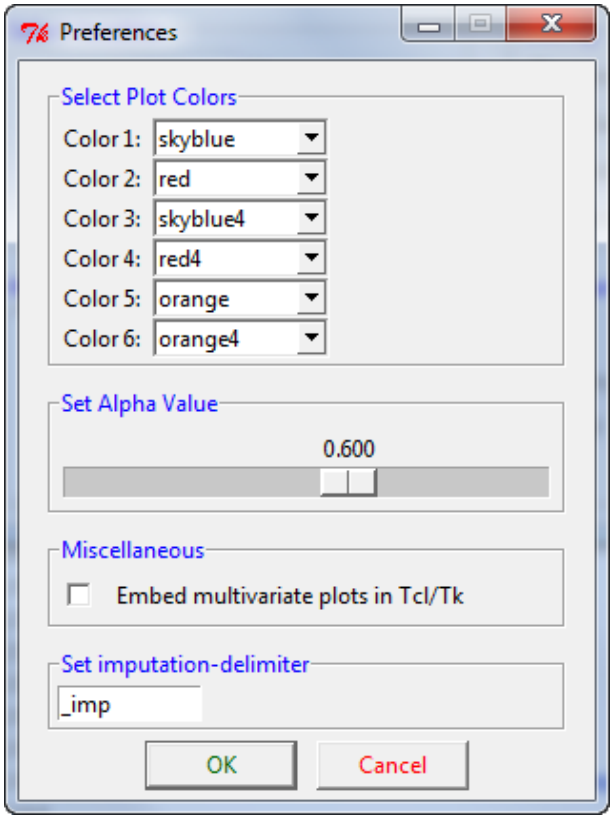


Figure 6: The Options menu

Additionally, the mean imputation method of the R package **e1071** [Dimitriadou et al., 2011] is used. This method isn't implemented in the package **VIM**, but it serves for comparison with the implemented methods in this paper. It uses the (column-wise) average value of the observed values for the imputation of missing

values. This means, the missing values of the first column are estimated with the mean of the observed values of the first variable. Although it's a very fast and popular imputation method, it's not recommended to use it in practice, because it doesn't preserve the relationship among variables.

The default color-scheme for all graphical methods is **blue** for observed values, **red** for missing values and **orange** for imputed values.

If not mentioned otherwise, the used data set for the plot methods are already imputed completely. Subsequently, the R-commands for the used imputation methods are given:

```
> sleep_kNN  <- kNN( sleep, k = 5)
> tao_kNN <- kNN(tao, k = 5)
> chorizon_kNN  <- kNN( chorizonDL ,variable=c("Ca","As","Bi"), dist_var=c("Hg","K","
```

3 Aggregate missings and imputed missings

First of all, it may be of help to get an overview of the data set, e.g. of the number of missing values or the number of missing values which have been imputed, may be of interest. It may be even more interesting to analyze if there are certain combinations of variables with missing or imputed values.

This can be reviewed easily by selecting the variables of interest and by clicking on `Diagnostics` → `Aggregate Missings and imputed missings`. Figure 7 shows the aggregation plot of the data set *sleep*, in which the variables *Sleep, Dream and NonD* have been imputed.

If the command line of R is preferred, the same plot can be created with following commands:

```
> sleep_kNN_part <- kNN(sleep, variable=c("Sleep","Dream","NonD"), dist_var=colnames(
> aggr(sleep_kNN_part, delimiter="_imp", numbers=TRUE, prop=c(TRUE,FALSE))
```

As indicated before, variables containing missings and variables with imputed values can be combined in the aggregation graphic.

In this particular plot, one can immediately see the three imputed variables, which are colored **orange**. The variables *Span* and *Gest* contain missing values, but haven't been imputed yet, thereby they are colored in **red**.

Subsequently the different plot regions of Figure 7 are explained:

Left plot region
A barplot with the proportion of missing or imputed values in each variable.

This example graphic shows, that the variables *NonD* and *Dream* are having the highest amount of imputed or missing values, while the amount is rather small in the other three variables.

Right plot region
An aggregation plot, showing all existing combinations of missing (red), imputed (orange) and observed (blue) values. Additionally, the frequencies of different combinations are visualized by a small barplot and by the number of their occurrence on the right side.

For example, this plot reveals, that if values in the variable *NonD* are missing, they

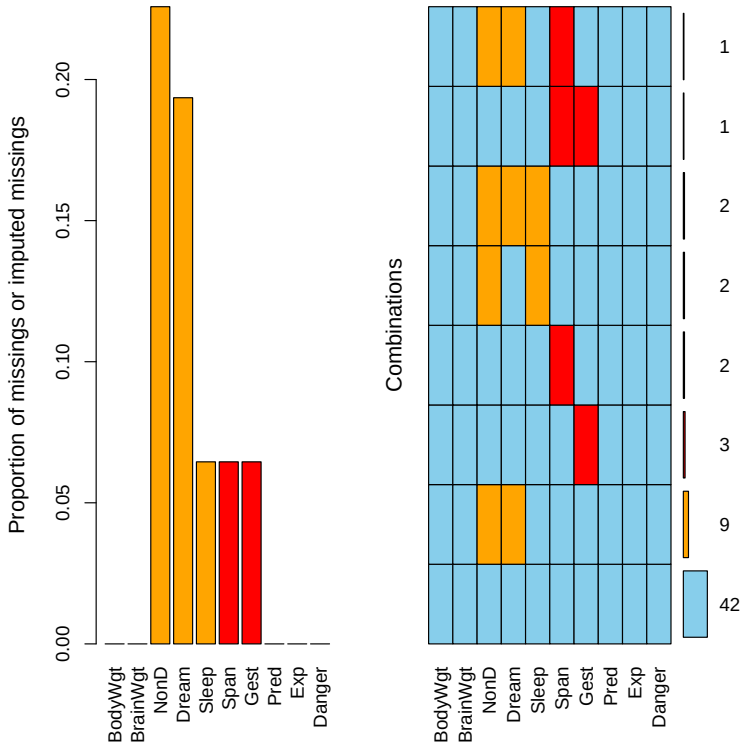


Figure 7: Aggregation graphic of the partially imputed data set *sleep*

are mostly also missing in the variable *Dream*, except for two times. It also shows, that the variable *Sleep* only has missing, which are also missing in the variable *NonD*.

Obviously, this plot method is identical for every imputation method, since it only displays which values of the variable are missing or imputed values.

3.1 Customizing the graphic

When using the command line of R, the graphic can be customized. Subsequently, an example code of customized aggregation plots is given and some of the adjustable parameters are explained:

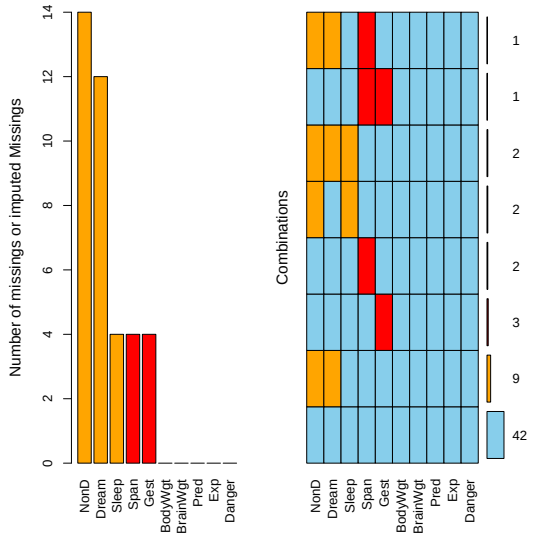
```
> aggr(sleep_kNN_part, delimiter="_imp", sortVars=TRUE, numbers=TRUE, prop=c(FALSE, T
> aggr(sleep_kNN_part, delimiter="_imp", sortVars=TRUE, numbers=TRUE, prop=c(FALSE, T
> aggr(sleep_kNN_part, delimiter="_imp", combined=TRUE, sortVars=FALSE, sortCombs=TRU
```

sortVars
When *sortVars* is set to `TRUE` the variables in the left barplot are sorted, according to their number of missing, respectively imputed values.

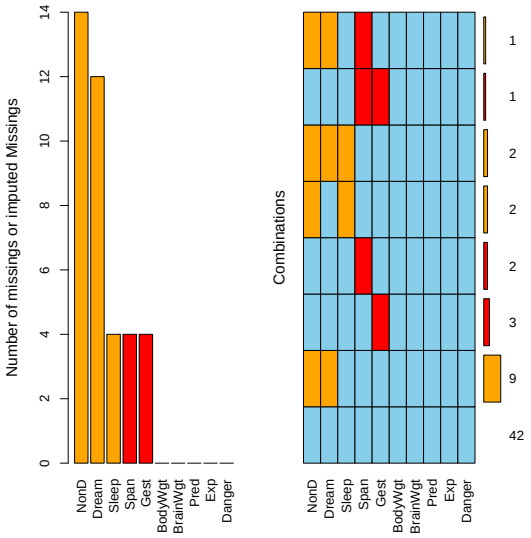
sortCombs
sortCombs is the equivalent of *sortVars* for the aggregation plot on the right side of the plot. If set, the combinations are sorted by the frequency of their occurrence. By default, it's already set to `TRUE`.

Figure 8a shows an aggregation plot with both parameters set to `TRUE`.

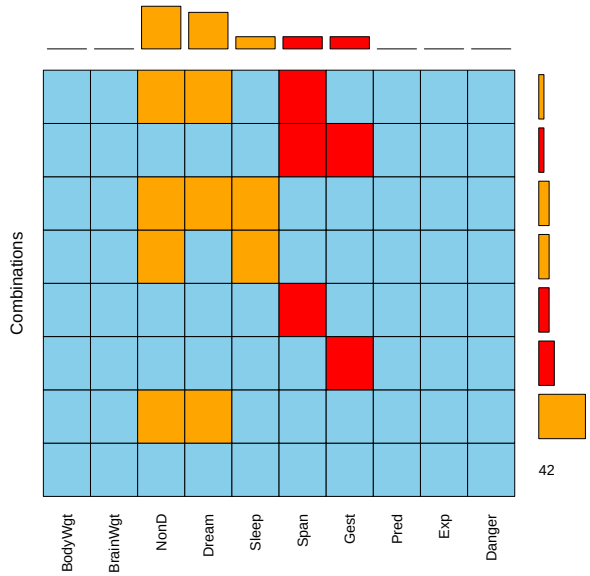
only.miss
If set to `TRUE`, the barplot on the right side of the plot region is only drawn for com-



(a) Aggregation graphic with sorted variables and combinations



(b) Aggregation graphic showing only combinations with missing or imputed values in the right barplot



(c) Combined aggregation graphic

Figure 8: Customized aggregation plots of the partially imputed data set *sleep*

binations including missing or imputed values. This is helpful if most observation are complete and therefore the corresponding bar would dominate the barplot and the remaining bars would be too compressed.

Figure 8b is a plot with this parameter set to `TRUE`. One can clearly see the difference to the barplot in Figure 8a.

combined

In figure 8c the two plot regions of the aggregation graphic are combined. The barplot with the proportion of missing or imputed values in each variable, which was on the left side of the plot, is now displayed on top of the graphic.

4 Histogram and barplot with imputed missings

Both methods are adaptations of the familiar graphics. If more than one variable is supplied, the bins for the variable of interest will be split according to the imputed values in the additional variables. Imputed values in the variable of interest are visualized with a barplot on the right side of the plot, which is separated by a small gap. This bar is also split if more than one variable is supplied.

A histogram is produced if the variable of interest is of type **numerical**. Otherwise, if it's of type **categorial**, a barplot is drawn.

Note that the type of the variable of interest is automatically checked and a suitable graphical method is then executed by the program. However, for user's convenience, the GUI still specifies different options for both methods. If the wrong method for the variable of interest is selected, the appropriate one is executed anyhow.

Both graphics are intended to visually analyze the distribution and the proportion of observed and imputed values of the variable of interest, to discover outliers and to detect possible bi- or multivariate dependencies.

They can be produced by first selecting the variable of interest in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Histogram with imputed missings` or `Diagnostics` → `Barplot with imputed missings` respectively.

Figure 9a shows a histogram of the variable *Air.Temp* of the *tao* data set. The variable *Humidity* is used for the highlighting of imputed values. Figure 9b instead demonstrates a barplot of the variable *Exp* of the data set *sleep*. The variables *Dream* and *NonD* are used for the highlighting in this plot.

