Journal of Statistical Software

December 2015, Volume 68, Issue 6. doi: 10.18637/jss.v068.i06

Visually Exploring Missing Values in Multivariable Data Using a Graphical User Interface

Xiaoyue Cheng
University of Nebraska at Omaha

Dianne Cook
Monash University

Heike Hofmann
Iowa State University

Abstract

Missing values are common in data, and usually require attention in order to conduct the statistical analysis. One of the first steps is to explore the structure of the missing values, and how missingness relates to the other collected variables. This article describes an R package, that provides a graphical user interface (GUI) designed to help explore the missing data structure and to examine the results of different imputation methods. The GUI provides numerical and graphical summaries conditional on missingness, and includes imputations using fixed values, multiple imputations and nearest neighbors.

Keywords: missing values, imputation, exploratory data analysis, statistical graphics, data visualization, graphical user interface.

1. Introduction

Missing values are a very common problem affecting data analysis. Many imputation methods have been developed but little has been done for exploring the missing data structure visually. Most plotting methods handle missing values by simply removing the incomplete records with or without a warning, especially when the data are continuous. Most statistical functions provide a limited list of methods for handling missing values, such as, delete all cases with any missing values, delete pairwise or on single variables only.

The issue is, that in order to decide what to do with the missing values before analyzing the data, we need to understand what the distribution of the missing values is, and how the missingness depends on the other collected variables. A few R packages, like **Hmisc** (Harrell 2015), **norm** (Novo and Schafer 2013), and **mice** (van Buuren and Groothuis-Oudshoorn 2011), have some routines for summarizing the number of missing values by variable, and by case, in preparation for imputing the missing values. To understand the distribution of missings

versus non-missings it is also important to make plots of the data.

For model-based imputation methods, it is important to check assumptions like missing completely at random (MCAR) or missing at random (MAR). These are not easy to verify. Little (1988) provided tests of the MCAR assumption, under normality conditions, and Jaeger (2006) proposed a test for MAR under some distributional conditions. Both tests employ inference based on likelihood ratios, and it has been cautioned that the tests are sensitive to model misspecification (Little 1988). Visual exploration of the missingness can help check the assumptions: It cannot prove that any randomness assumption holds but visual checks can be used to reject MCAR assumptions, or suggest what dependencies exist, and should be incorporated into imputation for MAR data.

Some existing work describing visual exploration of missingness, and implementations, can be found in Unwin, Hawkins, Hofmann, and Siegl (1996), Swayne and Buja (1998), and Templ and Filzmoser (2008). Package **MANET** (Unwin *et al.* 1996) implements interactive methods for missing data. It presents the segmented barcharts of missing versus non-missing values for each variable, and with its many plot types like histograms, scatterplots, and mosaic plots, encourages the user to select cases that are missing on any variable to highlight in other plots. This enables the user to explore the missing status dependence in the distributions of the complete cases of other variables. **XGobi** (Swayne, Cook, and Buja 1998), which implements the ideas described in Swayne and Buja (1998), is similar to **MANET**, but focuses on interactive graphics for exploring missing values in real-valued data. It creates a shadow matrix of the original data where entries are 0 (complete) or 1 (missing value). This additional data structure allows the user to explore the multivariate pattern of missing values, the dependence between missing value status and complete cases, and compare imputation methods. These ideas were re-implemented in **GGobi** (Swayne, Temple Lang, Buja, and Cook 2003).

For the statistical computing environment R (R Core Team 2014), package **VIM** (Templ, Alfons, Kowarik, and Prantner 2015) provides a GUI via **VIMGUI** (Schopfhauser, Templ, Alfons, Kowarik, and Prantner 2013), to explore the missing data structure and the quality of several single imputation methods (kNN, hotdeck, irmi). Some packages for multiple imputation have interfaces for easy manipulation, such as, for example, **migui**, `AmeliaView()` and **miP**. Package **migui** (Goodrich 2015) is an interface for **mi** (Su, Gelman, Hill, and Yajima 2011), which implements multiple imputation via Bayesian models and weakly informative prior distributions. The function `AmeliaView()` in **Amelia** (Honaker, King, and Blackwell 2011), generates a GUI, to implement its "EM with bootstrapping" algorithm. Package **miP** (Brix 2012) adopts **VIM** to visualize the imputation results from packages **mice**, **mi**, and **Amelia**.

This current work describes a new package for R, **MissingDataGUI** (Cheng, Cook, and Hofmann 2015), which allows the exploration of the missing data structure, and the comparison of different imputations, using static graphics and numerical summaries. The GUI makes these methods accessible for novice users. This work builds on the ideas developed in Unwin *et al.* (1996) and Swayne and Buja (1998). The package utilizes routines in **Hmisc**, **norm**, **mice**, and **mi** for multiple imputation, and provides several other routines including kNN, random sampling and fixed values for single imputation. It is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=MissingDataGUI`. Section 2 explains the GUI design, functionality and rationale. Section 3 gives a usage example.

2. Functionality

2.1. Overview of the missing data GUI

The appearance of the missing data GUI is shown in Figure 1. (Section 3 describes the data set.) All variables in the data set along with the variable type and the percentages of NA's are listed on the top left (region 1). The categorical variables (factor, ordinal factor, and character), auto-detected by their type, are shown on the bottom left as the potential conditioning variables (region 2). The variables having missing values are displayed under "Color.by.the.missing.of" on the top center (region 3). The graphical summary will distinguish the imputations from the observations by two colors, yellow (missing) versus blue (non-missing). This panel is used to choose what missing structure to color. Selecting the first row "Missing Any Variables" means that the color will depend on whether the case has

Figure 1: Overview of the missing data GUI. Region 1 contains the list of variables, variable type, and summary of missings on that variable. Region 2 has a list of the categorical variables that can be used for conditioning plots and imputations. Region 3 has a selection panel for selection of coloring by different types of missingness in the plots. Region 4 contains a radio button for the selection of imputation methods. Region 5 allows to select the plot type, and region 6 the numeric or graphical summaries and some output routines. The summaries are displayed in region 7.

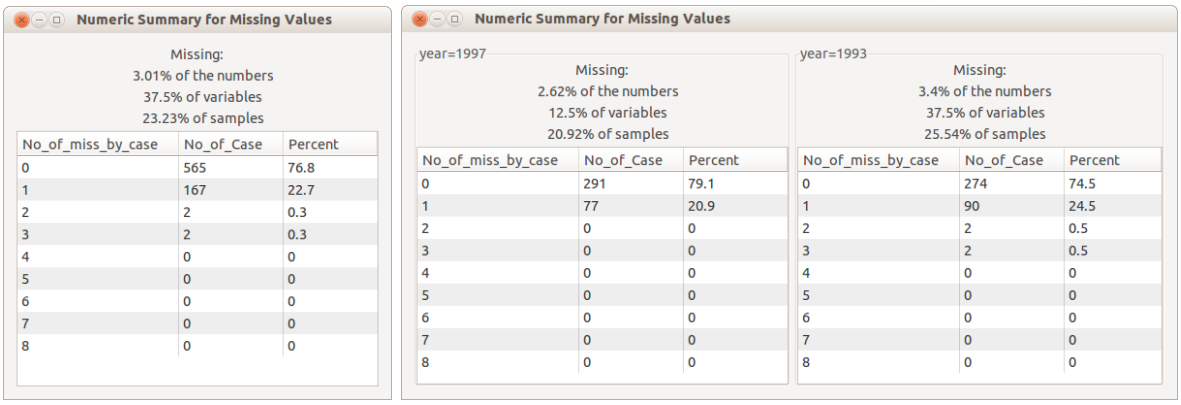


Figure 2: A numerical summary of missing values in the data set is shown in a pop-up window. The left panel is the overall summary. The right panel shows the summary conditioned on "year". The percentages of missings by total number of data values, by variables and by cases, is shown on the top. This data set has 8 variables and the missing values by variable are summarized in the bottom table. No cases have more than 3 missing values, 76.8% of cases are complete, 22.7% of cases have one missing value, and only 4 cases have more than one missing values. The right panel shows that the missingness pattern is different for each year.

missing values in any variables. The second row "Missing on Selected Variables" means the graph is colored by whether the case has missings in the selected variable. "Method" (region 4) and "Graph Type" (region 5) are two widgets illustrated in Sections 2.3 and 2.4. On the top right (region 6) there are five buttons: "Summary" can create a window as described in Section 2.2; "Plot" produces the plots in the graphics panel on the bottom right of the GUI (region 7); "Export data" saves the imputed data into a file or to an R data frame; "Save plot" saves the plots in region 7 to PNG files; "Quit" destroys the main GUI window and the derived child windows.

