Diplomarbeit

zum Thema

DAS+R: Subset Operations, Usage and Graphical Data Representations

ausgeführt am
Institut für Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter der Anleitung von
O.Univ.-Prof. Dipl.-Ing. Dr.techn. Rudolf Dutter

durch

Zainzinger Andreas
Matrikelnr.: 0325310
Kronbergstr. 24, 3683 Yspertal

Wien, am 10. 5. 2010 Andreas Zainzinger

Acknowledgments

First of all, I would like to thank Rudolf Dutter, my supervisor for all the effort and help he gave to me, for the opportunity to collaborate on a software package and all the discussions about the development of DASplusR.

Further I am very thankful to my parents for all the support during the years.

Many thanks also to Reka Horvath and Roman Schneider for the good time we could spend together, and especially to Roman for all his help during my studies.

Andreas Zainzinger

Contents

List of Figures

Chapter 1

Introduction

Data sets in applied statistical analysis often contain a large number of samples. Instead of the whole data set just a subset of the samples would be used for data analysis.

This thesis is about generating and manipulating subsets and their usage in diagrams for the software package DASplusR which is a Graphical User Interface (GUI) for the free statistic software R. This software and all its packages can be downloaded from http://cran.r-project.org. Since the DASplusR software is under development the latest version of the package can be downloaded from the homepage of Prof. Rudolf Dutter (http://www.statistik.tuwien.ac.at/public/dutt/) instead of the cran homepage. An introduction about the DASplusR and its features can be found in the book Reimann et al. (2008).

In Chapter 2 a description about the different ways of how to generate subsets is given. Further this chapter shows the different methods for manipulating subsets in DASplusR.

Chapter 3 gives an overview about applications. A subset can be used to make a new data set. Therefore just the samples of the original data set that are included in the subset are taken. Another application is the usage in diagrams such as boxplot, ternary plot, xy-plot, density plot and histogram.

Appendix A contains all the help files which are written for the programs of this thesis. The code that was used for generating the diagrams shown in this thesis and example code for the generation of subsets is listed in Appendix B.

Apart from the functions listed in this thesis I supported Prof. Dutter by some other functions for DASplusR for example implementing scrollbars, importing data, selecting variables, etc. Therefore I would thank him again for giving me the opportunity to collaborate on such an interesting project.

Chapter 2

Generating and Editing Subsets

Data sets in applied statistical analysis often contain a large number of samples. Instead of the whole data set just a subset of the samples would be used for data analysis. To receive such smaller data sets in DASplusR we use subsets. A subset is described by a set of indices. These are the row numbers of the data matrix which contain the desired samples.

2.1 Generating New Subsets

In the main window of DASplusR a new subset can be generated by clicking

`Edit` → `Subsets` → `Make`.

as shown in Figure 2.1.

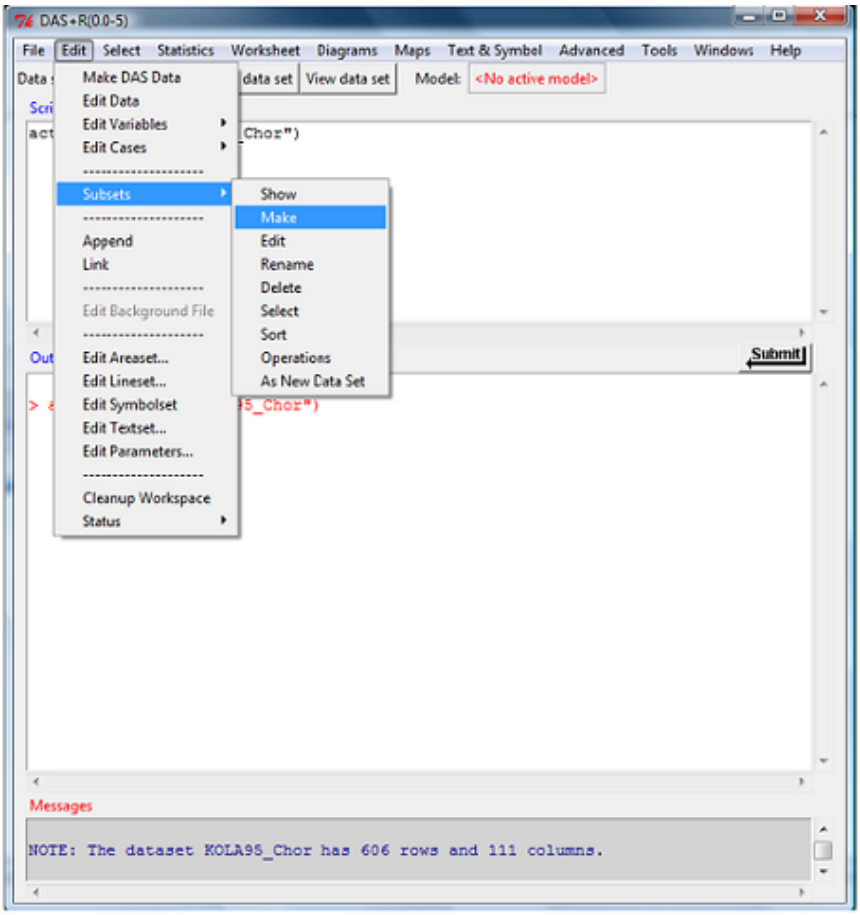


Figure 2.1: DASplusR main window

Figure 2.2 shows the main window for generating (making) new subsets. The "Subset list" displays all existing subsets. In the field "New subset", the name for the new subset has to be entered. If the name already exists the question arises whether an existing name should be overwritten. In "Description" an optional text describing the subset can be entered. Below these fields there are different options for defining a subset, namely

1. 0/1 vector
2. T/F vector
3. Row names
4. Row indices

5. Graphical
6. Expression.

Additionally, a subset can be generated by operations as described in Section 2.2.