If the command line of R is preferred, the same plots can be created with following commands:

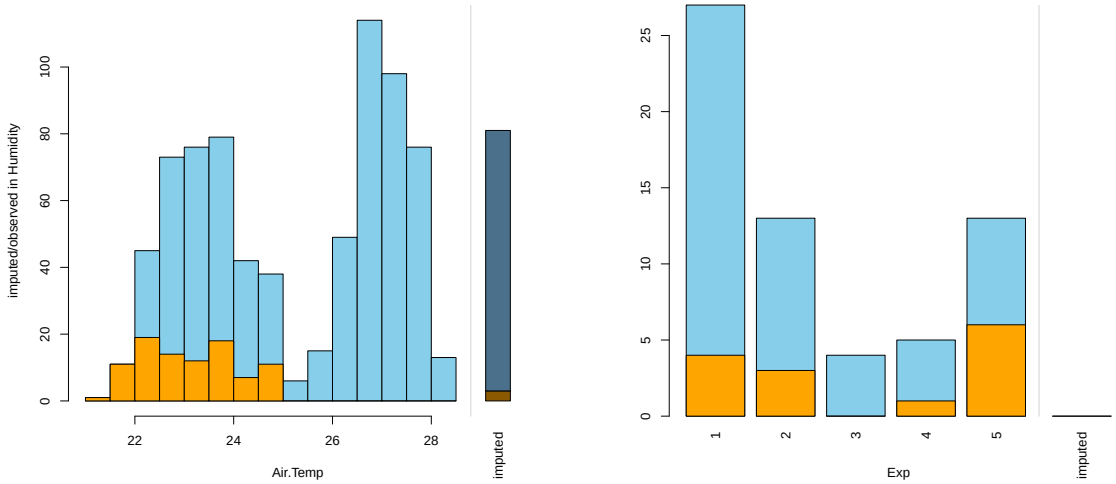
```
> tao_vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> histMiss(tao_kNN[,tao_vars], delimiter="imp", selection="any")
> sleep_vars <- c("Exp","Dream","NonD","Dream_imp","NonD_imp")
> barMiss(sleep_kNN[,sleep_vars], delimiter="_imp", selection="any")
```

Subsequently the different plot regions of Figure 9 are explained:

Histogram or barplot
The histogram or barplot of the variable of interest, depending on the type of this variable. The bars, which are colored **orange**, represent values, which are imputed in the additional variables and the values they're having in the variable of interest.

In graphic Figure 9a, it turns out, that values, which have been imputed in the variable *Humidity* are having low values in the variable *Air.Temp*. This can be a indicator that the distribution of the missing values was not missing completely at random.

By looking at Figure 9b, one can see, that there are no imputed values in any of the other variables, when the category of *Exp* is 3. Otherwise, if it's 5, the proportion of imputed values in the others raises to almost 50%. Note also that category 1 is very dominant in this variable.



(a) Histogram of the variable *Air.Temp* of the data set *tao*, imputed values in *Humidity* are highlighted

(b) Barplot of the variable *Exp* of the data set *sleep*, imputed values in *Dream* or *NonD* are highlighted

Figure 9: Histogram and barplot

Barplot

On the right side of each plot, a barplot is drawn, visualizing the imputed values in the variable of interest. The bar, which is colored **dark blue**, indicates imputed values in the variable of interest. The **dark orange** colored bar, represents values which are imputed values in the variable of interest *and also* in the additional variables.

The right bar plot of Figure 9a in the example shows, that the amount of imputed values in both variables, *Air.Temp* and *Humidity*, is rather small.

The fact, that no barplot is shown in Figure 9b is because there are no imputed values in the variable *Exp* at all.

4.1 Customizing the graphic

Again, when using the command line of R, the graphic can be customized. Since the customizable parameters are similar in both methods, they will be explained using the example of a histogram.

Subsequently, an example code of customized histogram plots is given and some of the adjustable parameters are explained:

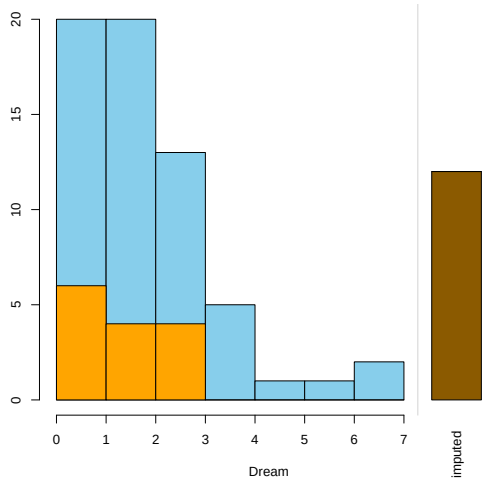
```
> vars <- c("Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> histMiss(sleep_kNN[,vars], delimiter="imp", selection="any")
> histMiss(sleep_kNN[,vars], delimiter="imp", selection="all")
> histMiss(sleep_kNN[,vars], delimiter="imp", selection="all", only.miss=FALSE)
```

selection

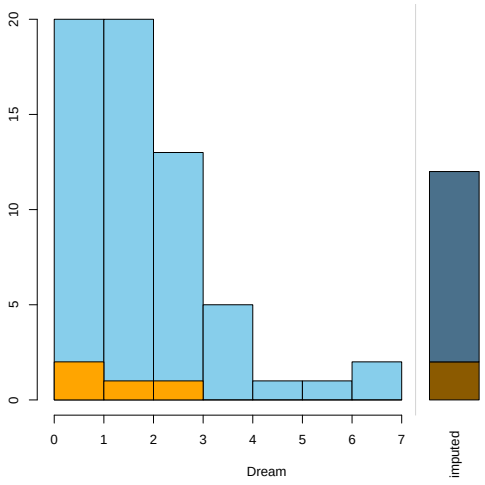
By altering the *selection*, one can choose to highlight values, which are imputed in *any* or in *all* of the additional variables.

Figure 10a and Figure 10b show the different results. The variable *Dream* of the data set *sleep* is set as variable of interest. The variables *NonD* and *Sleep* are used for highlighting.

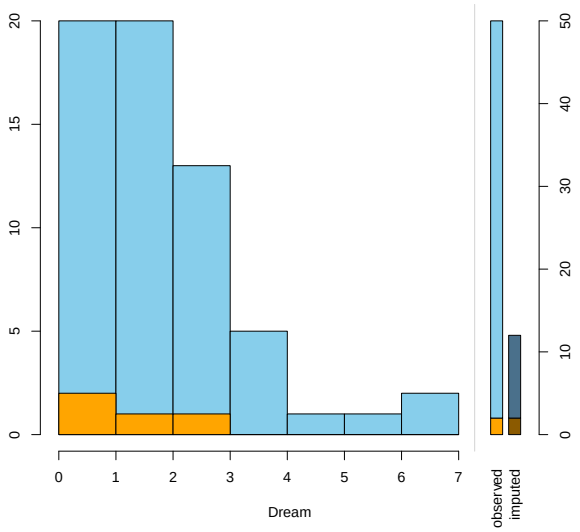
The outcome can be compared to the aggregation graphic of Figure 7. By selecting



(a) Histogram with selection: *any*



(b) Histogram with selection: *all*



(c) Histogram with selection: *all* and only.miss: `FALSE`

Figure 10: Customized histogram graphics of variable *Dream* of the data set *sleep*, imputed values in *NonD* and/or *Sleep* are highlighted

any, all values, which are either imputed in *NonD* or *Sleep* are highlighted. But, by changing it to *all*, only the 4 values, which are imputed in both variables are displayed.

The *tao* data set wasn't chosen for this example, since almost all imputed values are either in the variable *Air.Temp* or in *Humidity* and the third variable containing imputed values, *Sea.Surface.Temp*, only has 3 imputed values. Therefore, changing the *selection* would have marginal effect in the graphic.

only.miss

By setting *only.miss* to `FALSE`, a barplot with two bars, instead of just one, is printed on the right side of the plot. Both are split in two parts again. This barplot is not on the same scale as the main plot anymore, hence an additional y-axis is printed.

The first bar corresponds to observed values in the variables of interest. The bar, which is colored **orange**, marks values, which are observed in the variable of interest, but are imputed values in the additional variables.

The second bar is like the bar in the previous histogram plots, it indicates imputed values in the variable of interest. The **dark orange** bar describes values, which are imputed in the variable of interest and also in the additional variables.

The highlighting of the imputed values in the additional variables is based on the choice of the *selection*.

The Example in Figure 10c is a plot with *only.miss* set to `FALSE`. Since the *selection* is set to *all*, only imputed values in both of the additional variables are highlighted. Once again, the graphic can be compared to the aggregation graphic of Figure 7. The highlighted values of the first bar represent the two values, which are observed values in the variable *Dream*, but are imputed values in both variables, *NonD* and *Sleep*. Whereas the highlighted values of the second bar correspond to the two values which are imputed in all of the three variables.

4.2 Interactive features

This graphic supports interactive features. By clicking in the right margin of the plot region , the graphic is switched to the next variable in the given data set. Likewise, by clicking in the left margin, the graphic is switched to the previous variable. Clicking anywhere else on the graphic device quits the interactive session.

Note that the program automatically checks the type of the new variable of interest and executes the appropriate graphical method. This means, that by switching the variable of interest, the graphic can also switch from a histogram to a barplot and vice versa.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

5 Spinogram and spineplot with imputed missings

Depending on the type of the variable of interest, if it's either of type **numerical** or **categorial**, a spinogram, respectively a spineplot is created.

The spinogram is an alteration to the histogram of the previous chapter (see section 4), where, instead of the vertical axis, the horizontal axis is scaled according to relative frequencies of the categories/classes. The vertical axis is rather scaled to a height of 1.

The spineplot is the same modification performed on the barplot method.

Like before, if more than one variable is supplied, the bins are split according to the values, which are imputed in the additional variables. Hence, the proportion of highlighted observations in each category/class is displayed on the vertical axis. Since the height of each cell corresponds to the proportion of highlighted observations, it is now possible to compare the proportions of imputed values among the different categories/classes.

Significant differences in these proportions indicate a missing at random situation, which should be considered, e.g., when generating close-to-reality scenarios for missing data in simulation studies. [Templ et al., 2012]

Also, imputed values in the variable of interest are visualized with a spineplot on the right side of the plot, which is again separated from the main plot by a small gap and also split if more than one variable is supplied.

Note that the type of the variable of interest is automatically checked again and a suitable graphical method is then executed by the program. For user's convenience, the GUI also specifies different options for both methods and if the wrong method for the variable of interest is selected, the appropriate one is executed anyhow too.

The graphic can be produced by first selecting the variable of interest in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on either `Diagnostics` → `Spinogram with imputed missings` or `Diagnostics` → `Spine Plot with imputed missings` (depending on the type of the variable of interest).

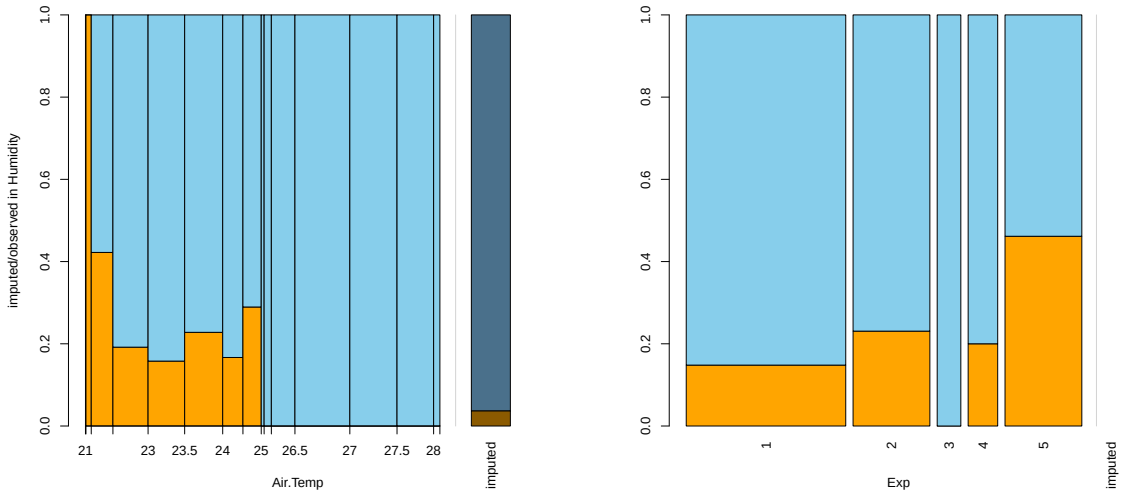
Figure 11 shows an example of the outcome with the two different types of variables of interest.