2.2. Summary of missing values

Numerical summaries

To investigate missingness in a data set, a good start point is to examine the numerical summaries of the missings. The "Summary" button will open a window with the overall missingness information (Figure 2, left panel) or conditional summary (Figure 2, right panel), depending on whether conditioning variables are chosen. Both summary windows present the percent of the values that are missing, the percent of variables that contain missing values, the percent of the cases that have at least one missing value, along with a tabulation of the number of values missing per case. The style of the table follows the summary provided by package **norm**. In Figure 2 (left) it can be seen that the data has two observations with 3 missing values, another two with 2 missing values, 167 observations with one missing value and 565 observations are complete. By percentages, 76.8% of the cases have no missings. Figure 2 (right) is conditioned on the variable "year", which produced two boxes for 1993 and 1997 respectively. We can see that there are fewer missing values in 1997 than 1993, and all the observations having more than 1 missings appeared in 1993.

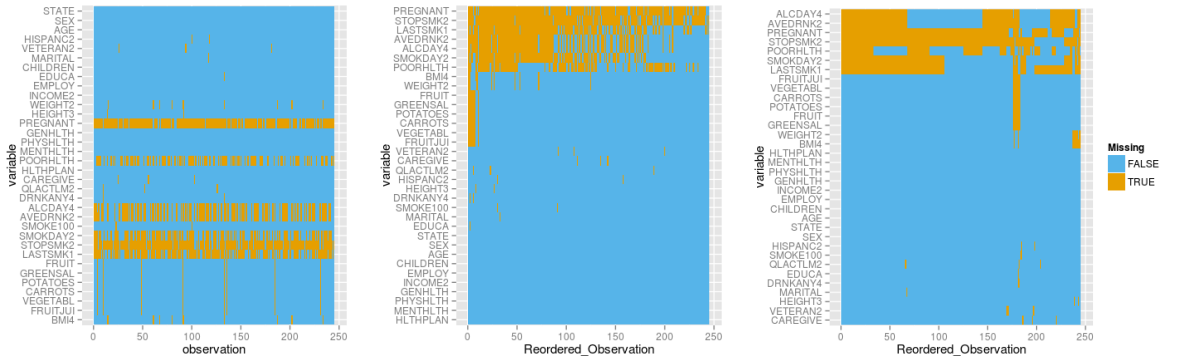


Figure 3: Missingness maps, same data but different ordering of variables (rows) and cases (columns): (left) raw data order, (middle) variables and cases sorted by decreasing number of missing values, (right) sorted by hierarchical clustering of missingness. From the raw data missingness map, the horizontal stripes indicate several variables have many missings, and the vertical stripes near the bottom indicate some structural missing cases. When variables and cases are sorted by missingness rate (middle), variables with missings often have missings on the same cases, and the additional few sporadic missing values can be easily spotted. Using the clustered missingness map (right) the blocks of missings on variables and cases is more easily seen.

Missingness map

The missingness map (Figure 3) provides a graphical summary of the missingness patterns. Like the shadow matrix used in **GGobi**, the missingness map shows the position of missing values relative to variables and cases. The **R** packages **Amelia** and **VIM** also provide versions of missingness maps. Organizing the missing values into blocks can be achieved by re-ordering variables and case ids, making it easier to see missingness patterns, especially for large data sets. Two re-ordered missingness maps are shown in Figure 3. One arranges the variables and cases by the number of missings, from the largest to the smallest; the other applies hierarchical clustering to both rows and columns. The strength of the missingness map is to reveal whether the missings occur at some variables simultaneously. If so, then a similar missingness pattern may indicate some association between the variables. If the missings happen at some observations synchronously, then it suggests dependence between those observations.

Figure 3 displays 245 observations and 34 variables for the data set `brfss` (described in Section 3). From the missingness maps we can see that most of the missings occurred in seven variables. The missingness on some variables occurs synchronously, indicating association. Users of the data set should check the data collection procedures for these variables. For example, in this data set, questions about the drinking time and amount (ALCDAY4 and AVEDRNK2, the top two variables in the right panel) were both skipped when the subject answered a previous question with "did not drink in the past 30 days".

2.3. Imputation

A number of imputation methods are available in the package. The purpose is two-fold: to enable exploring dependence between missings or non-missings, and also to produce a complete data set for later analysis. A few criteria were considered in the choices of methods

Method	Description	Deter-ministic	Uni-variate	Multiple imp.
Below 10%	below 10% of the range	×	×	
Simple	overall median	×	×	
	overall mean	×	×	
	random value		×	
Neighbor	mean of the nearest neighbors	×		
	random nearest neighbor			
MI:areg	predictive mean matching			×
MI:norm	multivariate normal model			×
MI:mice	multivariate imp. by chained equations			×
MI:mi	multiple iterative regression imputation			×

Table 1: Imputation methods included in the missing data GUI. Strictly speaking, "Below 10%" is not an imputation method, but a way to put the missing values in the same graph with the observations. "Deterministic" indicates whether the method has a stochastic component or not. "Univariate" means whether the imputation only uses the individual variable where imputation is needed, or makes use of other variables as well. "Multiple imp." indicates whether the methods is a type of multiple imputation that will provide multiple samples to impute the missings.

to make available and in the design: (1) easy to understand and implement; (2) computing complexity is medium or low; (3) adaptability to different situations, i.e., no strong model assumptions. Not all of the imputation methods available in R are available in the package because (1) there are too many methods and variations, so it is not practical to include all, and (2) there is also the possibility that users may use their own method and import the result to the missing data GUI for exploration.

The seven imputation methods provided are: "Below 10%", "Simple", "Neighbor", "MI:areg", "MI:norm", "MI:mice", "MI:mi". "Simple" and "Neighbor" contain more than one variant of the method. Some methods (e.g., "Below 10%") are only suitable for exploring the missingness patterns, and are not suitable to be used for producing a complete data set for analysis. Three tab labels interface to the three variants provided by "Simple", overall median, mean, and random value (Figure 1, region 7). "Neighbor" interfaces to two variants: mean of the nearest neighbors, and random nearest neighbor. The "Neighbor" methods also allow the user to change the number of neighbors. Table 1 summarizes and compares the imputation methods available in the GUI.

Univariate imputations

The simplest start involves setting the missing values to 10% below the minimum on each variable. The purpose of this is to place the missing values into the plot where they can be distinguished from the non-missing values. In a scatterplot, all missing values will lie along a vertical line on the left or a horizontal line on the bottom of the display (Figure 4(a)).This placement enables the distribution of missings to be compared with the distribution of non-missings. In the histogram, missing values will form a bar to the left of other data values. And in the parallel coordinates plot, the missing values are at the bottom of each axis.

Using the median, mean, or mode of the complete cases is a simple way to impute missing

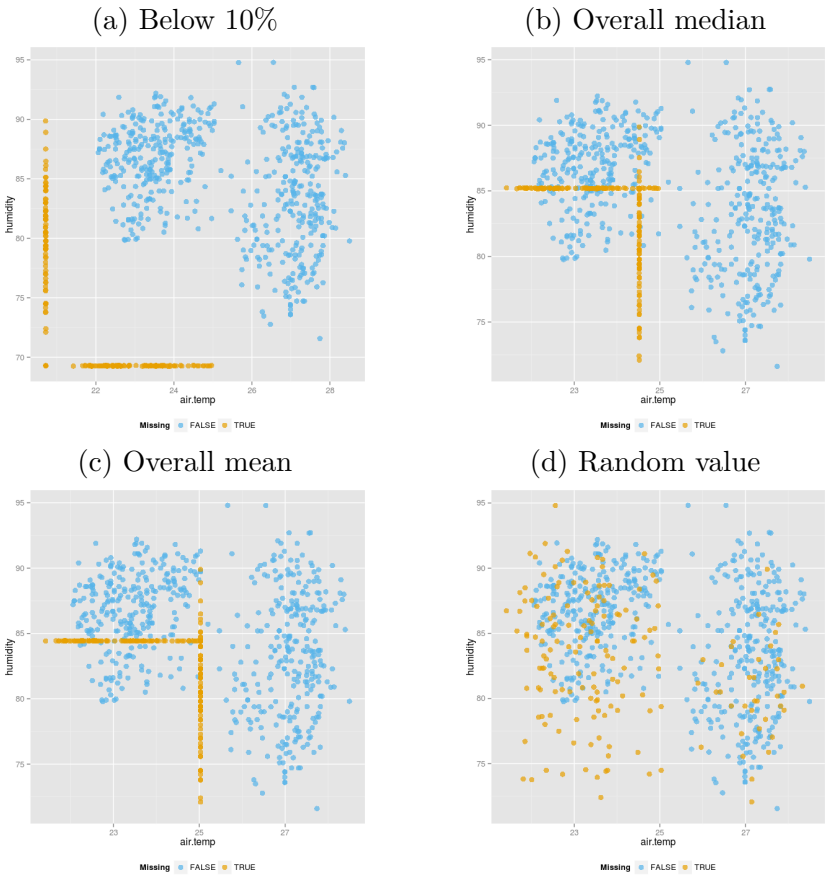


Figure 4: Four panels of scatterplots displaying the results of different univariate imputations: (a) 10% below the minimum (not strictly an imputation method, it is used for displaying missings as part of a plot of complete cases); (b) median of each variable; (c) mean of each variable; (d) random selection from the existing values.

values. The software makes some automatic choices for the user: if the user selects median but the variable type is nominal, or selects mean but the variable is categorical, then the mode is returned. In the graph, points and bars are colored according to the missing status of the case. Figures 4(b) and (c) show examples of the imputation by the median and mean for real-valued variables.