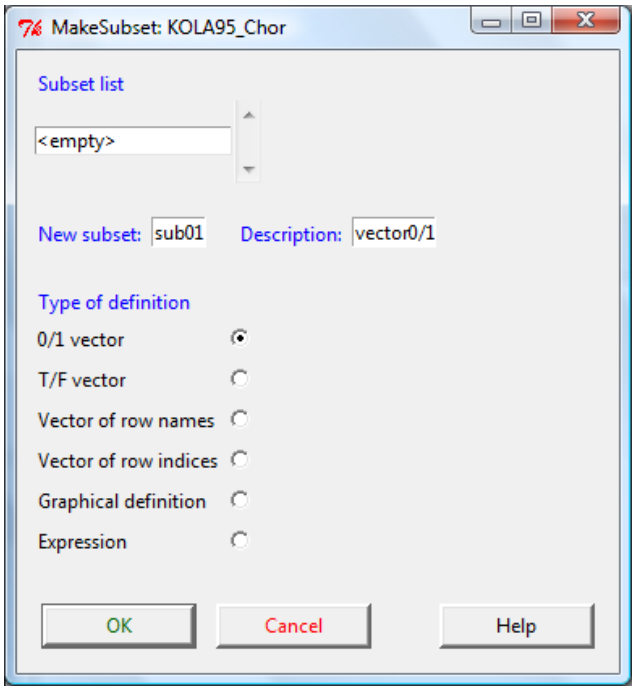
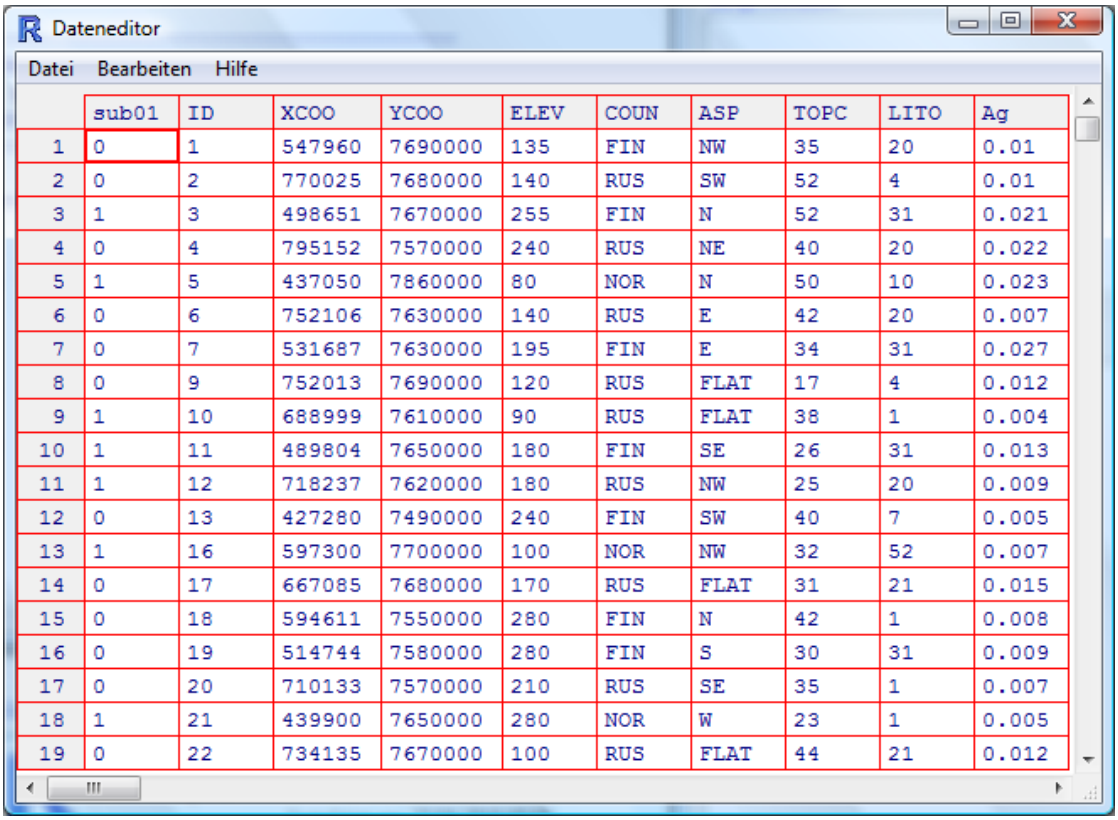


Figure 2.2: Main window for generating subsets

In the following we will comment on these possibilities in more detail.

2.1.1 0/1 Vector

The subset is defined by a vector (the first column of the matrix in Figure 2.3) containing 0's and 1's where the 1's represent the selected samples. Other numbers are treated as 0's. After the selection is finished click the "X" to save the changes and close the editor window.

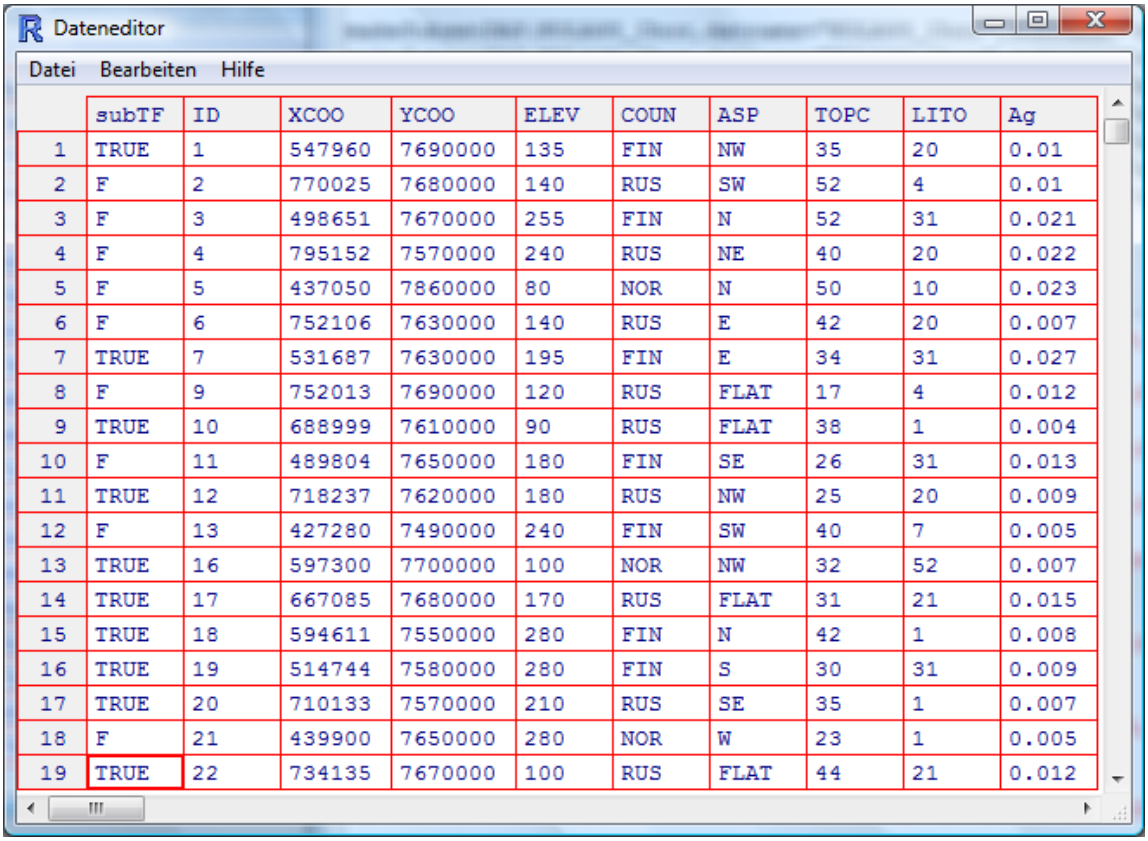


	sub01	ID	XCOO	YCOO	ELEV	COUN	ASP	TOPC	LITO	Ag
1	0	1	547960	7690000	135	FIN	NW	35	20	0.01
2	0	2	770025	7680000	140	RUS	SW	52	4	0.01
3	1	3	498651	7670000	255	FIN	N	52	31	0.021
4	0	4	795152	7570000	240	RUS	NE	40	20	0.022
5	1	5	437050	7860000	80	NOR	N	50	10	0.023
6	0	6	752106	7630000	140	RUS	E	42	20	0.007
7	0	7	531687	7630000	195	FIN	E	34	31	0.027
8	0	9	752013	7690000	120	RUS	FLAT	17	4	0.012
9	1	10	688999	7610000	90	RUS	FLAT	38	1	0.004
10	1	11	489804	7650000	180	FIN	SE	26	31	0.013
11	1	12	718237	7620000	180	RUS	NW	25	20	0.009
12	0	13	427280	7490000	240	FIN	SW	40	7	0.005
13	1	16	597300	7700000	100	NOR	NW	32	52	0.007
14	0	17	667085	7680000	170	RUS	FLAT	31	21	0.015
15	0	18	594611	7550000	280	FIN	N	42	1	0.008
16	0	19	514744	7580000	280	FIN	S	30	31	0.009
17	0	20	710133	7570000	210	RUS	SE	35	1	0.007
18	1	21	439900	7650000	280	NOR	W	23	1	0.005
19	0	22	734135	7670000	100	RUS	FLAT	44	21	0.012

Figure 2.3: Definition of a subset by a 0/1 vector

2.1.2 T/F Vector

In this case, the subset is defined by a logical vector where "TRUE" marks a selected sample. Instead of "TRUE" also "T", "True" and "true" are accepted. All other forms are treated as FALSE. "X" again closes the editor and saves the changes.