Figure 11a is a spinogram of the familiar variable *Air.Temp* of the *tao* data set. The variable *Humidity* is again used for highlighting. Since the variable of interest is **numerical**, a spinogram is drawn.

Figure 11b illustrates a spineplot, because the variable of interest, *Exp* of the data set *sleep*, is of **categorial** type. The variables *Dream*, *NonD* and *Sleep* are used for highlighting.

If the command line of R is preferred, the same plots can be created with following commands:

```
> tao_vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> spineMiss(tao_kNN[,tao_vars], delimiter="imp", selection="any")
> sleep_vars <- c("Exp","Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> spineMiss(sleep_kNN[,sleep_vars], delimiter="_imp")
```



(a) Spinogram of the variable *Air.Temp* of the data set *tao*, imputed values in *Humidity* are highlighted

(b) Spinoplot of the variable *Exp* of the data set *sleep*, imputed values in *Dream*, *NonD* or *Sleep* are highlighted

Figure 11: Spinogram and Spinoplot

Subsequently the different plot regions of Figure 11 are explained:

Spinogram or spineplot
The spinogram or spineplot, depending on the type of the variable of interest. Similar to the histogram method in section 4, the bars, which are colored **orange**, represent values, which are imputed in the additional variables and the values they're having in the variable of interest.

Instead of the height, in the spinogram the width of the bins represent the relative frequency of the categories/classes.

Figure 11a is a different representation of the histogram graphic in Figure 9a, thus the results are similar. Though, the proportion of imputed values can now be compared among the different categories/classes.

The same applies to Figure 11b, it's also a modified version of the barplot in Figure 9b. Certainly, the dominance of the value 1 is now more obvious.

Spineplot
A spineplot, visualizing the imputed values in the variable of interest is printed on the right side of each plot. The bar, which is colored **dark blue**, indicates imputed values in the variable of interest. The **dark orange** colored bar, represents values which are imputed in the variable of interest *and also* in the additional variables.

The missing spineplot in Figure 11b again denotes to the fact, that there are no imputed values in this variable at all.

5.1 Customizing the graphic

Since the important customizable parameters are the same as in the histogram method, please refer to section 4.1 for a detailed explanation.

Subsequently, the R code for a graphical representation of a spinogram with the parameter *only.miss* set to `FALSE` and *selection* set to *all* is given. The resulting plot is shown in Figure 12. This graphic shows a different representation of the graphic in Figure 10c.

```
> vars <- c("Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> spineMiss(sleep_kNN[,vars], delimiter="imp", selection="all", only.miss=FALSE)
```

5.2 Interactive features

This method also supports interactive features. By clicking in the right margin of the plot region, the graphic is switched to the next variable in the given data set. Likewise, by clicking in the left margin, the graphic is switched to the previous variable. Clicking anywhere else on the graphic device quits the interactive session.

The program automatically checks the type of the new variable of interest again and executes the appropriate graphical method. This means, that by switching the variable of interest, the graphic can also switch from a spinogram to a spineplot and vice versa.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

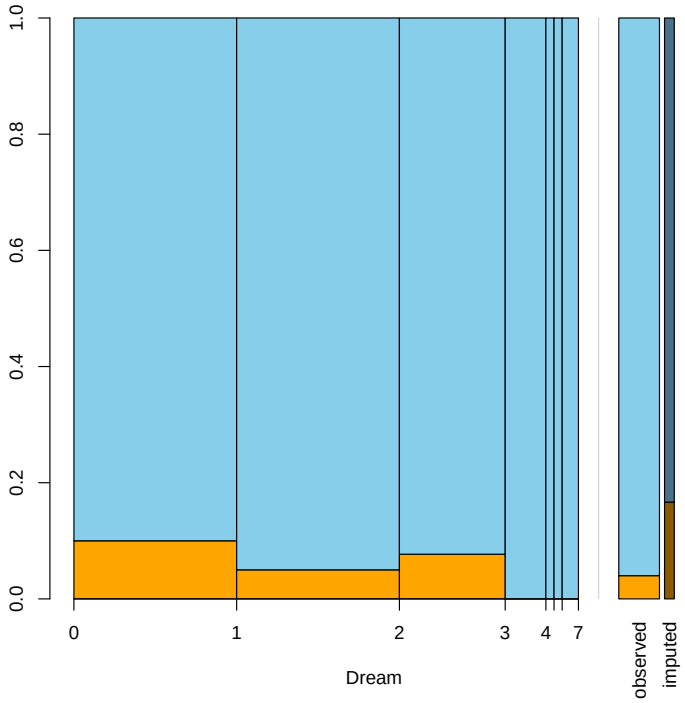


Figure 12: Spinogram of variable *Dream* of the data set *sleep*, imputed values in *NonD* and *Sleep* are highlighted and only.miss is set to `FALSE`

6 Boxplot with imputed missings

Like the previous methods, the well-known boxplot method is also altered to display information about imputed values. The plot consists of three boxplots: First, a normal boxplot of the variable of interest is produced. Second, two boxplots are drawn, which are grouped by observed and imputed values in the additional variables and the values of them in the variable of interest. Additionally, the frequencies of observed and imputed values will be given for each boxplot.

A lot of information can be retrieved from this graphic. It gives a good overview about the distribution of the variable of interest. Also, outliers can be identified easily. Furthermore, by grouping between observed and imputed values, it can be analyzed if the distribution differs. Or it can be reviewed, if, for instance, there are outliers in the variable of interest, which are imputed values in the additional variables. Both can be an indicator that the pattern of missings has a specific structure, because of which the imputed values have been missing.

The graphic can be produced by first selecting the variable of interest in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Boxplot with imputed missings`.

Figure 13 shows a boxplot of the, meanwhile well-known, variable *Dream* of the *sleep* data set. The variables *NonD* and *Sleep* are used for highlighting again.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> pbox(sleep_kNN[,vars], delimiter="_imp", selection="any")
```

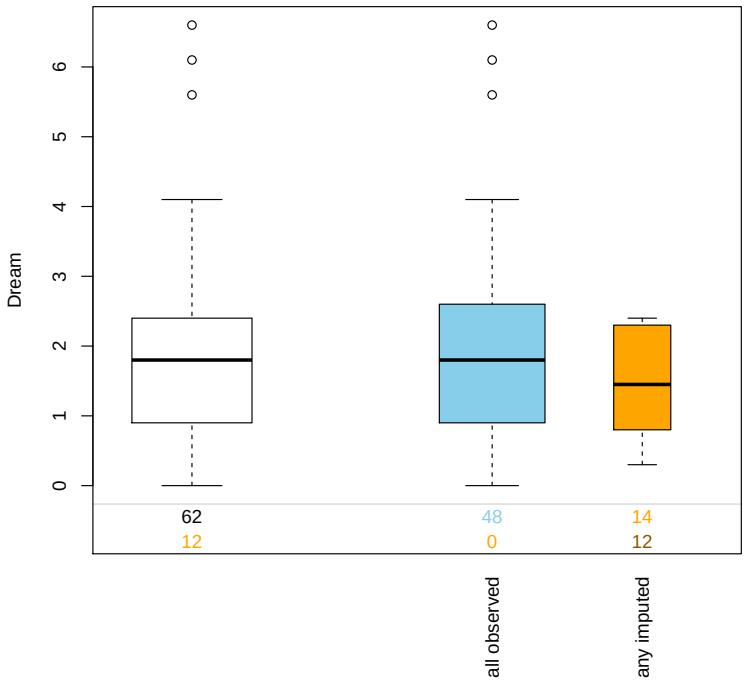


Figure 13: Boxplot of the variable *Dream* of the *sleep* data set, grouped by imputed values in *NonD* or *Sleep*

Subsequently the different plot regions of Figure 13 are explained:

Left Boxplot
A normal boxplot of the variable of interest.

This example shows, that three values of the variable *Dream* are obviously outliers. In the histogram of Figure 10a, this wasn't as visible as it's here.

Right two Boxplots
The first boxplot consists of values, which are observed in the additional variables and the values of them in the variable of interest. Whereas the second one marks values which are imputed in the additional variables and again, the values they're having in the variables of interest.

By looking at the graphic in Figure 13, one can see, that the three outliers of the variable *Dream* are observed values in the additional variables. This excludes the possibility, that the outliers in this variables have been causing the missing values in the other variables. Also the distribution of observed and imputed values can be compared, which apparently differs.

Frequencies
In the bottom of the plot region, the frequencies of observed and imputed values is printed. The first line corresponds to observed values in the variable of interest and their distribution in the two groups and the second line to the accordant imputed

values.

The frequencies of this example show, that the variable *Dream* has 62 values in total, whereat 12 values of them are imputed values. 48 values are also observed in the other variables, none are imputed in the variable *Dream* and observed in the others. Altogether, there are 14 values which are imputed in any of the additional variables and 12 of them are also imputed in the variable *Dream*, which means, that two values are only imputed in the variables *NonD* or *Sleep*. This result can be verified by looking at the aggregation plot of Figure 7.

6.1 Customizing the graphic

This graphic can also be customized when using the command line of R. Subsequently, an example code of a customized boxplot is given and some of the adjustable parameters are explained:

```
> vars <- c("Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> pbox(sleep_kNN[,vars], delimiter="_imp", selection="all")
```

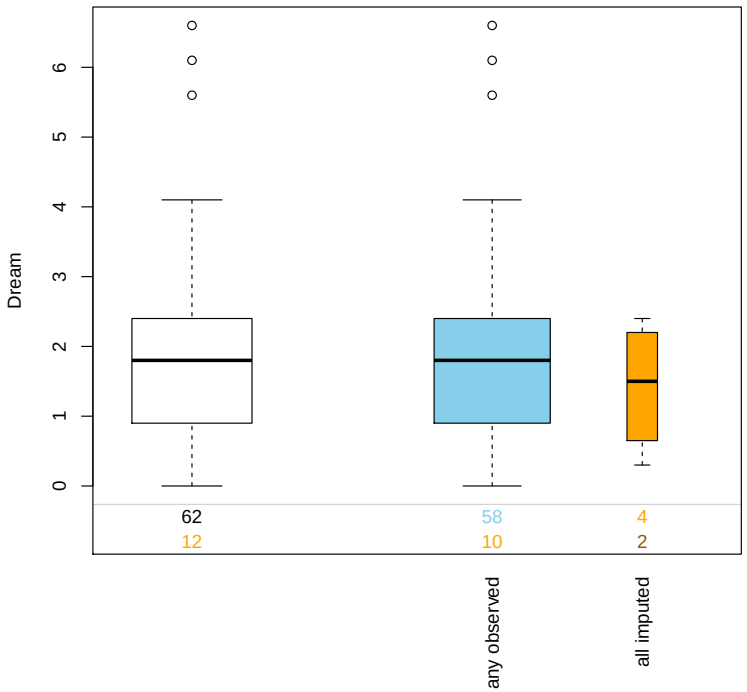


Figure 14: Boxplot of the variable *Dream* of the *sleep* data set, grouped by imputed values in *NonD* and *Sleep*

selection
By altering the *selection*, one can choose to group the values of the variable of interest according to values which are either observed or imputed in *any* or in *all* of the additional variables.

In Figure 14 the same variables as in Figure 13 are used, but the *selection* is set to *all*. The number of values which are imputed in *NonD* and *Sleep* is four. Two values are imputed in all variables. This result can again be verified by looking at the aggregation plot of Figure 7.

Note that also the labels of the grouped boxplots have changed.

numbers
If set to `FALSE`, the output of the frequencies of observed and imputed values is excluded from the graphic.

6.2 Interactive features

This method again supports interactive features. By clicking in the right margin of the plot region, the graphic is switched to the next variable in the given data set. Likewise, by clicking in the left margin, the graphic is switched to the previous variable. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

7 Parallel boxplots

This graphical method is very similar to the boxplot function of the previous chapter (see Section 6), the only difference is, that the additional variables aren't combined. Instead, boxplots for each additional variable are created, whereat the values of the variable of interest are grouped according to the observed and imputed values of each particular variable. *Note, that the variables that don't contain imputed values are excluded from the plot.*

This plot is therefore especially useful to explore whether one continuous variable explains the distribution of missing values in any another variable. [Templ et al., 2012]

The graphic can be produced by first selecting the variable of interest in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. It's displayed by clicking on `Diagnostics` → `Parallel Boxplots`.

Figure 15 shows a parallel boxplot graphic. The same variables of the same data set as in Figure 13 of the previous chapter are used.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Dream","NonD","Sleep","Dream_imp","NonD_imp","Sleep_imp")
> pbox(sleep_kNN[,vars], delimiter="_imp", selection="none")
```

Note, that by using the command line, changing from the normal boxplot to the parallel boxplot method is just a matter of changing the parameter *selection* to *none*.