The "random value" method (Figure 4(d)) randomly selects an existing value of the variable to impute the missing. When there is more than one missing value in an observation, then values are sampled independently from each related variable.

These imputation methods operate separately on each variable. Dependencies between variables are ignored, yielding covariance and correlation estimates that are potentially very different from those of the complete cases. This could be a big problem for some analyses. These methods are not ideal from a statistical perspective. In some situations where the inadequate estimation of covariance does not affect results and conclusions they can provide a simple solution requiring only few assumptions, but in most situations they are not advised. For the application here, we are primarily concerned about providing methods for analysts to explore the missing data structure, and the plots reveal quite clearly why these univariate

Figure 5: Scatterplots for nearest neighbor imputation methods: (a) mean of the 5 nearest neighbors, (b) a random value from the 5 nearest neighbors.

imputation methods are inadequate. Figure 4 shows the "cross structure" (orange) induced on the pattern of points by mean and median imputation, and makes it quite clear that the covariance estimates for the imputed data would not well match that of the complete cases.

Neighbor imputations

The "Neighbor" methods replace a missing value with the mean of, or a random selection from, its k nearest complete neighbors (Figure 5). The distance between two observations is calculated using Euclidean distance on the standardized variables that have no missings. Figure 6 illustrates the procedure. Ties are not considered, and only the first k entries are used. This method requires at least one case in the data set to be complete, and no categorical variables can be used. (Ordinal variables are treated as integers so that distance between observations can be calculated.) If there are less than k complete cases, then all of them are used to generate the mean or a random value. If none of the cases are complete, then the mean or a random value of the entire data will be returned. By default $k = 5$, but this is up to the user's choice.

The "Neighbor" methods in **MissingDataGUI** can be seen as two special cases of hot deck imputation (Andridge and Little 2010). The neighbor mean method averages the weights on all chosen neighbors, and the random neighbor method places all the weight on one arbitrary neighbor. When $k = 1$, the methods are deterministic hot deck.

Multiple imputations

Multiple imputation, first proposed by Rubin (1978), is a method to get valid inferences by simulation. Multiple imputed data sets are generated based on the joint distribution, and serve a wide variety of analytical purposes. Functions from four R packages are utilized to implement multiple imputations in **MissingDataGUI**. Figure 7 demonstrates the results from different multiple imputations on the same data set.

Among the four packages, **norm** is quite different from the other three. The ideas behind the package were introduced by Schafer and Olsen (1998). It assumes the observations are

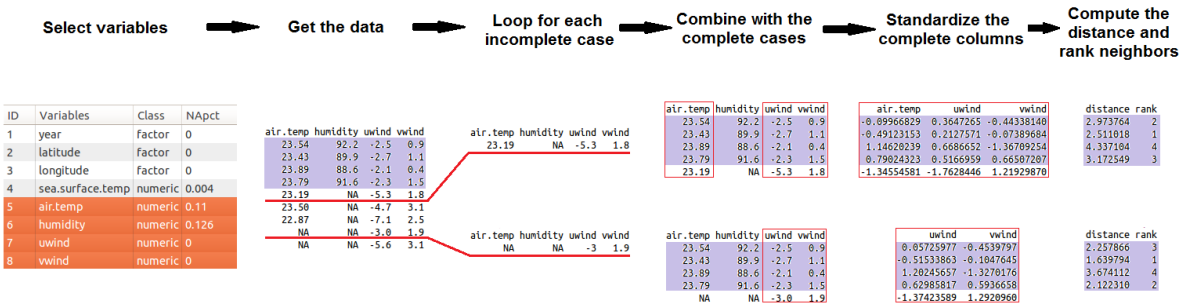


Figure 6: Illustration of the k nearest neighbors imputation method. The shaded entries are the complete observations to rank. The variables in red frames are used to compute the distance. After getting the rank of all complete observations, the first k are used as neighbors.

Algorithm steps	Hmisc	mice	mi
1. Fill in the missing	at random		
2. Specify the model	pmm/regression/normpmm	selectable model or user-specific model	
(Default model)	predictive mean matching		Bayesian generalized linear models
3. Decide the data	a bootstrap sample	the entire data set with the current imputed values	
4. Iterate imputation	in every cycle, variables with missings are imputed sequentially		
5. Stop when	achieving the max # of iterations		difference of within and between variance is small

Table 2: Comparison of the algorithm steps among three multiple imputation packages that use the chained equation approach.

sampled from a multivariate normal distribution, and uses the EM algorithm to estimate the mean and variance-covariance matrix. It utilizes a data augmentation method to update missing values and parameters in a Markov Chain and eventually converge in distribution.

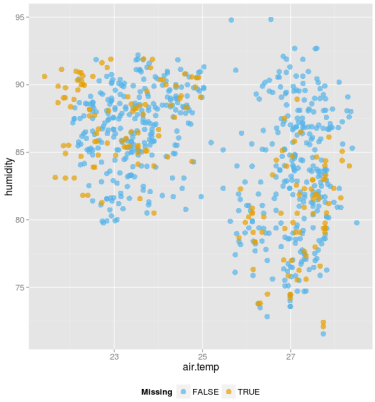
The other packages use a chained equation approach with similar steps but different settings. A comparison between the three packages is given in Table 2, based on Harrell (2015), van Buuren and Groothuis-Oudshoorn (2011), and Su *et al.* (2011). The main differences are that package **Hmisc** provides three models with flexible drawing methods around the predicted values for quantitative variables, and applies the bootstrap to obtain a sample for every iteration. Package **mi** uses a convergence criterion to stop the iteration with some allowance for special situations. In between these two is package **mice**: The models provided are more flexible than those in package **Hmisc**, but not as Bayesian as those in package **mi**.

By default, $m = 3$ chains are imputed and users can choose the number of chains. Each chain will produce a result shown in a separate graphical panel. By switching between the panels, the user can compare the results and observe discrepancies between the results. Figure 8 shows the results of four different chains produced by **mice**. Three of the four produced results where a small clump of imputed values occurred.

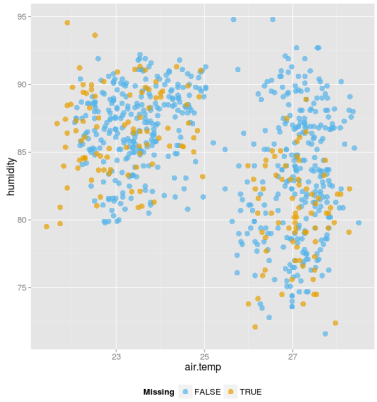
Conditional on categorical variables

When the variables of interest have bimodal or multi-modal distributions, using center statistics like the mean or median for imputation, or simulating from an overall estimate like package **norm** does, is inadequate because the center does not reflect the shape of the distri-

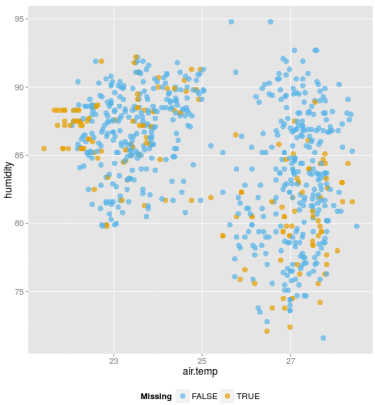
(a) **Hmisc**: predictive mean matching



(b) **norm**: multivariate normal model



(c) **mice**: chained equations



(d) **mi**: iterative regression

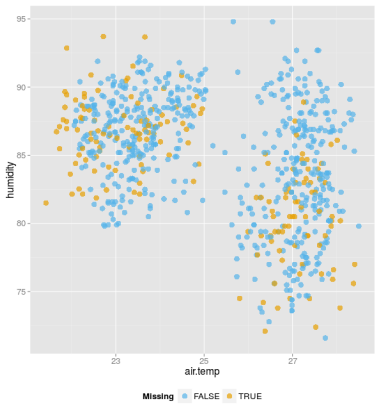


Figure 7: Scatterplots for the multiple imputations from four R packages: (a) predictive mean matching by **Hmisc**; (b) multivariate normal model by **norm**; (c) multivariate imputation using chained equations by **mice**; (d) multiple iterative regression imputation by **mi**. All the four imputations are conditioned on year.

bution properly. In many situations, the modes arise from the mixture of groups. Hence, a better imputation method is to condition by group, and then calculate the statistics.