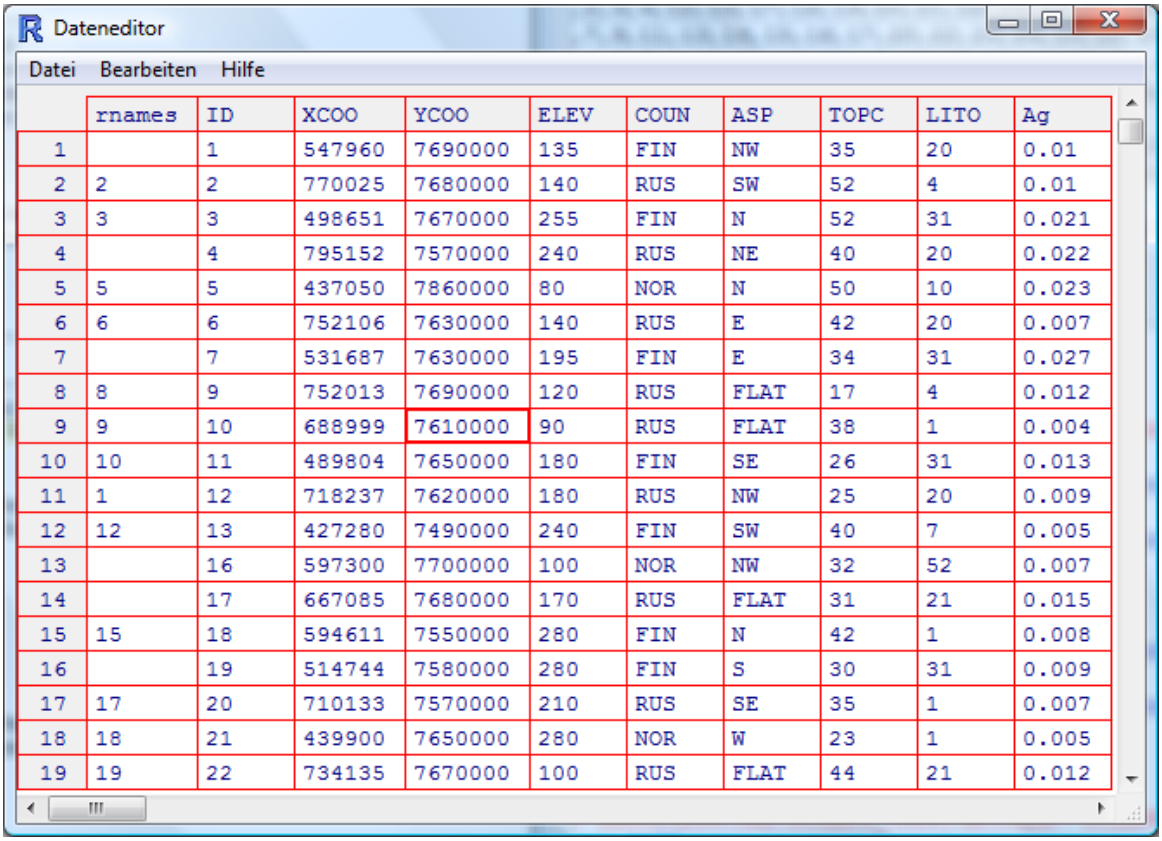


	subTF	ID	XCOO	YCOO	ELEV	COUN	ASP	TOPC	LITO	Ag
1	TRUE	1	547960	7690000	135	FIN	NW	35	20	0.01
2	F	2	770025	7680000	140	RUS	SW	52	4	0.01
3	F	3	498651	7670000	255	FIN	N	52	31	0.021
4	F	4	795152	7570000	240	RUS	NE	40	20	0.022
5	F	5	437050	7860000	80	NOR	N	50	10	0.023
6	F	6	752106	7630000	140	RUS	E	42	20	0.007
7	TRUE	7	531687	7630000	195	FIN	E	34	31	0.027
8	F	9	752013	7690000	120	RUS	FLAT	17	4	0.012
9	TRUE	10	688999	7610000	90	RUS	FLAT	38	1	0.004
10	F	11	489804	7650000	180	FIN	SE	26	31	0.013
11	TRUE	12	718237	7620000	180	RUS	NW	25	20	0.009
12	F	13	427280	7490000	240	FIN	SW	40	7	0.005
13	TRUE	16	597300	7700000	100	NOR	NW	32	52	0.007
14	TRUE	17	667085	7680000	170	RUS	FLAT	31	21	0.015
15	TRUE	18	594611	7550000	280	FIN	N	42	1	0.008
16	TRUE	19	514744	7580000	280	FIN	S	30	31	0.009
17	TRUE	20	710133	7570000	210	RUS	SE	35	1	0.007
18	F	21	439900	7650000	280	NOR	W	23	1	0.005
19	TRUE	22	734135	7670000	100	RUS	FLAT	44	21	0.012

Figure 2.4: Definition of a subset by a T/F vector

2.1.3 Row Names

The next possibility is to define the subset by a vector containing the sample names (row names) of the data set. If the chosen data set doesn't contain specified sample names then row numbers are taken as sample names. A sample can be deselected by deleting the name. Deleting just a part of the row name gives the same result as deleting the whole name. If a sample name is entered in another row it is treated like deselected as we can see in Figure 2.5. In row 11, 1 is entered as sample name instead of 11, that means that this sample is also not in the subset.