Subsequently the different plot regions of Figure 15 are explained:

Left Boxplot
This is the same boxplot as in Figure 13.

Boxplot for each additional variable
Two boxplots are drawn for each additional variable that contains imputed values. These boxplots are grouped according to the observed and imputed values in each particular variable, the values they're having in the variable of interest are used for the creation of the boxplots.

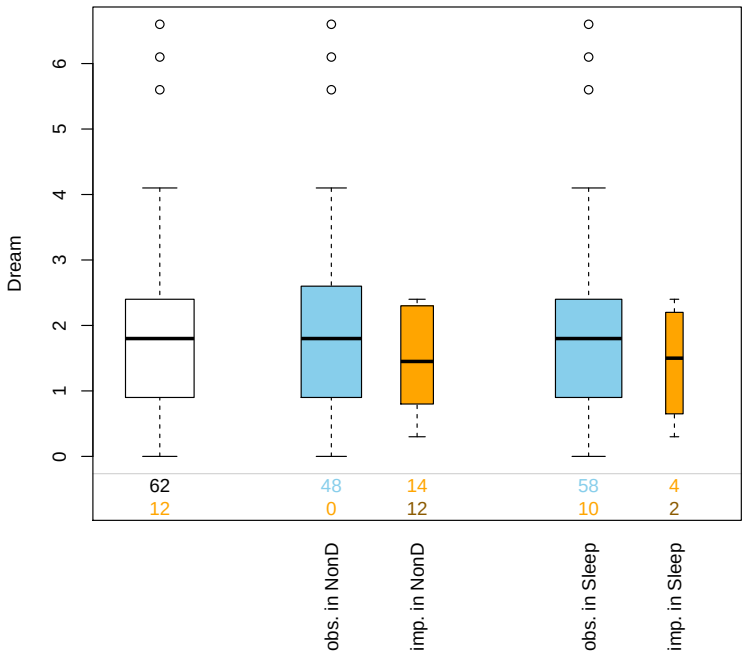


Figure 15: Parallel Boxplot of the variable *Dream* of the *sleep* data set, imputed values in *NonD* and *Sleep* are highlighted separately

In the example graphic, one can see the grouped boxplots for the variables *NonD* and *Sleep*. The distribution of the imputed values hasn't changed significantly.

Frequencies
In the bottom of the plot region, the frequencies of observed and imputed values is printed again. The first line corresponds to observed values in the variable of interest and their distribution in the different groups and the second line to the accordant imputed values.

Reading the second line shows, that all imputed values of *Dream* are also imputed in the variable *NonD*. In contrast, there are only two values, which are imputed in *Sleep* and *Dream*. Both can again be verified by looking at the aggregation plot of Figure 7

7.1 Customizing the graphic

Like the previous boxplot method of section 6, the parameter *numbers* can be adjusted to exclude the output of the frequencies. Therefore, please refer to section 6.1 for a detailed explanation.

Tcl/Tk window
The parallel boxplot method supports the embedment of the graphic in a *Tcl/Tk* window. This is helpful if there is a large number of variables, because scrollbars are added to move from one part of the plot to another.

To enable this option in the GUI, one has to go to `Options` → `Preferences` and

tick the checkbox *Embed multivariate plots in Tcl/Tk* in the *Miscellaneous*-section

When using the command line of R, the function call has to be changed from *pbox* to **TKRpbox**.

7.2 Interactive features

This method supports the same interactive features as the previous boxplot methods. By clicking in the right margin of the plot region, the graphic is switched to the next variable in the given data set. Likewise, by clicking in the left margin, the graphic is switched to the previous variable. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

8 Marginplot

The marginplot is an enhancement to the normal scatterplot, here imputed values are highlighted for each variable. In addition to the scatterplot, boxplots for available and for imputed values, as well as univariate scatterplots for the imputed values are given in the plot margins. Furthermore the frequencies of imputed values are displayed, again for each variable.

This graphical method provides a lot of information. The bivariate scatterplot gives an overview about the bivariate distribution of the chosen variables. Through the highlighting of imputed values, it can be inspected, if there was a certain structure, because of which theses values have been missing. Also errors which could have happened in the imputation process can be revealed. By looking at the boxplots and the univariate scatterplot for each of the variables, it can be analyzed if the distribution of observed and imputed values differs.

It can be produced by first selecting the variables of interest in the *Select Variables* dialog. *Since this is a bivariate plot, only two variables can be picked.* Afterwards, the graphic is displayed by clicking on `Diagnostics` → `Marginplot`.

Figure 16 shows a marginplot of the variables *Air.Temp* and *Humidity* of the data set *tao*.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> marginplot(tao_kNN[,vars], delimiter="imp", alpha=0.6)
```

Subsequently the different plot regions of Figure 16 are explained:

Bivariate scatterplot
A scatterplot of the two variables of interest. The values, which are highlighted with an **orange** color, represent values of the first variable, which are imputed values in the second one. Whereas **dark orange** colored values mark values of the second variable, which are imputed in the first one. If values are imputed in both variables, they are distinguished by a *black* color.

This is a very interesting example. It confirms what could be guessed in the histogram graphic of Figure 9a, the variable *Air.Temp* is clearly separated in two clusters. Furthermore, it reveals, that imputed values of *Humidity* only occur in the first cluster, which are values with low *Air.Temp* and high *Humidity* and that they are imputed

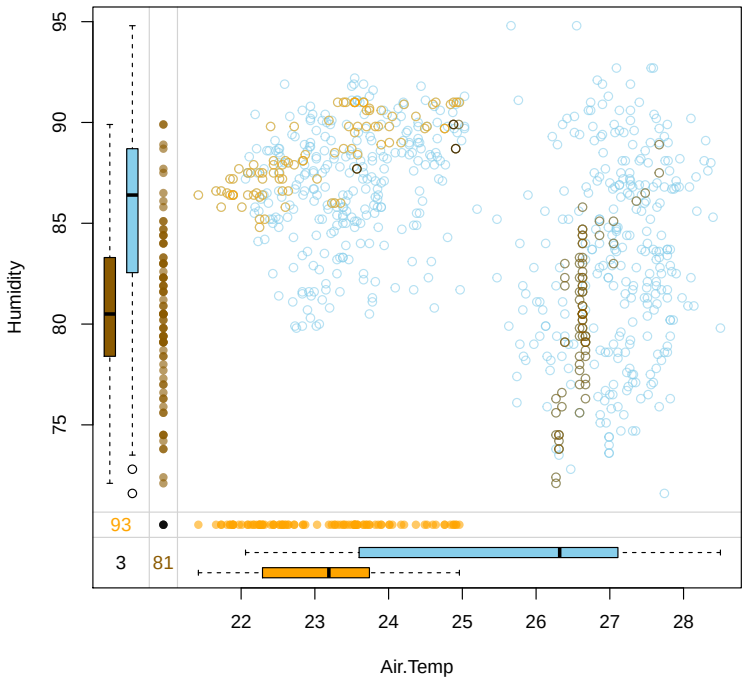


Figure 16: Marginplot of the variables *Air.Temp* and *Humidity* of the *tao* data set

with a value between 84 and 91. On the other hand, most of the imputed values of the variable *Air.Temp* have been imputed with a value between 26 and 27 and mostly occur in the second cluster, which are value with high *Air.Temp*.

Note, that this conclusion is based on the chosen imputation method, which was the k-Nearest Neighbor imputation method in this example. Therefore, the result could be different when applying a different method, which will be demonstrated later on.

Boxplots and univariate scatterplot

The boxplots can be compared to the boxplot method of section 6. They are grouped according to observed and imputed values in the other variable and the values they're having in current one.

The univariate scatterplots show the distribution of the imputed values of each variable in the respective other one. It's especially useful when used with transparency, so that the overplotting of values is prevented.

In this example, the boxplots and univariate scatterplots illustrate that the distribution of observed and imputed values evidently differs in both variables.

Frequencies of imputed values

In the bottom left corner of the plot region, the frequencies of imputed values are displayed. These values are also colored, according to the previous color-schema.

The example graphic supplies the information, that the variable *Air.Temp* contains 81 imputed values and there are 93 imputed values in the variable *Humidity*. Furthermore, 3 values are imputed in both variables.

8.1 Customizing the graphic

Unlike the previous methods, this graphic isn't as customizable. However, there are still a few interesting parameters, which will be explained subsequently:

alpha
The *alpha* parameter controls the level of transparency. It's a numeric value between 0 and 1 and helps to prevent overplotting of the points in the scatterplot. Supplementary, it can be set to `NULL`, which disables transparency and is equivalent to setting it to 1.

In the GUI, it can be changed in the `Options` → `Preferences` menu by changing the slider in the *Set Alpha Value*-Section.

zeros
This is a logical vector of length two, indicating if either of the variables is semi-continuous (i.e. contains a considerable amount of zeros), this parameter can be set to `TRUE` for the particular variable, so that only non-zero observations are used for the drawing of the respective boxplot.

8.2 Comparison of imputation methods

The marginplot can also be used to compare the outcome of different imputation methods. Subsequently in Figure 17, two different plots of the same variables as before are given. In the first plot, the variables are imputed with the *k-Nearest Neighbor* method, which is one of the built-in imputation methods of the package **VIM**. Whereas in the second plot, the variables are imputed with the *mean* method of the package **e1071**.

If the command line of R is preferred, the same plots can be created with following commands:

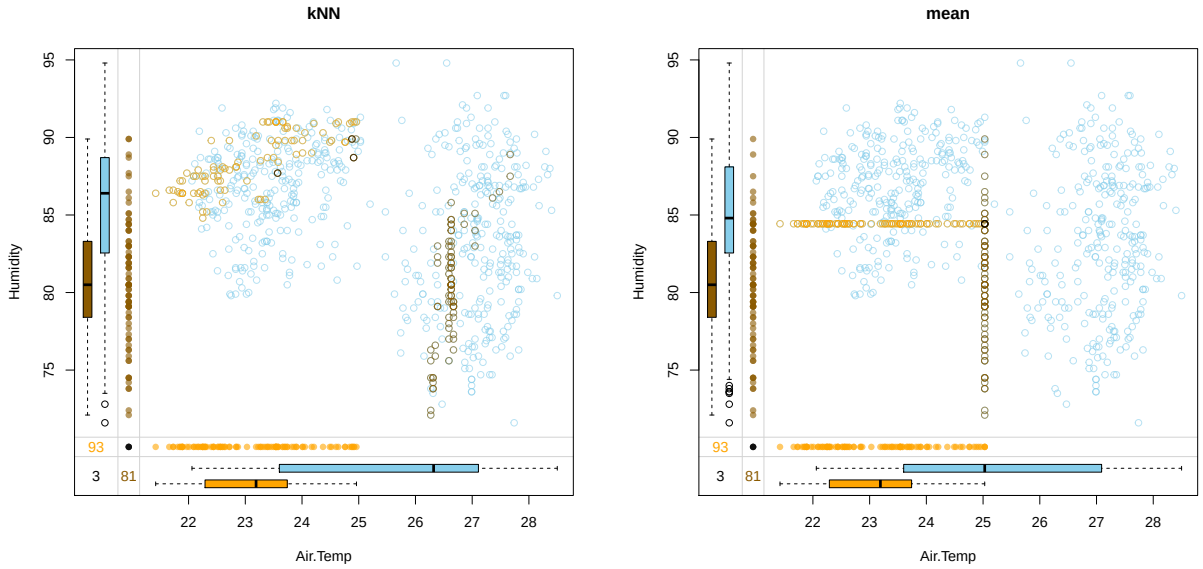
```
> tao_kNN <- kNN(tao, k = 5)
> tao_mean <- as.data.frame(impute(tao, what = "mean"))
> tao_mean <- cbind(tao_mean, tao_kNN[,9:11])
> vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> marginplot(tao_kNN[,vars], delimiter="imp", alpha=0.6, main="kNN")
> marginplot(tao_mean[,vars], delimiter="imp", alpha=0.6, main="mean")
```

Figure 17a shows the same graphic as Figure 16, the missing values are imputed with the most related values in the data set. However, by looking at Figure 17b one can see an obvious result of an *mean* imputation process, the missings of each variable are imputed with the same value (the mean value of the particular variable). The data distribution is completely ruined, which is especially obvious in the variable *Air.Temp* where all values are imputed in between the two clusters.

9 Scatterplot with imputed missings

Like the marginplot of the previous chapter (see Section 8), this method is an adaptation of the normal scatterplot method. Imputed values of only one of the chosen variables are highlighted and also a rug representation of these values is printed on the axis of the other variable. Additionally, percentage coverage ellipses will be drawn.

The scatterplot and the tolerance ellipses provide an overview about the bivariate data distribution and correlation of the chosen variables. Supplementary, the ellipses help to identify outliers. Unlike the marginplot method, imputed values are



(a) Marginplot, kNN-Imputation

(b) Marginplot, mean-Imputation

Figure 17: Marginplots of the variables *Air.Temp* and *Humidity* of the *tao* data set, imputed with different imputation methods

only highlighted for one particular variable, therefore the complexity of the graphic is reduced and the visibility can be improved. Still, special structures of the pattern of the missings, because of which the values have been missing, as well as errors which could have happened in the imputation process can be revealed. The rug representation indicates the values the highlighted values are having in the other variable. This method can be used to compare the outcome of different imputation methods too.

It can be produced by first selecting the variable of interest in the *Select Variables* dialog. *Since this is a bivariate plot, only two variables can be picked.* Afterwards, the graphic is displayed by clicking on `Diagnostics` → `Scatterplot with imputed Missings`.

Figure 18 shows a Scatterplot of the same variables as in the marginplot of Figure 16, *Air.Temp* and *Humidity* of the *tao* data set.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> scattMiss(tao_kNN[,vars], delimiter="imp", alpha=0.6)
```