This is available using the control "Categorical variables to condition on". All categorical variables are listed with checkboxes. The variables checked will partition the data into blocks and then the imputation method is implemented in each block of the data. However, the condition is not used when the method is "Below 10%", since the aim of "Below 10%" is simply to display the missings away from the non-missings. If the conditioning factor variable has missing values, then a "factor = `NA`" group will be generated to calculate the numeric summary or the imputed values. If the conditioning factor itself is one of the plotting variables, then a message box will emerge to ask the user to impute the missing values on the factor before other variables, and the plots are created without the condition.

The importance of conditioning in the imputation is illustrated in Figure 9. Without the condition, the distribution of imputed values does not match the distribution of complete values (Figure 9, left). Calculating separately by group provides a better result (Figure 9, right).

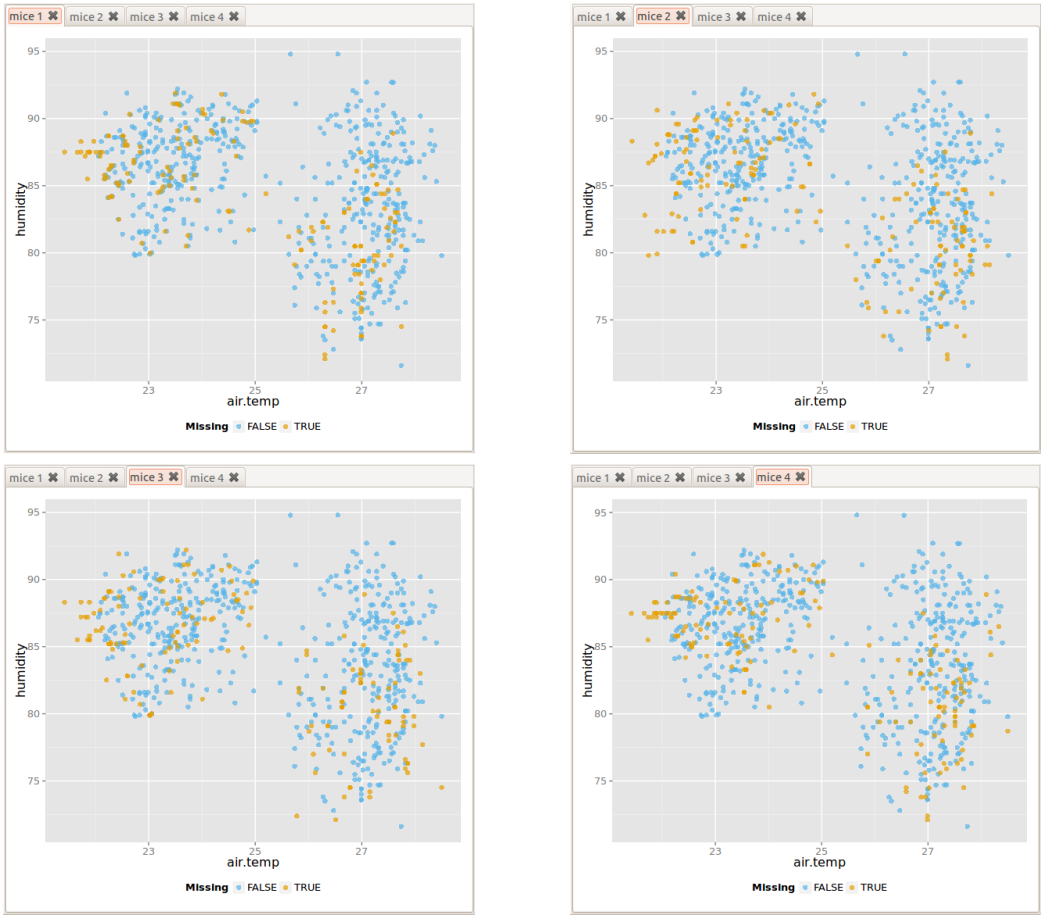


Figure 8: Results of four imputing chains by **mice**, starting with the default random seed. Users can switch the panels by clicking the tabs, or close a panel by hitting the 'x' sign. Focusing on the imputed values when air temperature is around 22 degrees, we see that the first, third and fourth chains cluster values in a small range of the y-axis, but the second chain spreads them very evenly in the y-direction.

2.4. Plot types

There are four types of graphs available in **MissingDataGUI**: histogram/barchart, spinogram/spineplot, pairwise plots, and parallel coordinates plot. Figure 10 displays all the graph types. Two color-blind friendly colors represent the observations and imputed values on any chosen variable. In Figure 10 the yellow color means that the value is originally missing in humidity.

Separate histograms (continuous variables) and barcharts (categorical variables) are shown for each of the variables selected. When the missing values and the complete values share one bar, the bar is cut into two parts, and the ratio of the two heights is equal to the ratio of missing and non-missing values in that bar.

The spinogram (continuous variable) and spineplot (categorical variable), introduced by Hummel (1996) and Theus and Lauer (1999), use width of the rectangle to represent count. The height is the same for all bars. The focus is on proportion for each group. The bars in the

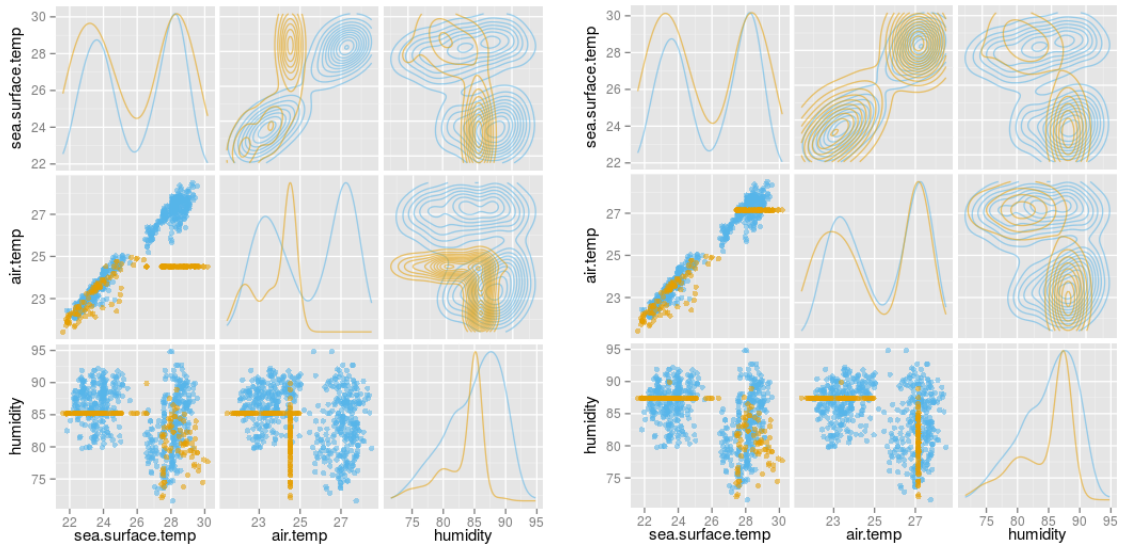


Figure 9: Effect of conditioning on imputed values. The left panel is the imputation by median without condition and the right one is conditioned on year. In the left plot we can see that the imputed values (yellow) fall between the two clusters, at the overall median. But when the imputation is conditioned on year (right plot), the imputed values are now better placed into the two clusters in the data.

spinogram or spineplot are partitioned into two colors for the missing and non-missing values.

A scatterplot matrix is used to display pairs of variables. Variable names and scales are placed on the diagonal. For the continuous variables, the pairwise scatterplots are placed in the lower triangle, and the contour plots are shown in the upper triangle. For the categorical variables, barcharts are displayed in both upper and lower triangles. Bars are colored in proportion to the missings. The combination of continuous and categorical variables is displayed as side-by-side boxplots of missing and non-missing values for each category on the upper triangle and side-by-side histograms on the lower triangle. The space available to the graphics device limits the number of variables that can be shown. The upper limit of the number of variables is set to be 5 and the lower limit is 2.