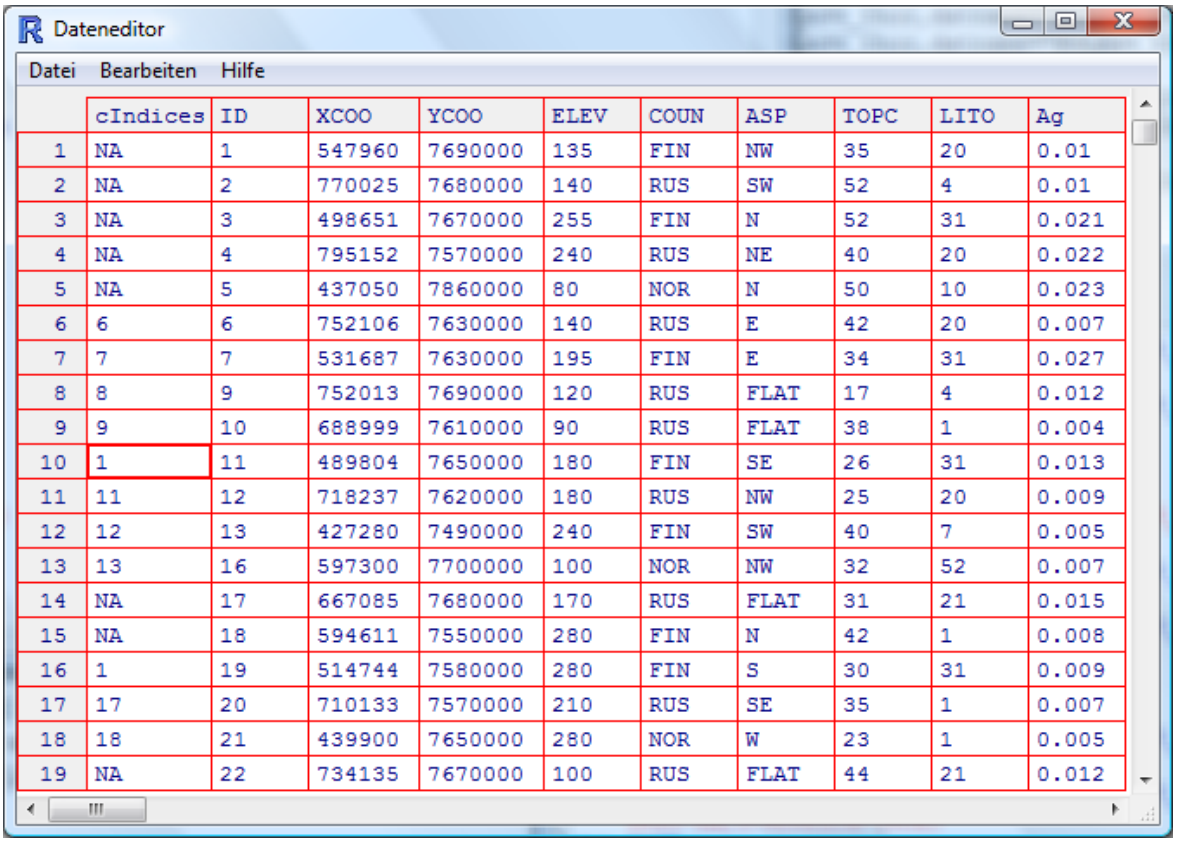


	rnames	ID	XCOO	YCOO	ELEV	COUN	ASP	TOPC	LITO	Ag
1		1	547960	7690000	135	FIN	NW	35	20	0.01
2	2	2	770025	7680000	140	RUS	SW	52	4	0.01
3	3	3	498651	7670000	255	FIN	N	52	31	0.021
4		4	795152	7570000	240	RUS	NE	40	20	0.022
5	5	5	437050	7860000	80	NOR	N	50	10	0.023
6	6	6	752106	7630000	140	RUS	E	42	20	0.007
7		7	531687	7630000	195	FIN	E	34	31	0.027
8	8	9	752013	7690000	120	RUS	FLAT	17	4	0.012
9	9	10	688999	7610000	90	RUS	FLAT	38	1	0.004
10	10	11	489804	7650000	180	FIN	SE	26	31	0.013
11	1	12	718237	7620000	180	RUS	NW	25	20	0.009
12	12	13	427280	7490000	240	FIN	SW	40	7	0.005
13		16	597300	7700000	100	NOR	NW	32	52	0.007
14		17	667085	7680000	170	RUS	FLAT	31	21	0.015
15	15	18	594611	7550000	280	FIN	N	42	1	0.008
16		19	514744	7580000	280	FIN	S	30	31	0.009
17	17	20	710133	7570000	210	RUS	SE	35	1	0.007
18	18	21	439900	7650000	280	NOR	W	23	1	0.005
19	19	22	734135	7670000	100	RUS	FLAT	44	21	0.012

Figure 2.5: Definition of a subset by a vector of the row names

2.1.4 Row Indices

The subset is generated from a vector containing the row indices of a data set. The row indices are the numbers of the samples. The vector contains all numbers of the selected samples. If an index is deleted, the system fills the empty field with NA. Different to editing by the row names is that, if a row index is entered in another row, the entered index is also treated as selected. In row 10 in the figure below the index 1 is entered. That means that although the index of the first row is deleted, the subset will contain the first sample but the 10th is deselected. In this figure row 16 also contains the index 1. Because 1 is already selected due to the index in row 10 it is treated as the 16th row were deselected.



	cIndices	ID	XCOO	YCOO	ELEV	COUN	ASP	TOPC	LITO	Ag
1	NA	1	547960	7690000	135	FIN	NW	35	20	0.01
2	NA	2	770025	7680000	140	RUS	SW	52	4	0.01
3	NA	3	498651	7670000	255	FIN	N	52	31	0.021
4	NA	4	795152	7570000	240	RUS	NE	40	20	0.022
5	NA	5	437050	7860000	80	NOR	N	50	10	0.023
6	6	6	752106	7630000	140	RUS	E	42	20	0.007
7	7	7	531687	7630000	195	FIN	E	34	31	0.027
8	8	9	752013	7690000	120	RUS	FLAT	17	4	0.012
9	9	10	688999	7610000	90	RUS	FLAT	38	1	0.004
10	1	11	489804	7650000	180	FIN	SE	26	31	0.013
11	11	12	718237	7620000	180	RUS	NW	25	20	0.009
12	12	13	427280	7490000	240	FIN	SW	40	7	0.005
13	13	16	597300	7700000	100	NOR	NW	32	52	0.007
14	NA	17	667085	7680000	170	RUS	FLAT	31	21	0.015
15	NA	18	594611	7550000	280	FIN	N	42	1	0.008
16	1	19	514744	7580000	280	FIN	S	30	31	0.009
17	17	20	710133	7570000	210	RUS	SE	35	1	0.007
18	18	21	439900	7650000	280	NOR	W	23	1	0.005
19	NA	22	734135	7670000	100	RUS	FLAT	44	21	0.012

Figure 2.6: Definition of a subset by a vector of the row indices

2.1.5 Graphically

In this case, the subset will be generated by a logical vector which we obtain by defining a polygon in a xy-plot. This polygon contains all the desired samples for the subset. After clicking the "OK" button in the main window (Figure 2.2) another window (Figure 2.7) appears where the x and y coordinates for the plot are requested. By clicking "OK", the plot will be created and it can be started with defining the area. The polygon can be drawn by clicking the left mouse button. Clicking the right button finishes the polygon. This will highlight the points in the area and generate the subset, see Figure 2.8.

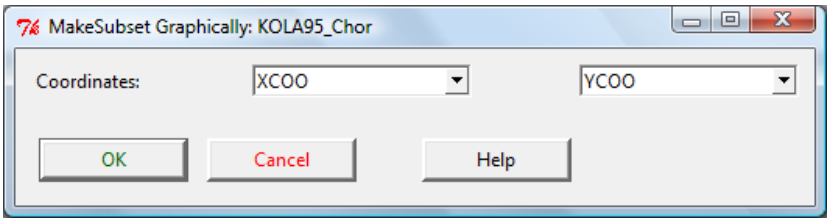


Figure 2.7: Graphical definition–choose coordinates

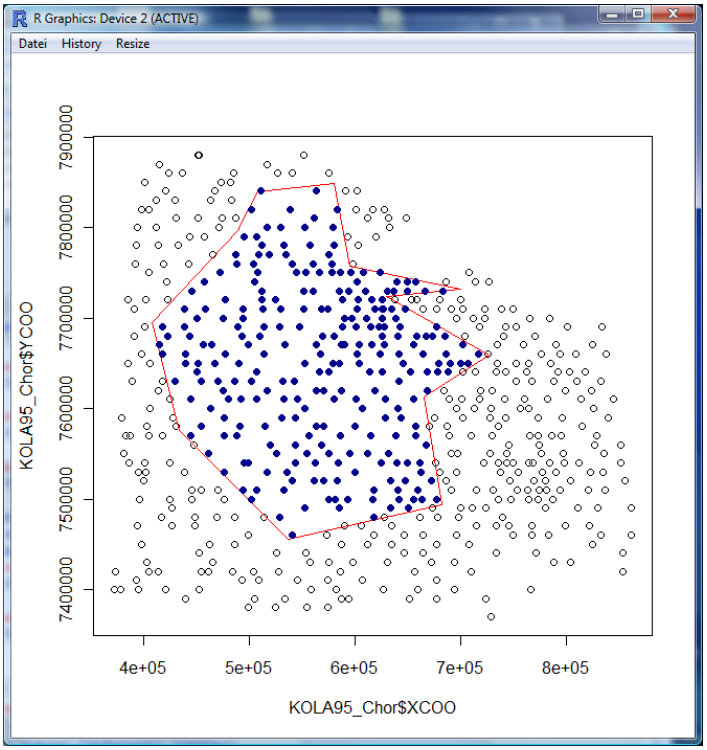


Figure 2.8: Graphical definition–creating via a polygon

2.1.6 Expression

In a second window(Figure 2.9), that appears after clicking "OK" in the main window(Figure 2.2), an expression can be entered as character string. This expression gives a condition, e.g. on a column of the data set, which defines the indices of the subset via a T/F vector. An example can be seen in Figure 2.9. Na is the name of a column (a variable).