Subsequently the different plot regions of Figure 18 are explained:

Bivariate scatterplot
Like the marginplot method, a scatterplot of the two chosen variables is printed. However, only imputed values of the one variable of interest are highlighted in **orange** color.

In contrast to the plot of Figure 16 the complexity is reduced a bit, but this example graphic still gives the same results. In this graphic, the imputed values of the variable *Humidity* are highlighted.

Percentage coverage ellipses
The percentage coverage ellipses provides information about the correlation of the

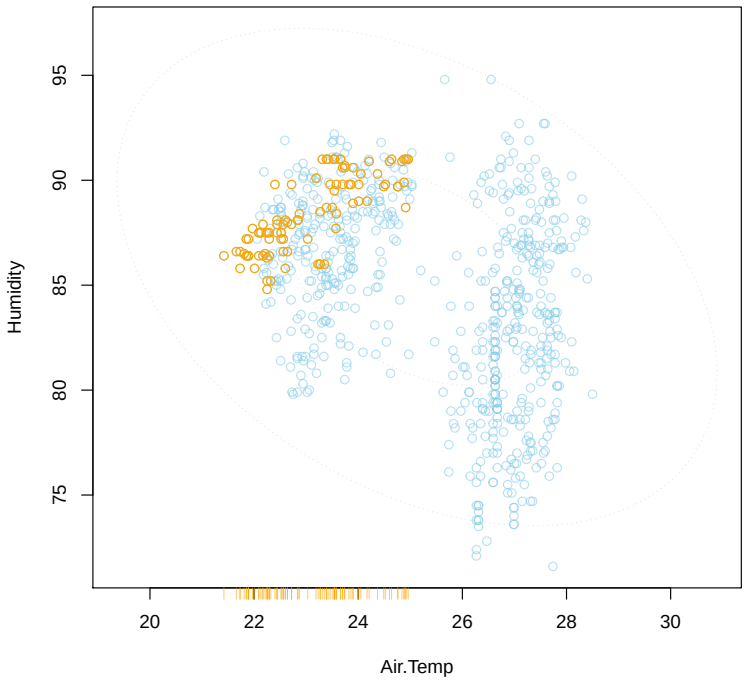


Figure 18: Scatterplot of the variables *Air.Temp* and *Humidity* of the *tao* data set

variables and the general bivariate data distribution. It can also be used to identify bivariate outliers.

Since the data in this example can clearly be split in two clusters, the ellipses don't give much information. The data should be split according to this clusters and each of the clusters should be examined separately.

Rug representation
The rug representation is similar to an univariate scatterplot, each line indicates, that there is an imputed value in the other variable at this value of the current variable.

Since imputed values in the variable *Humidity* are highlighted, the rug representation shows the values, they're having in the variable *Air.Temp*.

9.1 Customizing the graphic

Like the previous methods, there are some parameters can be customized. Subsequently, a few of them are explained:

alpha
The *alpha* parameter controls the level of transparency. It's a numeric value between 0 and 1 and helps to prevent overplotting of the points in the scatterplot. Supplementary, it can be set to `NULL`, which disables transparency and is equivalent to setting it to 1.

In the GUI, it can be changed in the `Options` → `Preferences` menu by changing the slider in the *Set Alpha Value*-Section.

Quantiles
The parameter *quantile* sets the quantiles of the chi-square distribution that should be used for the tolerance ellipses. Supplementary, it can be set to `NULL` to suppress the plotting of the ellipses.

Side
The *side* argument alters the initial variables that are used for the highlighting of imputed values and the rug representation. It can be set to either *1* or *2*, whereas *1* means, that the first variable is taken for the rug representation and the second one for the highlighting and vice-versa.

zeros
This is a logical vector of length two, indicating if either of the variables is semi-continuous (i.e. contain a considerable amount of zeros), this parameter can be set to `TRUE` for the particular variable, so that only non-zero observations of it are used for the computation of the tolerance ellipses.

9.2 Interactive features

Like most of the methods, this method also supports interactive features. Clicking in the bottom margin of the plot region changes the *side* argument to 1, which means, that the rug representation is printed on the axis of the first variable in the data set and the second one is used for the highlighting of imputed values. Likewise, by clicking in the left margin, the side argument is changed to 2. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

10 Bivariate jitter plot

In order to get information about the amount of observed and imputed values of the variables of interest, the bivariate jitter plot can be used. Thereby, the plot is split into up to four regions, according to the existence of imputed values in one or in both variables. Supplementary, this amount is represented by a number in each of the plot region.

Unlike the aggregation graphic of section 3, this graphic focuses on two variables. It particularly provides a better overview about the amount of imputed values in both of the variables if there are many combinations with other variables too. *Note, that if either of the variables doesn't contain imputed values, the respective plot region for the imputed values of this particular variable, as well as the plot region for imputed values in both of the variables, are excluded.*

The graphic can be produced by first selecting the variables of interest in the *Select Variables* dialog. *Since this is a bivariate plot, only two variables can be picked.* Afterwards, the graphic is displayed by clicking on `Diagnostics` → `Bivariate Jitter Plot`.

Figure 19 shows a bivariate jitter plot. Again, the meanwhile well-known variables as in the marginplot of Figure 16 and the scatterplot of Figure 18, *Air.Temp* and *Humidity*, are used.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
> scattJitt(tao_kNN[,vars], delimiter="imp", alpha=0.6)
```

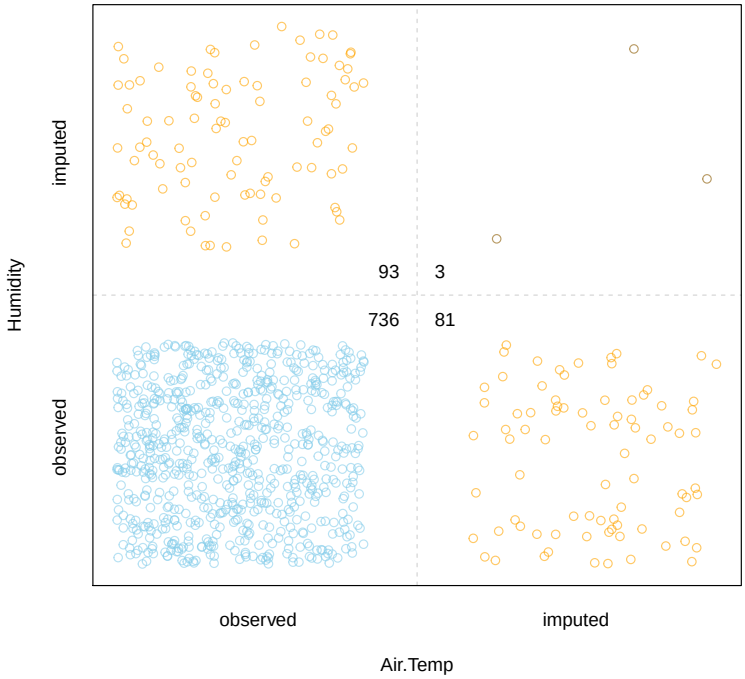


Figure 19: Bivariate Jitter Plot of the variables *Air.Temp* and *Humidity* of the *tao* data set

Subsequently the four plot regions of Figure 19 are explained:

Lower left
Here, all values, which are observed values in both variables are visualized by jittered points. Additionally, the total amount is given by a number in the upper right corner of this plot region.

This example graphic shows the 736 points, that are observed in both variables.

Upper left
The jittered points in the upper left corner, marks values, which are imputed in the second variable, but observed in the first. The total amount is displayed in the lower right corner of this plot region.

Here, one can see the 93 values, which are only imputed in the variable *Humidity*.

Lower right
In the lower right region, values, which are imputed values in the first, but observed in the second variable are shown by the jittered points. Again, the total number is given, here in the upper left corner of this plot region.

The graphic shows the 81 values, which are only imputed in the variable *Air.Temp*.

Upper right

Last, but not least, the upper right plot region illustrates values, which are imputed in both of the variables and again, the total number of this values given in the lower left corner of this plot region.

The example points out, that there are 3 values, which are imputed in both of the variables.

10.1 Customizing the graphic

Like the marginplot of section 8, this graphic isn't as customizable as other methods. However, there are still a few interesting parameters, which will be explained subsequently:

alpha
The *alpha* parameter controls the level of transparency. It's a numeric value between 0 and 1 and helps to prevent overplotting of the points in the jittered plots. Supplementary, it can be set to `NULL`, which disables transparency and is equivalent to setting it to 1.

In the GUI, it can be changed in the `Options` → `Preferences` menu by changing the slider in the *Set Alpha Value*-Section.

numbers
If set to `FALSE`, the output of the frequencies of observed and imputed values is excluded from the graphic.

11 Marginplot matrix

This function creates a scatterplot matrix, with a panel function that is based on the marginplot method of section 8. Therewith, it's possible to get an overview about the distribution of multiple parameters at once.

It can be produced by first selecting the variables of interest in the *Select Variables* dialog. Afterwards, the graphic is displayed by clicking on `Diagnostics` → `Marginplot Matrix`.

Figure 20 shows an example of a marginplot matrix. In addition to the beforehand used variables of the data set *tao*, *Air.Temp* and *Humidity*, the variable *Sea.Surface.Temp* is added to the plot.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","Humidity","Sea.Surface.Temp","Air.Temp_imp","Humidity_imp","S
> marginmatrix(tao_kNN[,vars], delimiter = "_imp", alpha=0.6)
```

In the panels of the plot of Figure 20, one can see the different marginplots. Interestingly, it turns out that the variable *Sea.Surface.Temp* is also clearly separated in two clusters. Values, which are imputed values in the variable *Humidity* almost only appear in the cluster, which indicates low values of *Sea.Surface.Temp*. Whereas imputed values in the variable *Air.Temp* almost only belong to the cluster, which marks high values in *Sea.Surface.Temp*. The highlighted values of *Air.Temp* in the upper right (or lower left) panel reveal, that the imputation process wasn't as good, as it was assumed from plots like Figure 16.