The parallel coordinates plot by Inselberg (1985) and Wegman (1990) can be used for high-dimensional data. Though many plot types, like the scatterplot or histogram, are helpful to reveal the missingness pattern, they are not convenient to display many variables simultaneously. The parallel coordinates plot can give an overview of a relatively large quantity of variables. In **MissingDataGUI**, the order of the variables can be chosen in one of two ways: the original order in the data, or by sorting the variables from the best separator to the worst of missing values by the F-statistic from ANOVA. In Figure 10, the best separating variable for the missingness of `humidity` is `humidity` itself, because the "Below 10%" method leads to a big gap between the missing and non-missing values. "Below 10%" is not an ideal method for the ordered parallel coordinates plot. However, the plot is still useful: It reveals that the missingness on `humidity` occurred in one year and one location, when `sea.surface.temp` and `air.temp` were low.

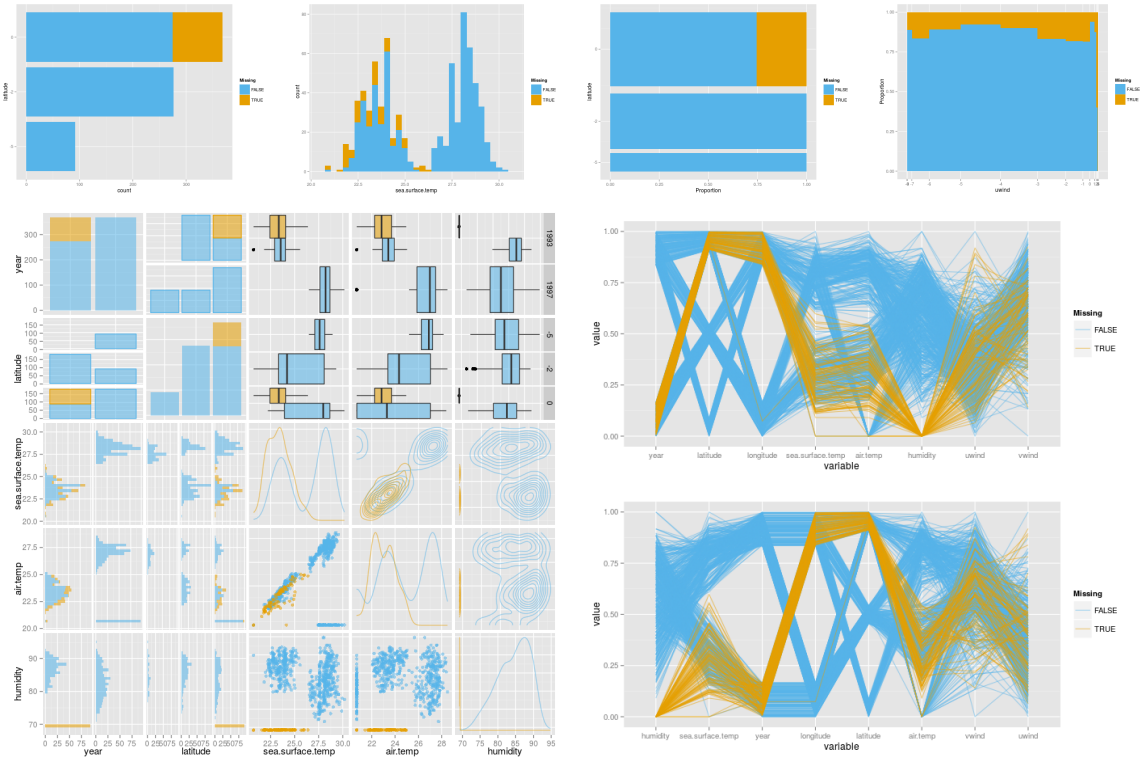


Figure 10: The four types of graphs available: (top, from left to right) barchart, histogram, spineplot, and spinogram, and (bottom, left to right) pairwise plots, and two parallel coordinates plots. The order of the variables in the parallel coordinates plot is changed from the original (upper plot) to being ordered by difference between missings and non-missings. All the plots use "Below 10%" imputation and are colored by the missingness on `humidity`.

2.5. Design issues

The missing data GUI is organized as one window with three tabs. As shown in Figure 1, the "Summary" tab includes all the important widgets: list of variables, radio for imputation methods, checkboxes for the conditional variables, the graphics device, etc. An appropriate layout makes the widgets less crowded, and is easy to maintain. The other two tabs are not as critical as the main tab, but also play important roles.

The "Help" tab shown in Figure 11 (left) has the same layout as the summary tab. The only difference is that the graphics device is replaced by the help document. The corresponding help shows up when the user moves the mouse upon a widget.

The "Settings" tab shown in Figure 11 (right) allows the user to choose options for the imputation methods in package **mice**, as well as other settings for multiple imputation, neighbor selection, and the display of a parallel coordinates plot. To change the imputation models, users can double-click a variable in the left table, and select any method provided in the pop-up window. The choices vary depending on the type of the variable.

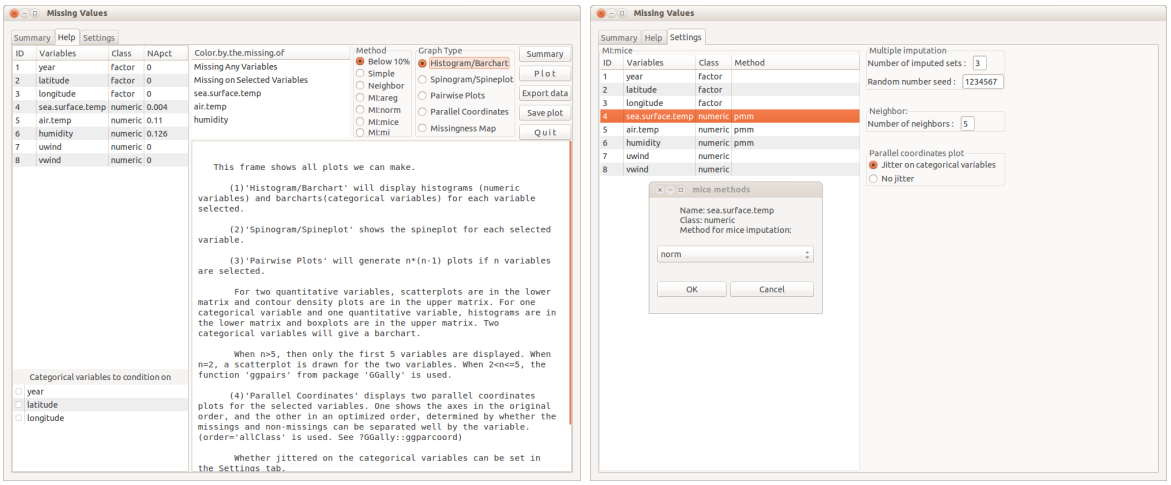


Figure 11: Subsidiary GUI tabs: (left) "Help" tab, (right) "Settings" tab. The layout of the "Help" tab mirrors the actual functional GUI. Mousing over any part of it or clicking the radio/checkbox items will have text explanations pop up in the summary region. All the widgets have a detailed introduction. The "Settings" tab is used to make changes to the variable types and algorithm options. Users can modify the number of imputed sets to generate, the random number seed, the number of neighbors, and the jitter setting for the parallel coordinates plot.

2.6. Data input and output

Data can be entered as either a data frame or a comma separated file (CSV). The preferred approach is to read an existing data frame in R because the type of variables (e.g., factor, numeric) are preserved. `MissingDataGUI(data)` is used to achieve this.

If reading from a CSV file, `MissingDataGUI()` will trigger the data import GUI (Figure 12), from which to select a file. The "Open" button is for choosing files and the "Watch Missing Values" buttons will launch the missing data GUI. The file format must be CSV, and only one data set can be imported into the missing data GUI at a time, although several files can be opened in the data import GUI.

Once values are imputed, and a complete data set created, it can be saved using the "Export data" button (Figure 13). Only the selected variables will be imputed, but users could choose whether to export the selected columns or all the columns (with `NA`'s existing in the unselected variables). The shadow matrix is exported by default, so that analysts can always track back to find the locations of the real missings. Data can be saved in three ways: a CSV file, an rda file, or a data frame. The multiple imputed sets from several chains will be saved as a list in rda format or data frame, or in separate CSV files.

The exported data with its shadow matrix can be loaded back into the GUI, which implies the imputed data from other imputation methods (not provided by the missing data GUI) can also be imported. Users only need to provide a shadow matrix which indicates the locations of missings. In other words, the imported structure should be a data frame or a CSV file with the first n columns being the imputed data and the next n columns being the shadow matrix.