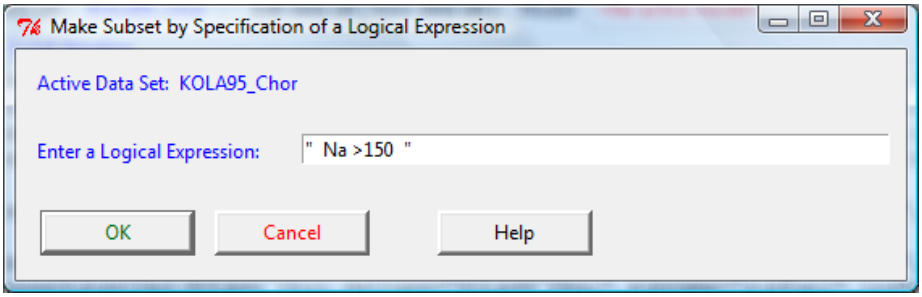


Figure 2.9: Definition of a subset via an expression

2.2 Subset Operations

Figure 2.10 shows the window for generating subsets by using set operations. For all operations except the complement, it is required to use two subsets. The subsets are selected from the two lists. The lists contain the names of the existing subsets and in braces, there is the size of the subsets. As in the main window for generating subsets (Figure 2.2) this window also contains a field for the name and the description of the new subset. Independently of how the subsets in the lists are generated, the new subset is defined by a vector containing the row indices.

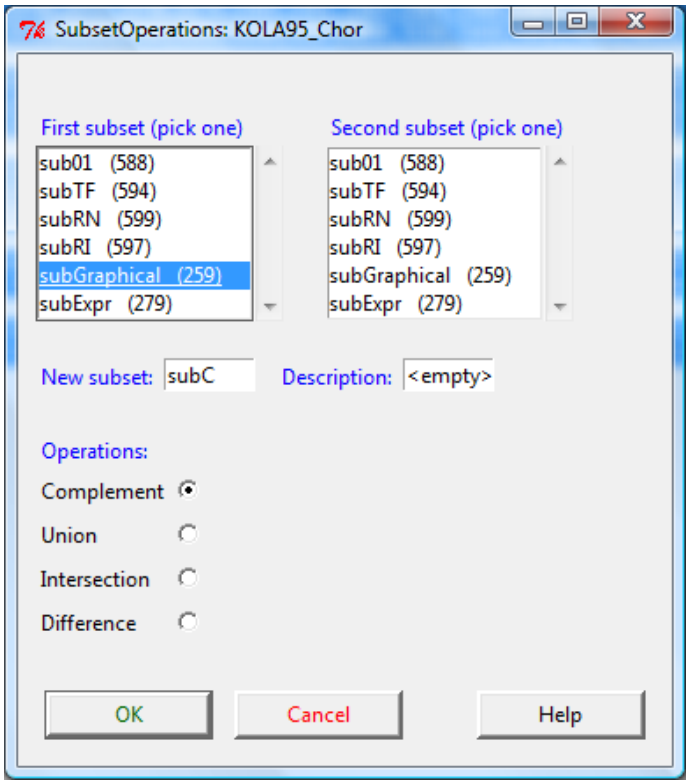


Figure 2.10: Window for generating subsets via operations

2.2.1 Complement

As mentioned above, for this operation only one subset is necessary. It must be selected from the first list. The operation creates a subset that contains all rows of the data set apart from the rows of the selected subset.

2.2.2 Union

The union of two subsets includes all samples that at least one subset contains.

2.2.3 Intersection

The subset generated by the intersection of the two selected subsets includes all samples that both subsets contain.

2.2.4 Difference

The difference is the set of elements included in the selected subset of the first list but not in the subset of the second list.

2.3 Subset Manipulation

After generating subsets it may be necessary to add/cancel samples to/from subsets, change the description or rename them. We can also delete or sort them or display some information. It is also possible to select subsets if we do not want to work with all subsets.

DASplusR has the following functions.

2.3.1 Show

This window displays again a list with all subsets and below some attributes. We see the name with the size of the subset, the type of definition (how the subset was generated) and the description. This window is just for getting some information about the subsets. Therefore there is no difference if the button "OK" or "Cancel" is used for closing this window.

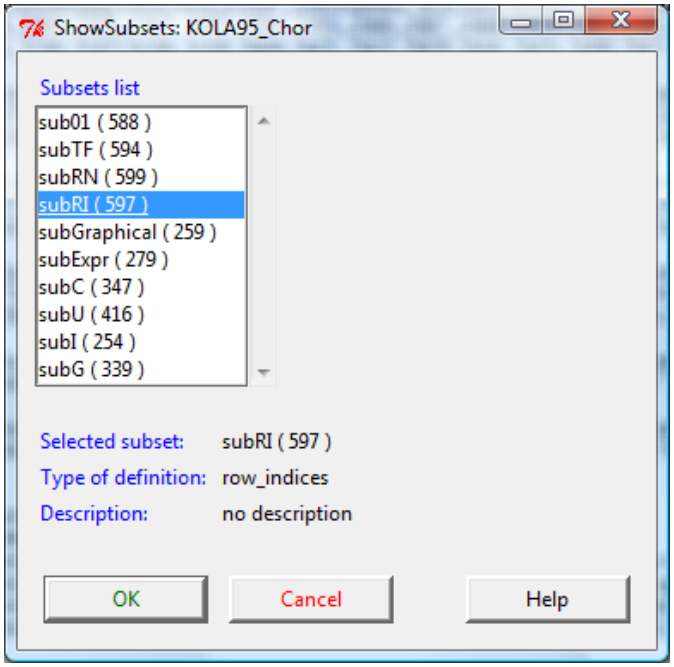


Figure 2.11: Window for showing subsets

2.3.2 Edit

With this window, the user can edit a subset. Its content can be changed as well as the name and the description. By clicking once at the desired subset, the details are shown below the list and the name and the description can be changed. By double clicking a subset name from the list, the window for editing the values immediately appears. Editing the values works in the same way than generating a new subset. Depending on the definition type the different windows appear, see Section 2.1. An exception is the definition by a graph. In this case it works in the same way like editing by row indices. For changing the name, it just has to be replaced in the field "Selected subset". If the new name already exists, the subset will not be renamed.