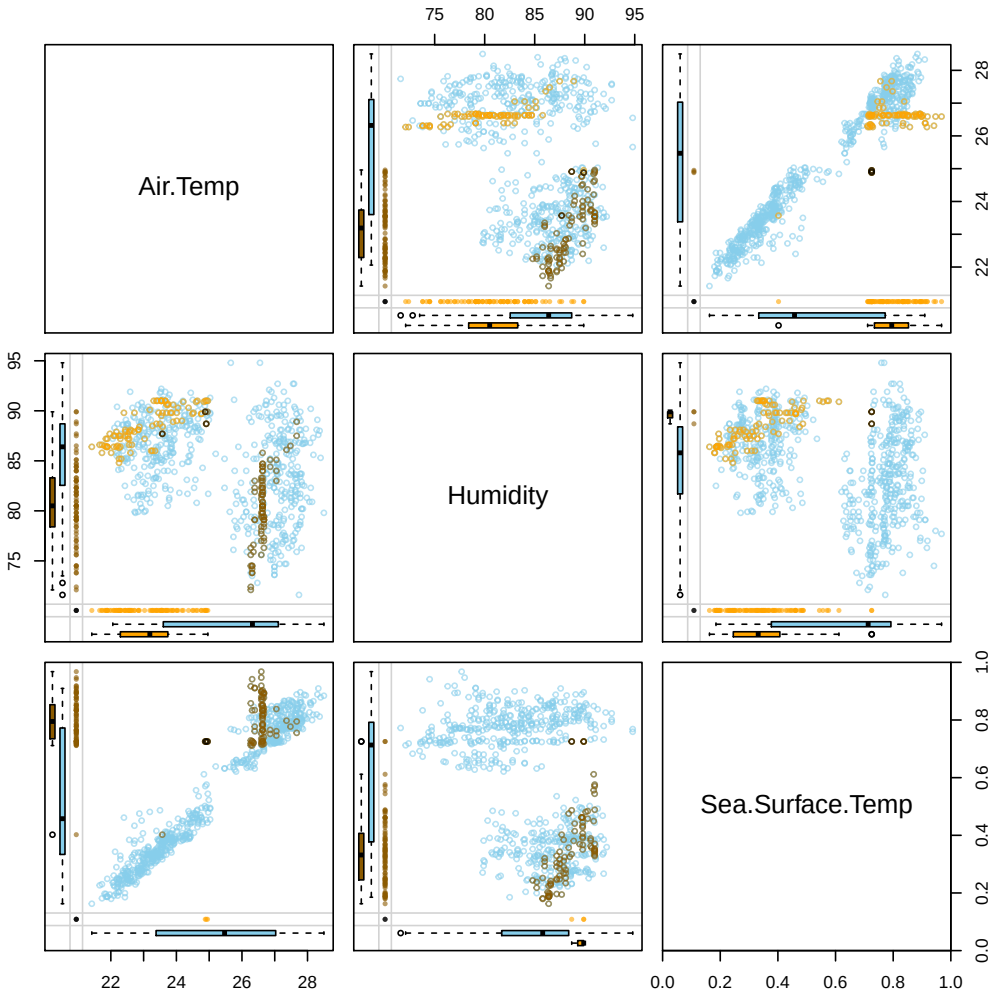


Figure 20: Marginplot Matrix of the variables *Air.Temp*, *Humidity* and *Sea.Surface.Temp* of the *tao* data set

11.1 Customizing the graphic

The marginplot matrix method supports the same adjustable parameters as the marginplot method it is based on, therefore, please refer to section 8.1 for a detailed explanation. However, there is still one additional feature that's worth mentioning:

Tcl/Tk window
Like the parallel boxplot method of section 7, this method also supports the embedment of the graphic in a *Tcl/Tk* window. This is helpful if there is a large number of variables, because scrollbars are added to move from one part of the plot to another.

To enable this option in the GUI, one has to go to `Options` → `Preferences` and tick the checkbox *Embed multivariate plots in Tcl/Tk* in the *Miscellaneous*-section

When using the command line of R, the function call has to be changed from *marginmatrix* to **TKRmarginmatrix**.

12 Scatterplot matrix with imputed missings

Similar to the marginplot matrix method of the previous chapter (see section 11), this function also creates a scatterplot matrix, but with a panel function that is based on the scatterplot method of section 9. It's also intended to get an overview about the distribution of multiple parameters at a single view. In contrast to the marginplot matrix method, certain variables can be chosen to be used for the highlighting of imputed values. In the diagonal panels of the plot, density plots for each variable are drawn. Furthermore, these plots are split according to the observed and the imputed values of the variables which have been selected for highlighting.

It can be produced by first selecting the variables that should be plotted in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Scatterplot Matrix with imputed Missings`.

Figure 21 shows an example of a scatterplot matrix. The same variables, as in the marginplot matrix of Figure 20 are used for this plot. The variable *Air.Temp* is used for the highlighting of imputed values.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","Humidity","Sea.Surface.Temp","Air.Temp_imp","Humidity_imp","S
> scattmatrixMiss(tao_kNN[,vars], highlight="Air.Temp", delimiter = "_imp", alpha=0.6
```

Subsequently the different plot regions of Figure 21 are explained:

Diagonal Panels
In each diagonal panel a density plot is drawn. This plot is split in two lines, the **blue** line corresponds to the density of the values of the particular variable of the panel, which are observed in the variables, that are selected for highlighting. Whereas the **orange** colored line represents the density of the values, which are imputed values in these variables.

In the example, only the variable *Air.Temp* is used for highlighting. It basically confirms the conclusions that have been drawn from the previous graphical methods, but one can see it at a single glance now.

Scatterplot Panels
In the other panels of the plot, the scatterplot method of section 9 is applied to each pair of variables. To improve the visibility, the percentage coverage ellipses are left out in this method. Like in the diagonal panels, the **orange** highlighted values mark values of the particular pair of variables, which are imputed values in the variables, that are chosen for highlighting.

The example shows, that the possibility of choosing which variables should be highlighted in the plot is very convenient. The output of the graphic attest the results of the marginplot matrix of Figure 20.

12.1 Customizing the graphic

This graphic is highly customizable, when using the command line of R. Subsequently, an a few of the adjustable parameters are explained:

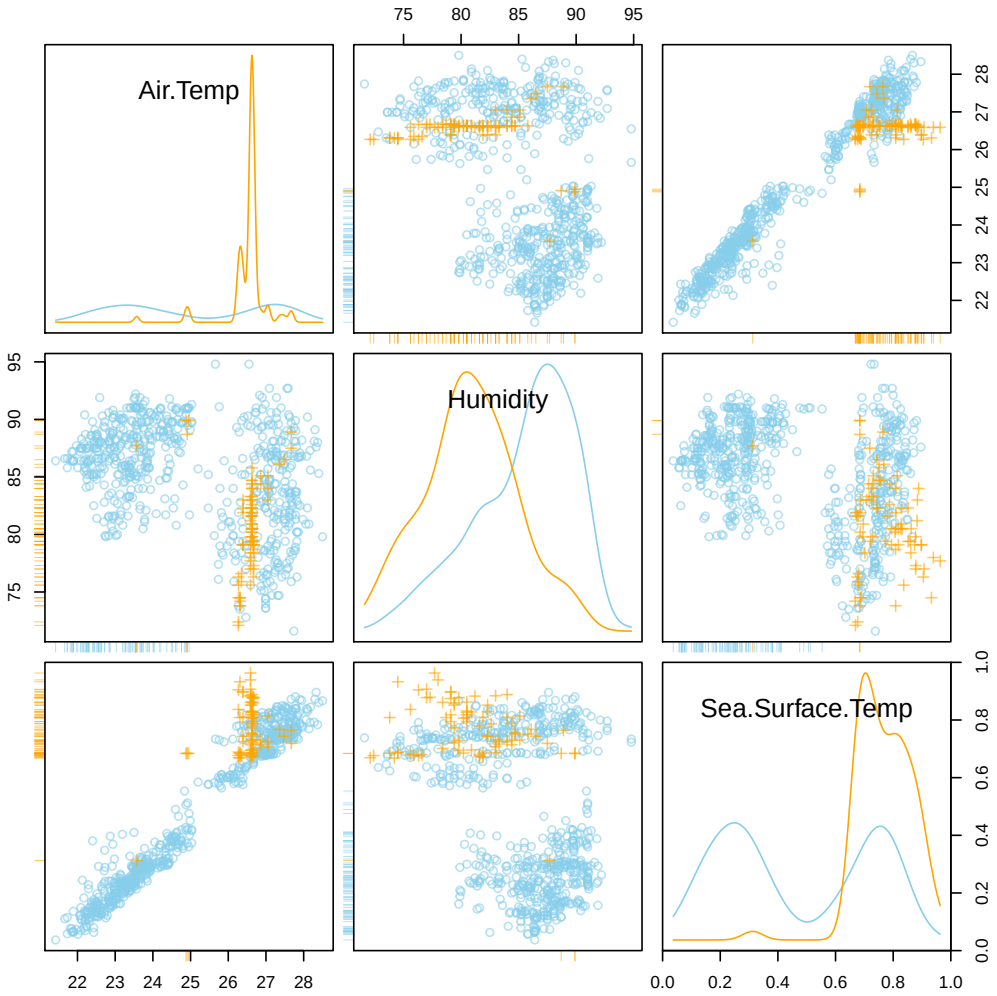


Figure 21: Scatterplot Matrix of the variables *Air.Temp*, *Humidity* and *Sea.Surface.Temp* of the *tao* data set, imputed values in *Air.Temp* are highlighted

alpha

The *alpha* parameter controls the level of transparency. It's a numeric value between 0 and 1 and helps to prevent overplotting of the points in the scatterplots. Supplementary, it can be set to `NULL`, which disables transparency and is equivalent to setting it to 1.

In the GUI, it can be changed in the `Options` → `Preferences` menu by changing the slider in the *Set Alpha Value*-Section.

highlight

The parameter *highlight* is a vector giving the names of the variables, or the position in the given data set, that should be used for highlighting. Alternatively, it can be set to `NULL` (the default value), which means, that all variables are used.

selection

By altering the *selection*, one can choose to highlight values, which are imputed in *any* or in *all* of the variables, that are selected for highlighting.

plotvars

Similar to *highlight*, the parameter *plotvars* is a vector with the names of the variables, or the position in the data set, that should be used for plotting. Is can also be set to `NULL` (the default value), so that all variables are plotted.

By combining the parameters *highlight* and *plotvars*, it's possible to use certain variables just for highlighting and omit the separate scatterplot of this variable. For instance, this can be helpful to find special structures in other variables, because of which the imputed values of the highlighted variables have been missing, or structures, which had an impact on the imputation process.

Figure 22 shows an example of such a scatterplot matrix. The variable *UWind* and *VWind* are plotted and values of the variable *Air.Temp* are used for highlighting only.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Air.Temp","UWind","VWind","Air.Temp_imp")
> scattmatrixMiss(tao_kNN[,vars], highlight="Air.Temp", plotvars=c("UWind","VWind"),
```

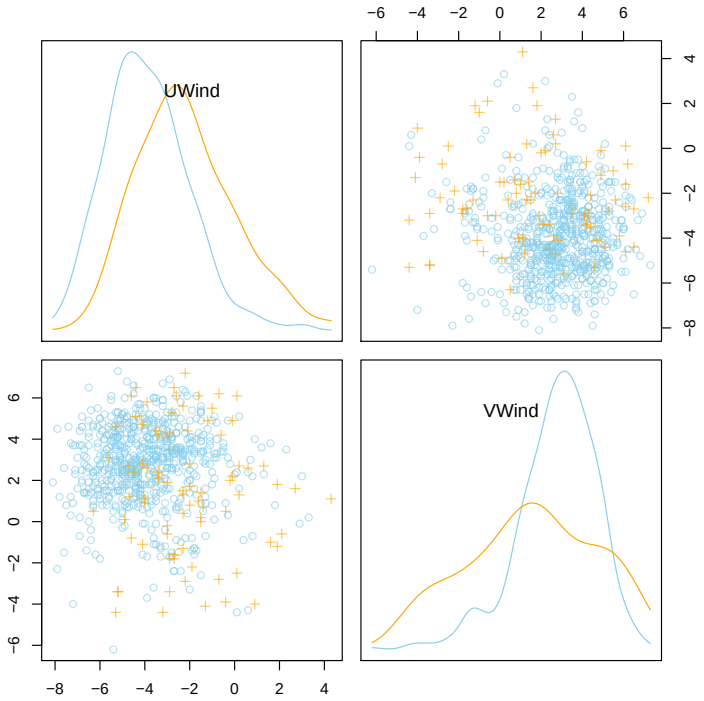


Figure 22: Scatterplot matrix of the variables *UWind* and *VWind* of the *tao* data set, imputed values in *Air.Temp* are only highlighted

This example shows that the plotted variables have no evident influence on the imputed values of *Air.Temp*. Additionally, one can see, that both variables aren't correlated.