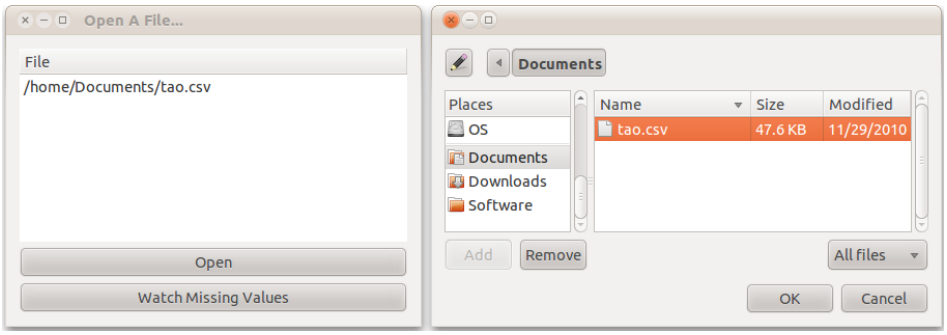


Figure 12: The data import GUI, with file selector, which pops up upon clicking the "open" button. More than one file could be listed in the GUI, but only one data set is allowed to be active in the missing data GUI. The first file is automatically imported if none of the data sets are chosen when the "Watching Missing Values" button is hit.

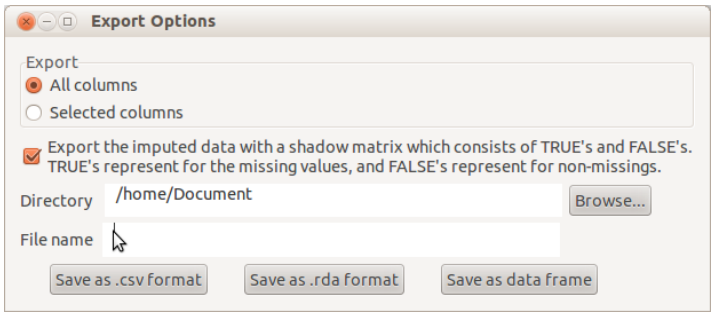


Figure 13: The data export GUI. By default, all columns are exported with a shadow matrix. The current working directory is set to be the location for the exported files. Three exporting formats are provided.

2.7. Additional features of the GUI

Change the variable attributes. Double-clicking on any variables in the top left table of the "Summary" tab will open an attribute window, as displayed in Figure 14. Users could edit the variable name, or assign another class to the variable. When the class of a variable is switched from numeric/integer to character/factor/ordinal, the variable will be automatically loaded into the checkbox group as a potential conditioning variable.

Search a variable by typing text. The variable table, the conditioning checkboxes, and the color-by-variable selector allow text entry to find a variable. This feature is especially useful when there are many variables in the data set.

Save the plots. Plots can be saved to PNG files with the "Save plot" button. The imputation method and plot type will be auto-completed in the file name.

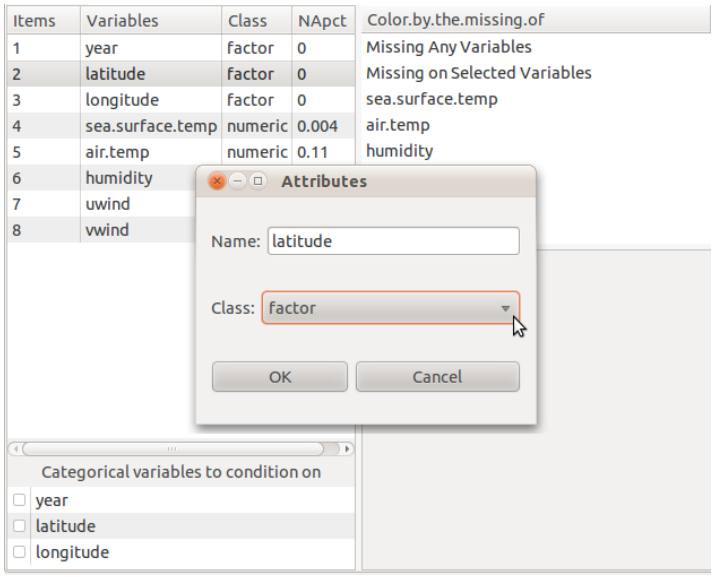


Figure 14: The attributes list for variable selection is interactive. The name can be edited, and the class could be changed to one of the five classes: integer, numeric, character, factor, or ordinal (factor). When a numeric variable is changed to a categorical variable, the widget for conditions will be updated.

3. Example

3.1. Data

Two data sets are provided with the package: `tao`, which is used as the example in this section, and `brfss`. The `brfss` data is a subset of the 2009 survey from the Behavioral Risk Factor Surveillance System, an ongoing data collection program designed to measure behavioral risk factors for the US adult population (18 years of age or older). The website for this program is http://www.cdc.gov/BRFSS/.

The data set `tao` is from the Tropical Atmosphere Ocean project (TAO; McPhaden 2011). The TAO array consists of approximately 70 moorings in the Tropical Pacific Ocean, telemetering oceanographic and meteorological data to shore in real-time via the Argos satellite system. A subset of the data from 6 moorings in 1993 and 1997 is used for the example. The data consists of 8 variables (year, latitude, longitude, sea surface temperature, air temperature, humidity, `uwind` and `vwind`) and 736 observations. The numeric summary of the 8 variables is shown in Figure 2. This subset is provided by Cook and Swayne (2007). We can open the GUI using the following commands:

```
R> library("MissingDataGUI")
R> MissingDataGUI(tao)
```

3.2. Exploring missings

Three of the 8 variables have missing values. First, we have a look at the distribution of missings on these variables. Figure 15 (left) shows the pairwise plots of three variables

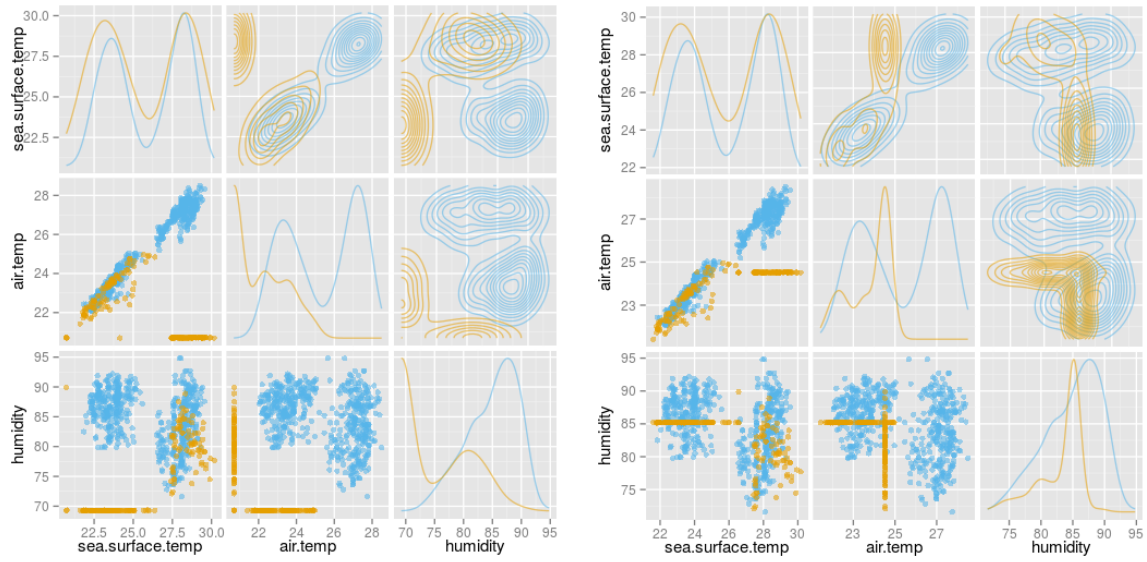


Figure 15: (Left) Exploring the missingness (yellow) on humidity, sea and air temperature. Missings on humidity (the bottom line of the third row) occur at the lower temperature values, suggesting a dependence relationship. Missing values are not missing completely at random. (Right) Imputation using the medians. Median imputation introduces a cross structure to the point scatter, and the imputed values do not match the data well.

(`sea.surface.temp`, `air.temp`, and `humidity`) with missing values on any of the three variables colored in yellow, and shown as 10% below the minimum data value. Cases which are missing on humidity (string of points at bottom of bottom row of plots) have low values of sea and air temperature. This suggests the dependence between humidity missingness and the temperature variables. Imputation methods that incorporate this dependence may be preferable.

Figure 15 (right) shows the data imputed with median values. This imputation imposes a cross structure on the data, which does not match the shape of the complete cases. This would not be a recommended method for creating a complete data set.