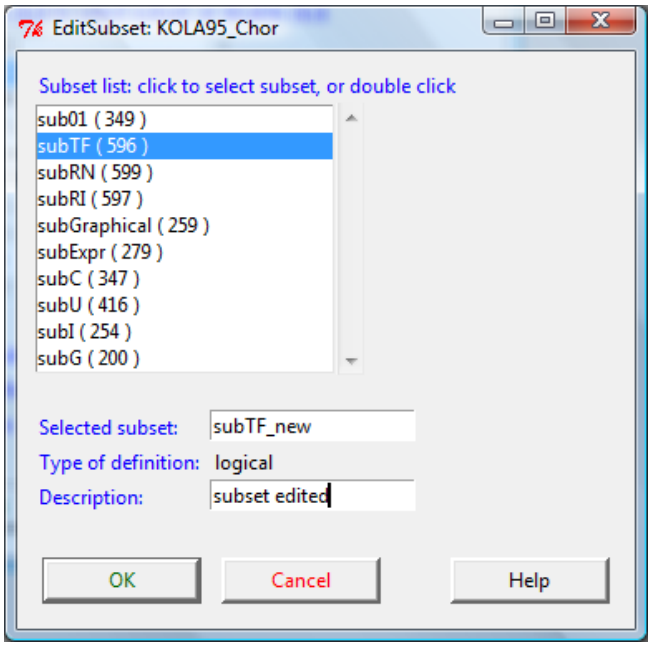


Figure 2.12: Window for editing subsets

2.3.3 Rename

The window for changing the name is shown in Figure 2.13. Select a subset in the list and then enter the new name in the field below. Here it is the same as for editing a subset. The subset will not be renamed if the new name already exists.

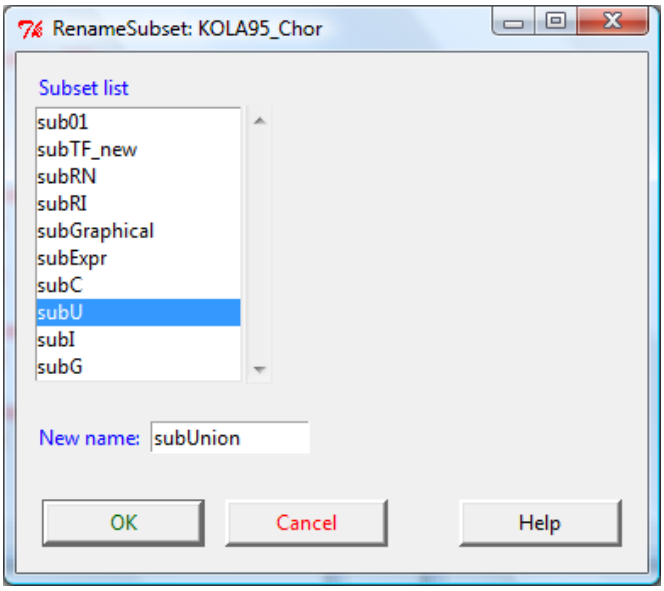


Figure 2.13: Window for changing the name of a subset

2.3.4 Delete

The next window is used to delete subsets. It can either be deleted one or all subsets. For deleting one, the subset has to be selected in the list and then removed by clicking the "Delete" button. Clicking "Delete all" removes all subsets at once. In this window the "OK" and "Cancel" button have the equivalent effect, both just close the window.

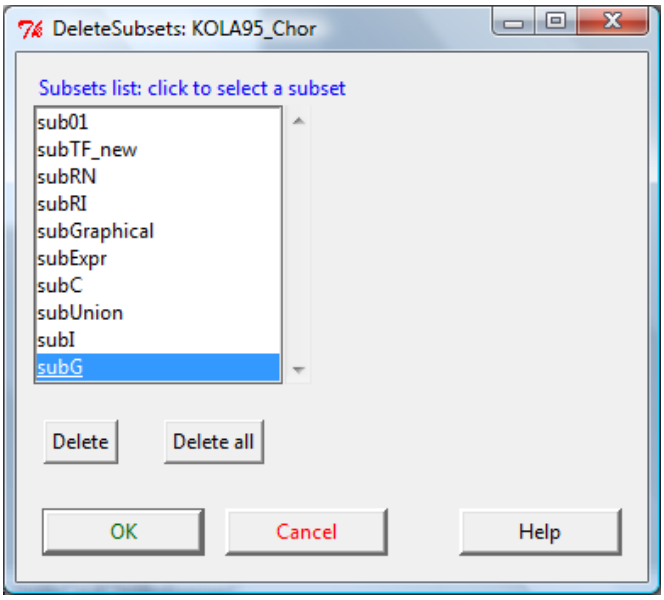


Figure 2.14: Window for deleting subsets

2.3.5 Select

In the window shown in Figure 2.15 some subsets from all existing subsets can be selected. This is useful if someone wants to work with just a few subsets. On the right to the list there are two buttons. One for selecting all subsets and one for deselecting all. For selecting or deselecting one by one, just click on the desired subset in the list which then is ordered.

If the subset is selected it gets a blue background and on the left side of the name appears a number. This number is for the order in which the subsets are selected. By clicking on a selected subset the number and the blue background disappear. If this subset has not the highest number, all higher numbers get reduced by one so that no gap in the order number arise.

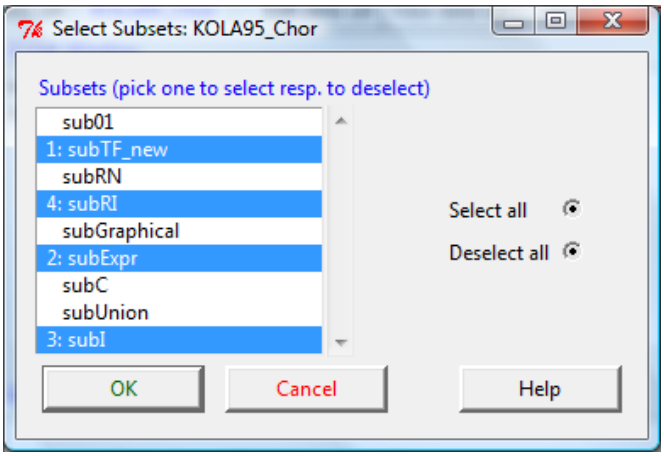


Figure 2.15: Window for selecting subsets

2.3.6 Sort

With this window the subsets can be ordered. This can be done alphabetically or manually. For this one can press one of the first two buttons on the right side of the list in Figure 2.16. With the third button "initial list" the order can be reset to the order at the beginning. Sorting manually works in a similar way to selecting subsets. The difference is that instead of numbers showing the order, the positions in the list change.

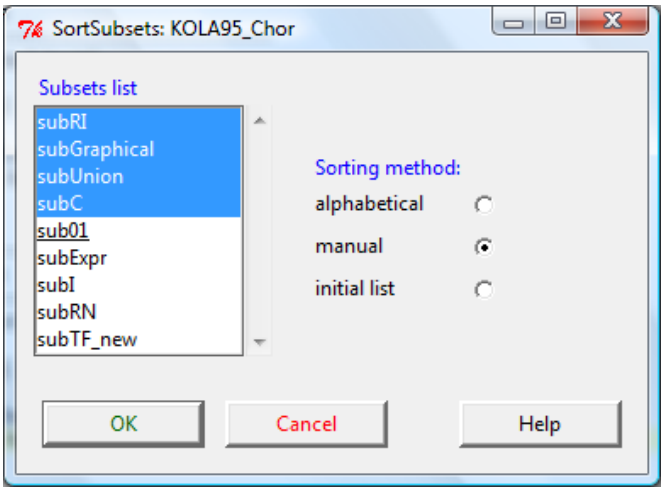


Figure 2.16: Window for sorting subsets

Chapter 3

Usage of Subsets

3.1 Make a New Data Set of a Subset

Sometimes it may happen that a handy subset has been created which should be further considered, and efficiently treated with all the power of DASplusR. Then it is advisable to declare the subset as new active data set.

This can simply be done by clicking

`Edit` → `Subsets` → `As New Data Set`

in the main window of DASplusR (Figure 2.1) which opens a window as shown in Figure 3.1.

The field below the list is for the name of the new data set. This data set has the same samples as the subset. It inherits all attributes from the original data set, also the subsets except the one the data set is created from. The subsets of course just contain samples which are included in the new data set. Therefore the size of the subsets changes. Compare Figure 3.2 with Figure 2.11. For example subset "SubC" contains 347 samples in the original data set, however none of these samples are in the new data set therefore "SubC" now has 0 samples.