Tcl/Tk window

Like the marginplot matrix method of section 11, this method also supports the embedment of the graphic in a *Tcl/Tk* window. This is helpful if there is a large number of variables, because scrollbars are added to move from one part of the plot to another.

To enable this option in the GUI, one has to go to `Options` → `Preferences` and tick the checkbox *Embed multivariate plots in Tcl/Tk* in the *Miscellaneous*-section

When using the command line of R, the function call has to be changed from *scattmatrixMiss* to **TKRscattmatrixMiss**.

12.2 Interactive features

This method supports the interactive selection of the variables, that should be used for highlighting. Clicking in the diagonal panels of the plot, adds the particular variable to to the selection. If the variable was already selected, it is removed instead. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

13 Parallel coordinate plot with imputed missings

Like most methods of the package **VIM**, the parallel coordinate plot is also an adjustment of the standard method to support the highlighting of imputed values. The variables are represented by parallel axes. The data is scaled and each observation is displayed as a continuous line, indicating the value it's having in each of the variables. Imputed values of certain variable can be highlighted. Thereby, the whole line is emphasized to view the values, the particular observation is having along the variables.

This graphical method is very useful to detect multivariate dependencies, or structures, because of which the imputed values have been missing, since it can easily show multiple parameters in one graphic.

It can be produced by first selecting the variables that should be plotted in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Parallel Coordinate Plot with imputed Missings`.

Figure 23 shows an example of a parallel coordinate plot of the *chorizonDL* data set. The variable *Bi* is the only one, which contains imputed values in this subset, hence it is only used for highlighting. The plotted variables are *Al_XRF, Ca_XRF, Fe_XRF, K_XRF, Mg_XRF, Mn_XRF, Na_XRF, P_XRF, Si_XRF* and *Ti_XRF*.

If the command line of R is preferred, the same plot can be created with following commands:

```
> chorizon_kNN <- kNN(chorizonDL[,c(15,101:110)], k=5)
> parcoordMiss(chorizon_kNN, delimiter = "_imp" , plotvars=2:11)
```

The example in Figure 23 shows, that, except of two outliers, the imputed values of *Bi* are having accumulation points throughout most of the variables. Especially when looking at the variables *Al_XRF, P_XRF, Si_XRF* and *Ti_XRF*. This could be an indicator that the variables are having an influence on the structure of the imputed values of *Bi*, thus, the used imputation method has to be chosen with special consideration.

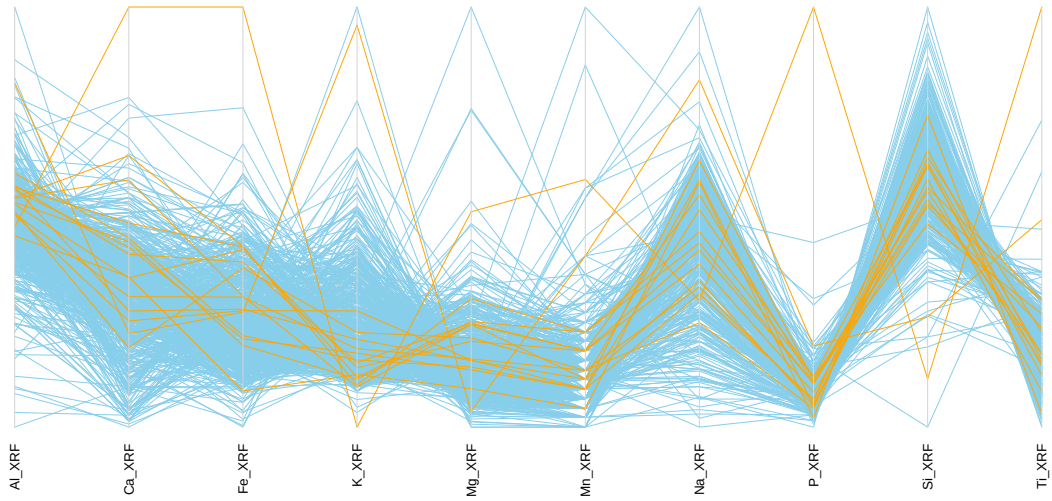


Figure 23: Parallel coordinate plot of the variables *Al_XRF, Ca_XRF, Fe_XRF, K_XRF, Mg_XRF, Mn_XRF, Na_XRF, P_XRF, Si_XRF* and *Ti_XRF* of the *chorizon* data set, imputed values in *Bi* are highlighted

13.1 Customizing the graphic

Since the important adjustable parameters are the same as in the scatterplot matrix function, please refer to section 12.1 for a detailed explanation.

However, when using the command line of R and it's desired to utilize the feature of embedding the graphic in a *Tcl/Tk* window, the function call has to be changed from *parcoordMiss* to **TKRparcoordMiss**.

13.2 Interactive features

Like the previous method, this method again supports the interactive selection of the variables, that should be used for highlighting. A particular variable is added to the selection, by clicking on the respective coordinate axis in the plot. If the variable was already selected, it is removed instead. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

14 Matrix plot

The matrix plot is a very useful multivariate plot, it helps to detect multivariate dependencies and patterns, but it's also possible to find outliers in the data set with this function. Each cell of the data matrix is visualized by a rectangle. Observed values are colored according to a gray scale, whereas missing or imputed values are highlighted by a clearly distinguishable color. To determine the gray level of observed values, the variables are scaled to the interval [0,1], small values are colored in light gray and high values with dark grey. Supplementary, the data matrix can be sorted by the magnitude of a particular variable. The currently selected variable is then printed out in the R console.

Similar to the aggregation method of section 3, variables containing missing and

variable with imputed values can be combined in this graphic. Missings are colored **red**, whereas imputed values are highlighted with an **orange** color.

It can be produced by first selecting the variables of interest in the *Select Variables* dialog. Afterwards, the graphic is displayed by clicking on `Diagnostics` → `Matrix Plot`.

Figure 24 shows an example of a matrix plot of the entire data set *sleep*, which is imputed completely. The variable *Span* is used for the sorting of the data matrix.

If the command line of R is preferred, the same plot can be created with following command:

```
> matrixplot(sleep_kNN, delimiter="_imp", sortby="Span")
```

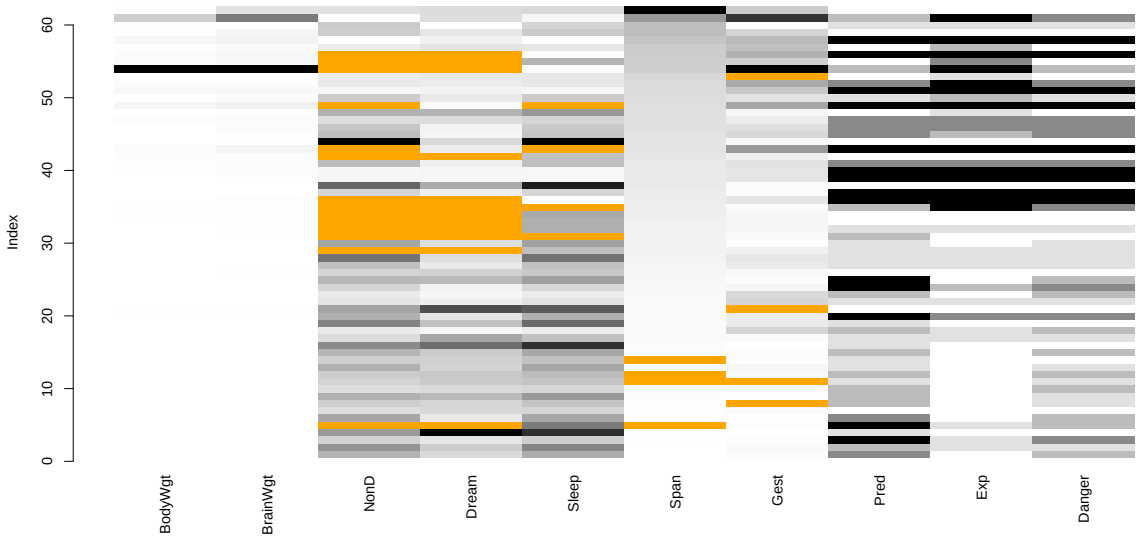


Figure 24: Matrix plot of the complete imputed *sleep* data set

By looking at Figure 24, one can see that there is one clear outlier in variables *BodyWgt*, *BrainWgt* and *Span*. Which can be seen on the fact, that there is almost no color gradient in these variables, instead, there are many white and light gray colored rectangles and one black rectangle. Since the graphic is sorted by the values of the variable *Span*, it reveals, that there are accumulation points of imputed values in *NonD* and *Dream* in a certain value interval of *Span*.

14.1 Customizing the graphic

The matrix plot isn't as customizable as some of the other graphical methods of the package **VIM**. However, there are still some important adjustable parameters, which will be explained subsequently.

col
Instead of a gray scale, colors of the `RGB` or `HCL` colorspace can be supplied for observed values of the data matrix. RGB colors can be given as string, or as objects of class `RGB`. HCL colors can only be specified as objects of class `polarLUV`.

Tcl/Tk window
Like the previous multivariate methods, this method also supports the embedment of the graphic in a *Tcl/Tk* window. This is helpful if there is a large number of

variables, because scrollbars are added to move from one part of the plot to another.

To enable this option in the GUI, one has to go to `Options` → `Preferences` and tick the checkbox *Embed multivariate plots in Tcl/Tk* in the *Miscellaneous*-section

When using the command line of R, the function call has to be changed from *matrixplot* to **TKRmatrixplot**.

14.2 Interactive features

The data matrix of this plot can be sorted by a particular variable interactively. To select the variable that should be used, one has to click on the column of the certain variable in the graphic. Clicking anywhere else on the graphic device quits the interactive session.

Interactivity is always active by default. However, when using the command line of R it can be disabled by setting the parameter *interactive* to `FALSE`.

15 Mosaic plot with imputed missings

The mosaic plot is a graphical representation of multi-way contingency tables, therefore it's mainly intended for categorial variables. The frequencies of the different cells are visualized by area-proportional rectangles (tiles). For constructing a mosaic plot, a rectangle is first split vertically at positions corresponding to the relative frequencies of the categories of a corresponding variable. Then the resulting smaller rectangles are again subdivided according to the conditional probabilities of a second variable. This can be continued for further variables accordingly. Additionally, imputed values in certain variables are highlighted in order to explore their structure. [Templ et al., 2012]

It can be produced by first selecting the variables that should be plotted in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the additional variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Mosaic Plot with imputed Missings`.

Figure 25 shows an example of a mosaic plot of the variables *Pred* and *Exp* of the data set *sleep*. Imputed values of *NonD* are highlighted.

If the command line of R is preferred, the same plot can be created with following command:

```
> mosaicMiss(sleep_kNN, highlight=3, plotvars=8:10, delimiter="_imp", miss.labels=FAl
```

The example in Figure 25 shows, that most of the values occur if the values of the variables are both in the category 1 or in category 5. Also, many values in this categories are imputed values in the variable *NonD*. This also applies to values, which are in category 3 in the variable *Pred* and 5 in *Exp* respectively. The values, which are of category 1 in *Pred* and 4 in *Exp* are all imputed values in *NonD*. Category 1 is dominant in *Exp*, whereas the same can be said about category 2 of the variable *Pred*.

15.1 Customizing the graphic

In terms of customization, the mosaic plot is very similar to the scatterplot matrix method of section 12, except of the parameter *alpha*, which isn't available. Thus, please refer to section 12.1 for a detailed explanation of the adjustable parameters.