Figure 16 (left) shows the data imputed with median values conditional on year. This better matches the distribution of complete cases, although the imputed values still form bands in the scatterplot. This might be a problem because the variance estimation will be affected.

For this data set, better ways to impute the data would take the strong association between the variables into account. This suggests that neighbor or multiple imputation might be more desirable imputation methods. Figure 16 (right) shows the results for MI:areg, the regression-based imputation, conditional on year. The imputed values match the distribution of complete cases reasonably well. There are a few slight concerns: Some of the imputed values have lower air temperature values than any of the complete cases, the spread of the imputed values is a little greater than the complete cases. But overall, this is probably as good as it is going to get with imputing the missings for this data set. It would be reasonable to export the imputed data for further analysis at this point.

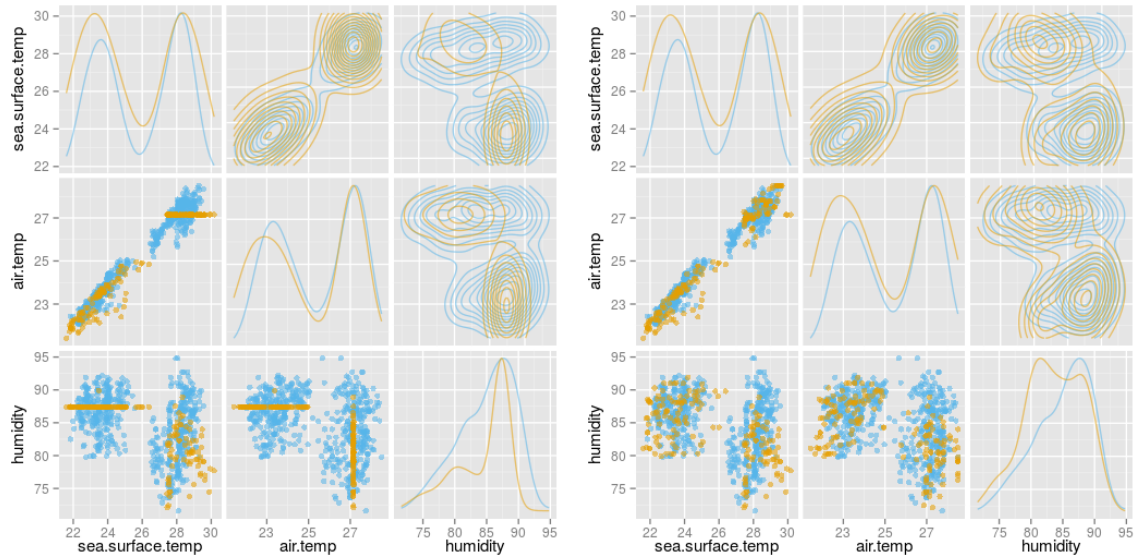


Figure 16: (Left) Imputation using the median, conditional on year. Imputed values better match the complete cases, with the exception of the banding due to a fixed median value. (Right) Imputation using the multiple imputation MI:areg conditional on year. The distribution of imputed values is fairly close to the distribution of complete cases.

3.3. Check assumptions

In the statistical imputation literature, there are three types of missing data mechanisms: MCAR (missing completely at random), MAR (missing at random), and MNAR (missing not at random). Many imputation methods, including multiple imputation, assume MCAR or MAR. However, MCAR is the most difficult mechanism to substantiate, because it requires that missingness be independent of the observed or other missing values. MAR is less strict, because it allows for missings to be dependent on observed values, but it still expects the missingness to be independent from other missing values. To assess the adequacy of the assumptions, an important step is to review the data generation process. Beyond this we follow a process of elimination. The missingness pattern is believed MCAR unless there is strong evidence against it. If MCAR is negated, then it is believed that the pattern is MAR, unless there are strong indications in the data generation process that render MAR implausible. If not MAR, then MNAR has to be assumed.

As an example, let us check whether the two incomplete variables (`air.temp` and `humidity`) in the data set `tao` follows MCAR or MAR. Figure 17 gives two parallel coordinates plots, colored by the missingness on `air.temp` (left) and `humidity` (right). The yellow lines are the cases that are missing only on `air.temp` (or `humidity`). The missing values are represented by the "Below 10%" method. Most of the missings on `air.temp` occurred in one year and one location, with higher `sea.surface.temp`, higher `uwind`, and lower `humidity`. Most of the missings on `humidity` happened in the same location as the missings on `air.temp` but in another year, with lower `sea.surface.temp` and lower `air.temp`. This is strong evidence against MCAR.

Rejecting MAR is very difficult generally, even when an obvious difference between missings and non-missings can be seen in the plots, because the real values of the missings are un-

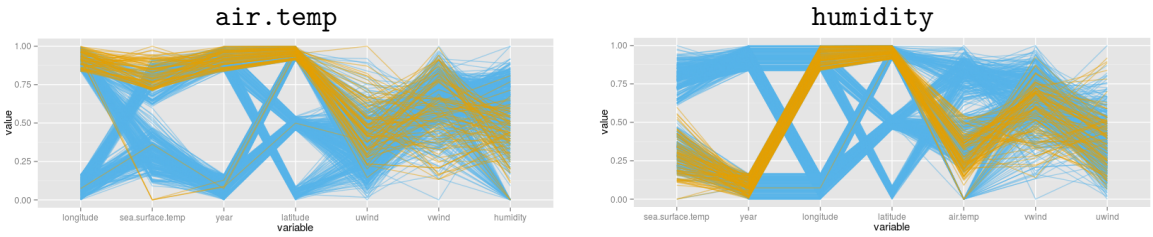


Figure 17: Parallel coordinates plots colored by whether missing on `air.temp` (left) or `humidity` (right). The variables are sorted by the F-statistic of ANOVA, i.e., the difference between the missing data and the observed data on a standard scale. Any missing values in the present variables are imputed by the "Below 10%" method. Obviously the missingness on `air.temp` and `humidity` associates with other variables like year and location, so the MCAR assumption on these two variables is violated.

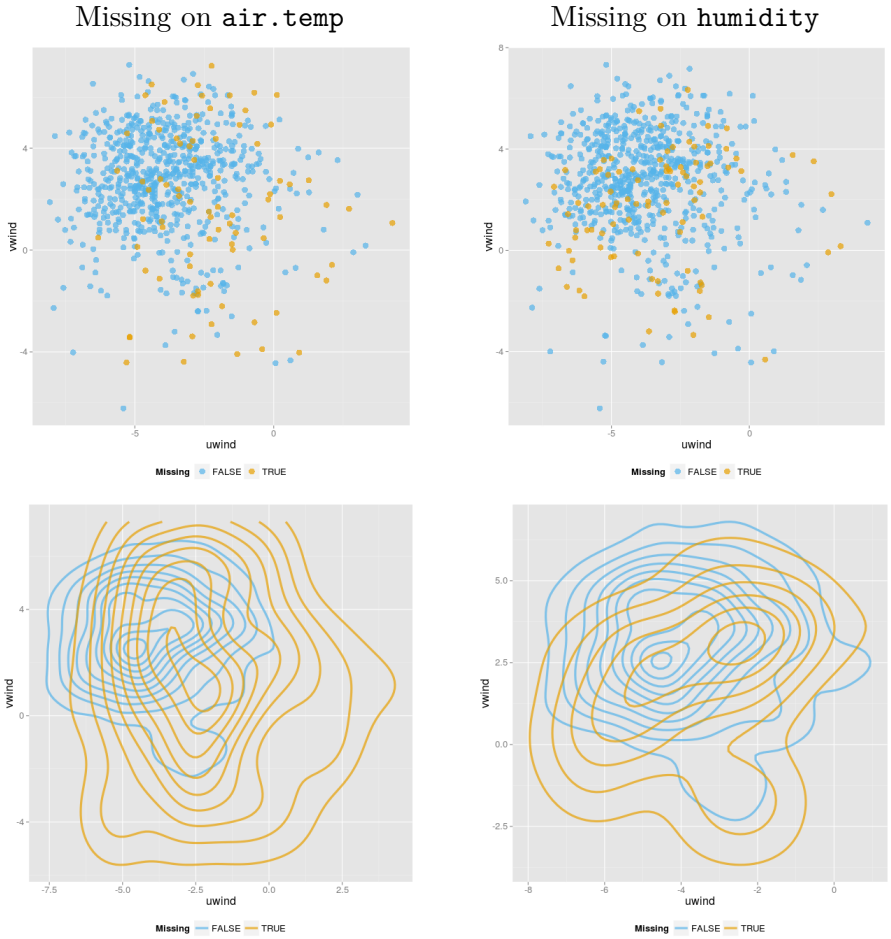


Figure 18: Scatterplots (top row) and contour plots (bottom row) of `uwind` and `vwind`, two complete variables, colored by whether missing on `air.temp` (left column) or `humidity` (right column). The missings of `air.temp` is averagely higher on `uwind` and has a larger variation, than the non-missings of `air.temp`. In reverse, the joint distribution of `uwind` and `vwind` on the missings of `humidity` is closer to that of the non-missings.

known. For example, in Figure 18, the distributions of `uwind` and `vwind` conditioned on the missingness of `air.temp` are different, since the missings on `air.temp` are higher in `uwind` and more scattered in `vwind`. The reasoning is complicated, but we cannot reject MAR. It is possible that, conditional on `uwind` and `vwind`, the distribution of true `air.temp` values of the missings is the same as that of the observed values. Thus, `uwind` and `vwind` remove any dependence of missing status for the `air.temp` variable. Generally, it is not possible to establish MNAR without actually knowing the true values of the missings. However, for this data set, the plots suggest that the imputation of `air.temp` should involve `uwind` and `vwind`.