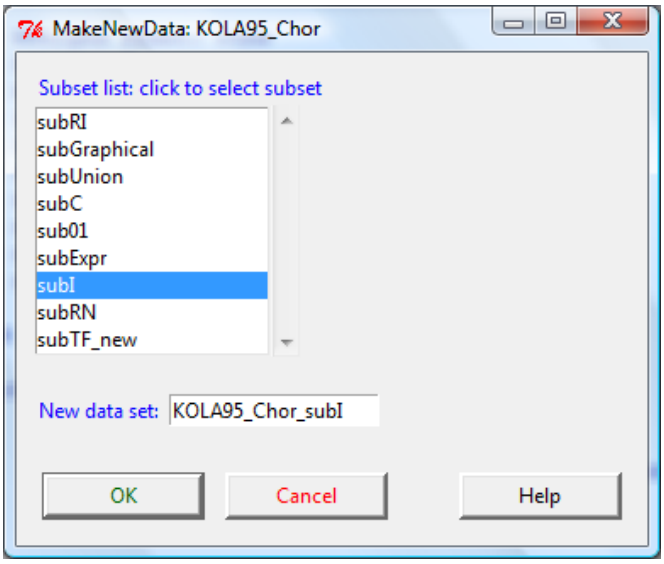


Figure 3.1: Window for making a new data set of a subset

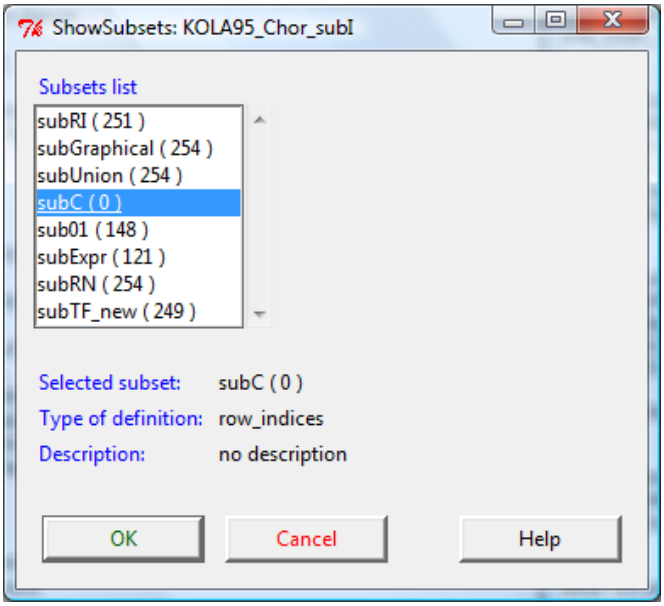


Figure 3.2: Subsets of the new data set

3.2 Usage in Diagrams

In this section we will show how subsets can be used in diagrams.

The windows for creating diagrams can be called by clicking `Diagrams` in the main pull down menu of DASplusR, and then on the desired plot. See Figure 3.3.

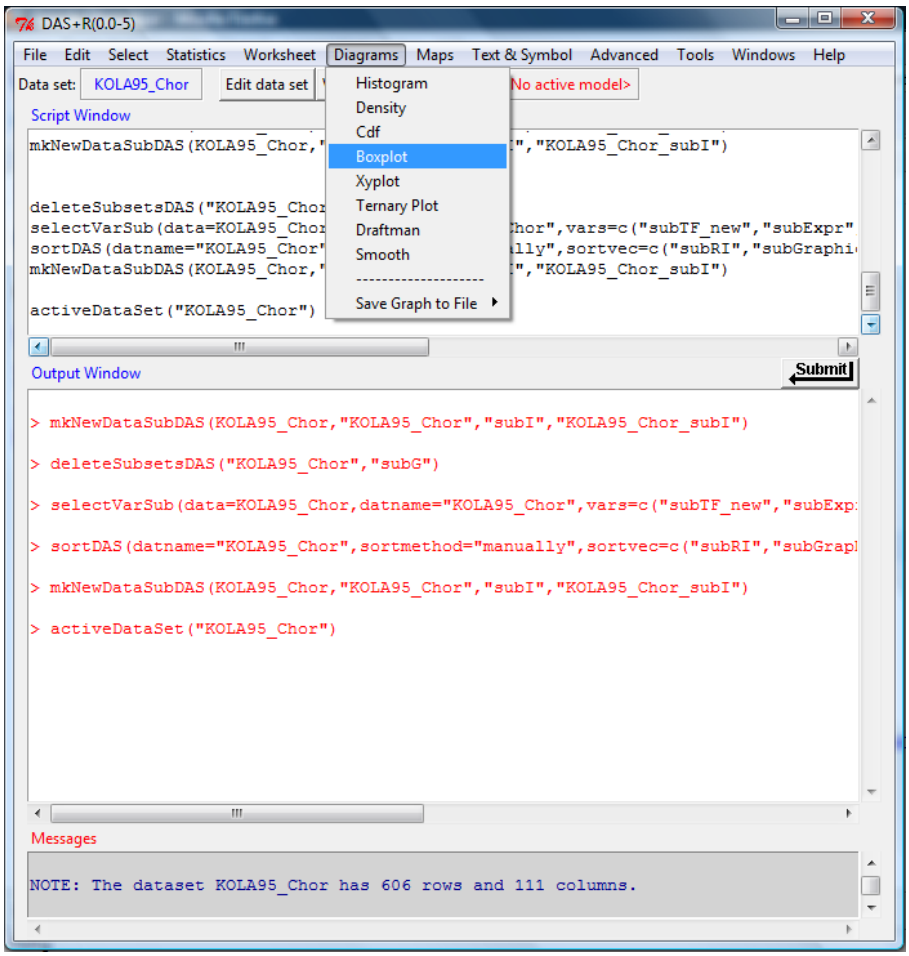


Figure 3.3: DASplusR main window: Diagrams

3.2.1 Boxplot

The main window for creating boxplots is shown in Figure 3.4. The first field in this window is for entering an optional title of the plot. On the left side below this field, there is a list containing variables of the data set. From this list we can choose as many variables as we want by holding the control key while selecting variables. Next to this list there are some important options that can be set. Clicking on the radio button "Further details" opens another window where more options can be specified. This window is shown in Figure 3.5.

In the main window of Boxplot, the three buttons below the list are for plotting the selected variables depending on subsets or groups. Groups are similar to subsets but they are defined by a grouping variable, internally stored as factor. For more information about groups see Reimann et al. (2008). By clicking on the first button a window opens (Figure 3.6) where the desired subsets for the plot can be selected. It works the same way as described in Section 2.3.5. The second button is for selecting groups, while the third one is for deselecting subsets or groups such that boxplots are plotted without any grouping.

Selecting more than one variable or more than one subset leads to a graphic with the corresponding boxplots put beside one another. Figure 3.7 shows the boxplots of two variables("Al" and "Fe") and two subsets("subExpr" and "subRI") which means that four boxplots are displayed. The first boxplot is generated from the samples of "Al" which are in the subset "subExpr". The second is generated from "Al" depending on the subset "subRI". The other two boxplots are created from the variable "Fe".