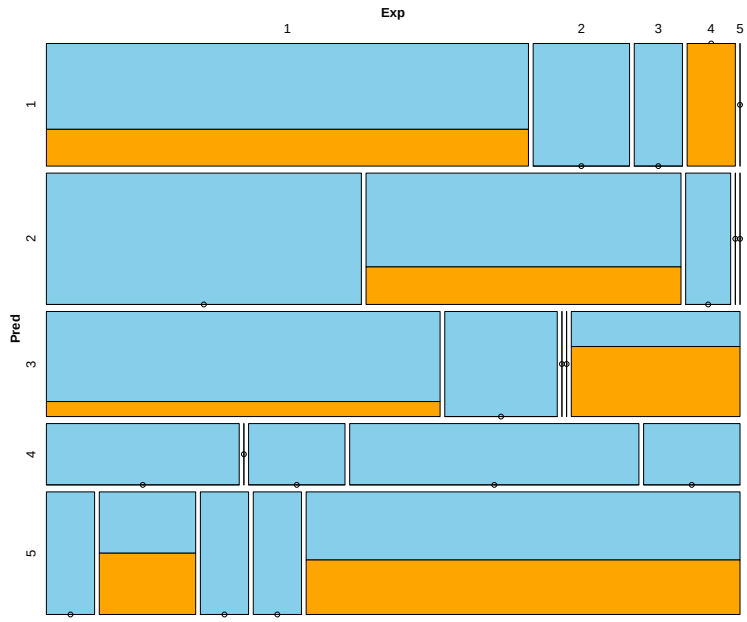


Figure 25: Mosaic plot of the variables *Pred* and *Exp* of the *sleep* data set, imputed values in *NonD* are highlighted

16 Map of imputed missings

It may occur, that the structure, because of which the imputed values have been missing is affected by spatial patterns. The *map of imputed missings* method is developed to explore this matter. It plots variables in a given map and highlights imputed values. Therefore, the x- and y-coordinates, which indicate the point of each value on the map, have to be supplied.

This method is focused on the coordinates, hence it only indicates that there are observations at certain points of the map and highlights imputed values. The graphic will be the same for different imputation methods. Still it provides valuable information about the structure of the values, which have been missing.

The graphic can be produced by first selecting the background map in the *Select Background Map* dialog, which can be opened by clicking on `Data` → `Background Map`. In addition, the variables which represent the x- and y-coordinates have to be defined in the *Select Coordinates* section of this dialog. After selecting the variables that should be used for the plot in the *Select Variables* dialog and choosing, whether imputed values in *any* or *all* of the variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Map of imputed Missings`.

Figure 26 shows an example of a map of imputed missings. The variables *As*, *Bi* and *Ca* of the data set *chorizonDL* are plotted on the *kola.background*-map (which is also included in the package **VIM**), whereas imputed values in *any* of the variables are highlighted. The map is linked to the data by the variables *XCOO* and *YCOO*.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("As","Bi","Ca","As_imp","Bi_imp")
> coo <- chorizon_kNN[, c("XCOO", "YCOO")]
> mapMiss(chorizon_kNN[,vars], coo, map=kola.background, delimiter="_imp", alpha=0.6)
```

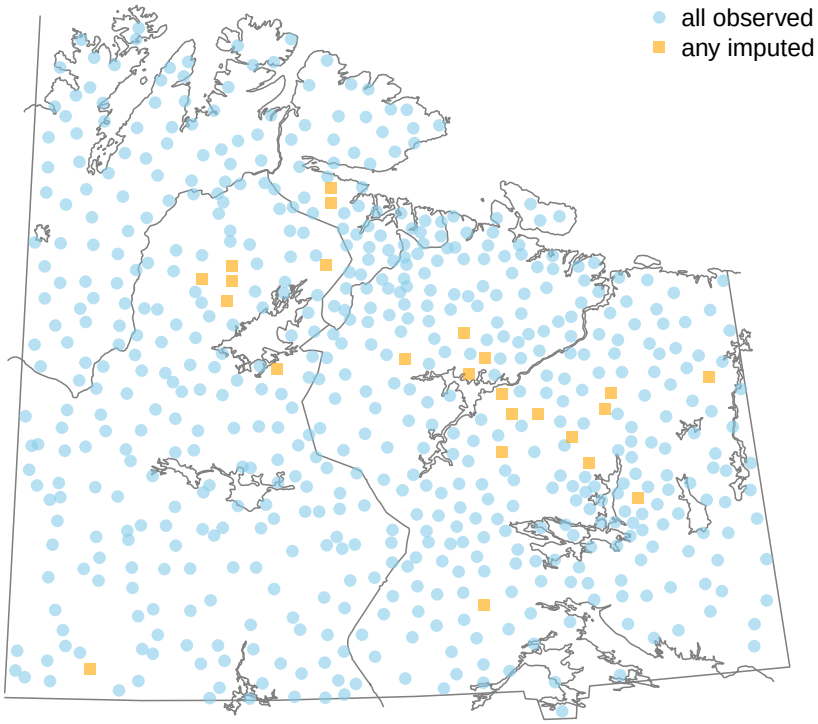


Figure 26: Map of imputed missings of the variables *As*, *Bi* and *Ca* of the *chorizonDL* data set, plotted on the *kola.background*-map. Imputed values in any of the variables are highlighted.

The highlighted imputed values of the variables *As*, *Bi* or *Ca* of Figure 26 reveal, that there is a regional dependency. This indicates, that the distribution of the missing values was not missing completely at random. Therefore, the imputation method has to be chosen with special consideration. Note, that the chosen *selection* method, which is *any* in this example, can be seen in the legend in the top right corner of the graphic.

16.1 Customizing the graphic

This graphic can also customized, like most of the methods, when using the command line of R. Subsequently, two important parameters are explained:

alpha
The *alpha* parameter controls the level of transparency. It's a numeric value between 0 and 1 and helps to prevent overplotting of the points in the plot. Supplementary, it can be set to `NULL`, which disables transparency and is equivalent to setting it to 1.

In the GUI, it can be changed in the `Options` → `Preferences` menu by changing the slider in the *Set Alpha Value*-Section.

selection
By altering the *selection*, one can choose to highlight values, which are imputed in *any* or in *all* of the variables.

16.2 Interactive features

This method again supports interactive features. Clicking on any point in the graphic prints information about the values it's having in the supplied variables in the R console. This is intended to further analyze the structure of the values. Clicking in a region that does not contain any points quits the interactive session.

17 Growing dot map with imputed missings

Like the *map of imputed missing* function of the previous chapter (see section 16), this method is also developed to analyze geographical data. However, the values of the variable of interest are represented by growing dots instead of normal points in this function. Imputed values are highlighted again.

This graphical method is intended to reveal relationships of the imputed values to both their spatial location and also of the values of the variable of interest. The imputed values are distinguished by their occurrence in the supplied variables. Imputed values of the variable of interest are represented by a *dark blue* rectangle. Values, which are imputed in both, the variable of interest and *any* or *all* (depending on the *selection*) of the additional variables are plotted as a *dark orange* rectangle. Last but not least, the growing dot representations of values of the variable of interest are colored *orange* for values, which are imputed in the additional variables only.

The graphic can be produced by first selecting the background map in the *Select Background Map* dialog, which can be opened by clicking on `Data` → `Background Map`. In addition, the variables for the x- and y-coordinate have to be defined in the *Select Coordinates* section of this dialog. Subsequently, the variable that should be used for the growing dots need to be selected in the *Select Variables* dialog and the variable(s), which should be used for highlighting, in the *Highlight Variables in Plots* dialog. After choosing, whether imputed values in *any* or *all* of the variables should be highlighted in the *Selection for Highlighting* dialog, the graphic is displayed by clicking on `Diagnostics` → `Growing Dot Map with imputed Missings`.

Figure 27 shows an example of a map with growing dots. The variable *Ca* of the data set *chorizonDL* is plotted on the *kola.background*-map, whereas imputed values in the variables *As* and *Bi* highlighted. Again, the map is linked to the data by the variables *XCOO* or *YCOO*.

If the command line of R is preferred, the same plot can be created with following commands:

```
> vars <- c("Ca","As","Bi","As_imp","Bi_imp")
> coo <- chorizon_kNN[, c("XCOO", "YCOO")]
> growdotMiss(chorizon_kNN[,vars], coo, kola.background, delimiter="_imp", alpha=0.6)
```

The example of Figure 27 gives similar results to the graphic of Figure 26, a regional dependency is existing. However, the growing dot representation of the values of *Ca* shows, that there is no relationship between the values of this variable and the imputed values in the others.

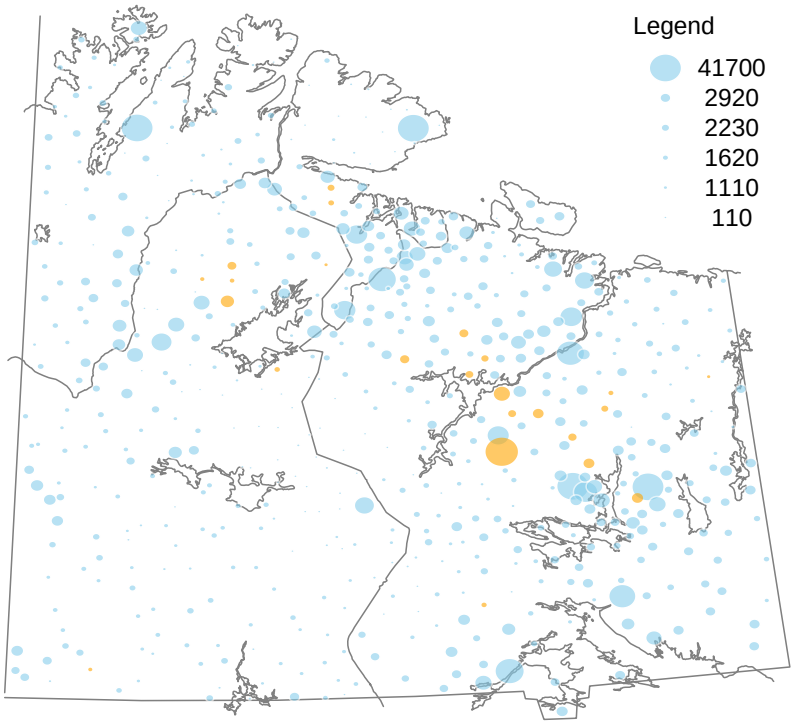


Figure 27: Growing dot map of the variable *Ca* of the *chorizonDL* data set, plotted on the *kola.background*-map. Imputed values in *As* or *Bi* are highlighted.

17.1 Customizing the graphic

Since the important adjustable parameters are the same as in the map of imputed missing function, please refer to section 16.1 for a detailed explanation.

17.2 Interactive features

This method also supports interactive features. Clicking on any point in the graphic prints information about the values it's having in the supplied variables in the R console. This is intended to further analyze the structure of the distribution of the values. Clicking in a region that does not contain any points quits the interactive session.

18 Conclusion

Package **VIM** offers various very interesting graphical methods to analyze the structure of missing and imputed values in a simple manner. By using this package, it's possible to reveal patterns and mechanisms of the missing values, as well as errors which happened in the imputation process. The interactive features of the implemented methods alleviate the information retrieval, by making it possible to quickly

change between different variables in the data set or getting information about certain points in the plot. Additionally, a graphical user interface is provided for an easy handling of the included functions. Supplementary, **VIM** can be used to produce high-quality graphics for publications.

References

T. Allison and D. V. Cicchetti. Sleep in mammals: ecological and constitutional correlates. *Science*, 194(4266):732–734, 1976.

E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2011. URL http://CRAN.R-project.org/package=e1071. R package version 1.6.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL http://www.R-project.org. ISBN 3-900051-07-0.

C. Reimann, P. Filzmoser, R.G. Garrett, and R. Dutter. *Statistical Data Analysis Explained: Applied Environmental Statistics with R*. Wiley, Hoboken, New Jersey, 2008.

D. Swayne, D. Lang, A. Buja, and D. Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Comput Stat Data Anal*, 43(4):423–444, 2003.

M. Templ and A. Alfons. *An application of VIM, the R package for visualization of missing values, to EU-SILC data*, August 2009. URL http://cran.r-project.org/web/packages/VIM/vignettes/VIM-EU-SILC.pdf.

M. Templ, A. Alfons, A. Kowarik, and B. Prantner. *VIM: Visualization and Imputation of Missing Values*, 2011a. URL http://CRAN.R-project.org/package=VIM. R package version 3.0.0.

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

M. Templ, A. Alfons, and P. Filzmoser. Exploring incomplete data using visualization techniques. *Advances in Data Analyis and Classification*, 2012. Accepted for publication.