4. Summary

The **MissingDataGUI** package makes it possible to explore patterns of missing values in data and the impact of various imputation methods on the distribution of values in the data. Future work would add interaction to the plots so that it is possible to brush points to more completely explore the missing data structure, as can be done in **GGobi** and **MANET**.

Software

MissingDataGUI is written in R 3.1.1 (R Core Team 2014) and based on the package **gWidgets** (Verzani, Urbanek, Grosjean, and Lawrence 2014) with the toolkit **RGtk2** (Lawrence and Temple Lang 2010). On different platforms (Windows, Linux, Mac) the appearance of the GUI will differ slightly, but the functionality will be the same.

The histogram/barchart, spinogram/spineplot, and missingness map are generated using **ggplot2** (Wickham 2009). The scatterplot matrix and parallel coordinates plot are produced by package **GGally** (Schloerke, Crowley, Cook, Hofmann, Wickham, Briatte, Marbach, and Thoen 2014).

Acknowledgments

This work was partially supported by an unrestricted fellowship from Novartis, and National Science Research grant DMS0706949.

References

Andridge R, Little R (2010). "A Review of Hot Deck Imputation for Survey Non-Response." *International Statistical Review*, **78**(1), 40–64. doi:10.1111/j.1751-5823.2010.00103.x.

Brix P (2012). ***miP**: Multiple Imputation Plots*. R package version 1.1, URL http://CRAN.R-project.org/package=miP.

Cheng X, Cook D, Hofmann H (2015). ***MissingDataGUI**: A GUI for Missing Data Exploration*. R package version 0.2-4, URL http://CRAN.R-project.org/package=MissingDataGUI.

Cook D, Swayne D (2007). *Interactive and Dynamic Graphics for Data Analysis: With R and **GGobi***. Springer-Verlag, New York, NY.

Goodrich B (2015). ***migui**: Graphical User Interface to the **mi** Package*. R package version 1.1, URL http://CRAN.R-project.org/package=migui.

Harrell F (2015). ***Hmisc**: Harrell Miscellaneous*. R package version 3.16-0, URL http://CRAN.R-project.org/package=Hmisc.

Honaker J, King G, Blackwell M (2011). "**Amelia II**: A Program for Missing Data." *Journal of Statistical Software*, **45**(7), 1–47. doi:10.18637/jss.v045.i07.

Hummel J (1996). "Linked Bar Charts: Analysing Categorical Data Graphically." *Computational Statistics*, **11**(1), 23–34.

Inselberg A (1985). "The Plane with Parallel Coordinates." *The Visual Computer*, **1**(2), 69–91. doi:10.1007/bf01898350.

Jaeger M (2006). "On Testing the Missing at Random Assumption." In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 671–678. Springer-Verlag. 17th European Conference on Machine Learning Berlin, Germany, September 18–22, 2006 Proceedings.

Lawrence M, Temple Lang D (2010). "**RGtk2**: A Graphical User Interface Toolkit for R." *Journal of Statistical Software*, **37**(8), 1–52. doi:10.18637/jss.v037.i08.

Little R (1988). "A Test of Missing Completely at Random for Multivariate Data with Missing Values." *Journal of the American Statistical Association*, **83**(404), 1198–1202. doi:10.1080/01621459.1988.10478722.

McPhaden M (2011). "Tropical Atmosphere Ocean Project." http://www.pmel.noaa.gov/tao/index.shtml.

Novo A, Schafer J (2013). ***norm**: Analysis of Multivariate Normal Datasets with Missing Values*. R package version 1.0-9.5, URL http://CRAN.R-project.org/package=norm.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rubin D (1978). "Multiple Imputations in Sample Surveys – A Phenomenological Bayesian Approach to Nonresponse." In *Proceedings of the American Statistical Association, Section on Survey Research Methods*, pp. 20–34.

Schafer J, Olsen M (1998). "Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective." *Multivariate Behavioral Research*, **33**(4), 545–571. doi:10.1207/s15327906mbr3304_5.

Schloerke B, Crowley J, Cook D, Hofmann H, Wickham H, Briatte F, Marbach M, Thoen E (2014). ***GGally**: Extension to **ggplot2***. R package version 0.5.0, URL http://CRAN.R-project.org/package=GGally.

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

Su YS, Gelman A, Hill J, Yajima M (2011). "Multiple Imputation with Diagnostics (**mi**) in R: Opening Windows into the Black Box." *Journal of Statistical Software*, **45**(2), 1–31. doi:10.18637/jss.v045.i02.

Swayne D, Buja A (1998). "Missing Data in Interactive High-Dimensional Data Visualization." *Computational Statistics*, **13**(1), 15–26. doi:10.2307/1390754.

Swayne D, Cook D, Buja A (1998). "**XGobi**: Interactive Dynamic Data Visualization in the X Window System." *Journal of Computational and Graphical Statistics*, **7**(1), 113–130. doi:10.1080/10618600.1998.10474764.

Swayne D, Temple Lang D, Buja A, Cook D (2003). "**GGobi**: Evolving from **XGobi** into an Extensible Framework for Interactive Data Visualization." *Computational Statistics & Data Analysis*, **43**, 423–444. doi:10.1016/s0167-9473(02)00286-4.

Templ M, Alfons A, Kowarik A, Prantner B (2015). ***VIM**: Visualization and Imputation of Missing Values*. R package version 4.3.0, URL http://CRAN.R-project.org/package=VIM.

Templ M, Filzmoser P (2008). "Visualization of Missing Values Using the R Package **VIM**." *Research Report CS-2008-1*, Department of Statistics and Probability Therory, Vienna University of Technology.

Theus M, Lauer S (1999). "Visualizing Loglinear Models." *Journal of Computational and Graphical Statistics*, **8**(3), 396–412. doi:10.1080/10618600.1999.10474821.

Unwin A, Hawkins G, Hofmann H, Siegl B (1996). "Interactive Graphics for Data Sets With Missing Values: **MANET**." *Journal of Computational and Graphical Statistics*, **5**(2), 113–122. doi:10.1080/10618600.1996.10474700.

van Buuren S, Groothuis-Oudshoorn K (2011). "**mice**: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, **45**(3), 1–67. doi:10.18637/jss.v045.i03.

Verzani J, Urbanek S, Grosjean P, Lawrence M (2014). ***gWidgets**: **gWidgets** API for Building Toolkit-Independent, Interactive GUIs*. R package version 0.0-54, URL http://CRAN.R-project.org/package=gWidgets.

Wegman E (1990). "Hyperdimensional Data Analysis Using Parallel Coordinates." *Journal of the American Statistical Association*, **85**(411), 664–675. doi:10.1080/01621459.1990.10474926.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York, NY. URL http://had.co.nz/ggplot2/book.

Affiliation:

Xiaoyue Cheng
Department of Mathematics
University of Nebraska at Omaha
225 Durham Science Center
Omaha, NE 68182, United States of America
E-mail: xycheng@unomaha.edu
URL: http://chxy.github.io/

Dianne Cook
Department of Econometrics and Business Statistics
Monash University
E869 Menzies Building
20 Chancellors Walk
Clayton, VIC 3800, Australia
E-mail: dicook@monash.edu
URL: http://dicook.github.io/

Heike Hofmann
Department of Statistics and Statistical Laboratory
Iowa State University
2413 Snedecor Hall
Ames, IA, 50011, United States of America
E-mail: hofmann@iastate.edu
URL: http://hofmann.public.iastate.edu/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
December 2015, Volume 68, Issue 6 *Submitted:* 2013-02-14
doi:10.18637/jss.v068.i06 *Accepted:* 2015-01-12