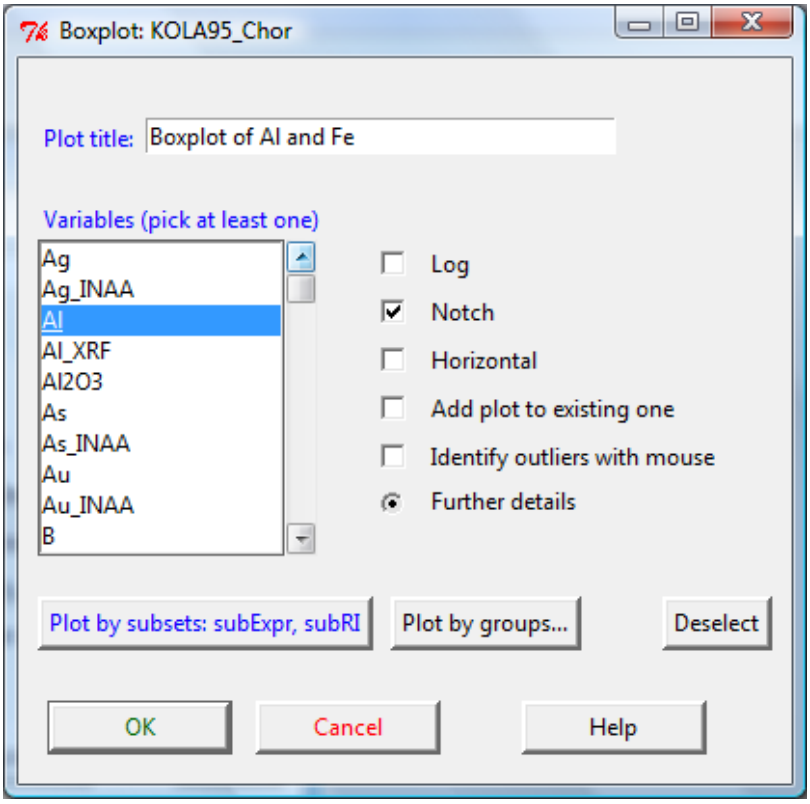


Figure 3.4: Main window for boxplot

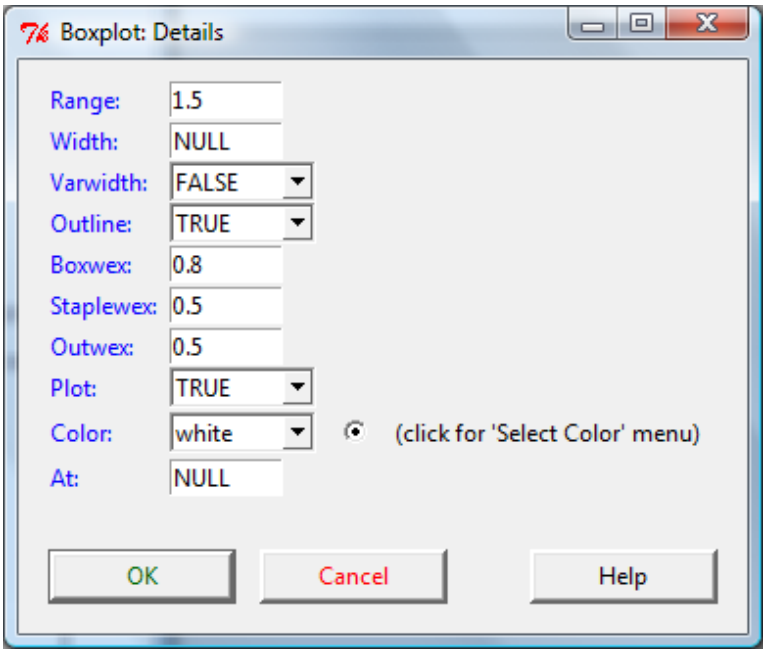


Figure 3.5: Details for boxplot

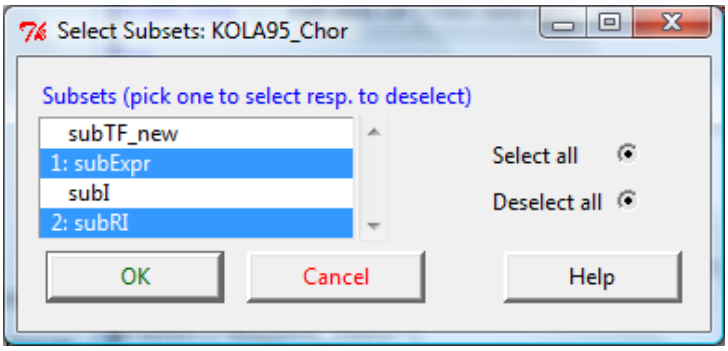


Figure 3.6: Selecting subsets for boxplot

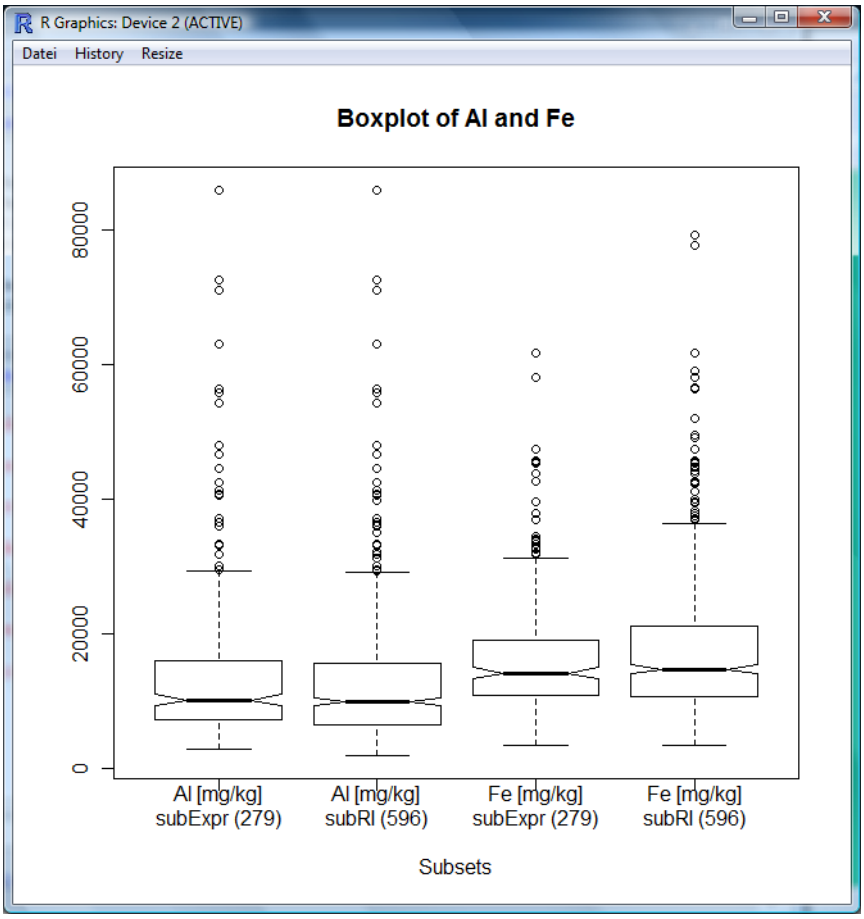


Figure 3.7: Boxplot of two variables with two subsets

3.2.2 Ternary Plot

The window for creating a ternary plot (Figure 3.8) contains three lists, one for each variable to be used. Above the lists the field for the title of the plot can be found. Below the lists there is a button which opens the details window (Figure 3.9) and a checkbox. If it is marked, samples can be identified in the plot. Below the checkbox we find the three buttons for subsets and groups as described in Section 3.2.1. The difference to the boxplot is that just one ternary plot is drawn. Therefore every sample in the data set appears just once. If the same sample is included in several subsets then it gets the color and the symbol of the first subset which includes it. This can be seen in Figure 3.10. The subset subRI contains more samples than plotted, because they are covered by the elements of subset "subExpr".

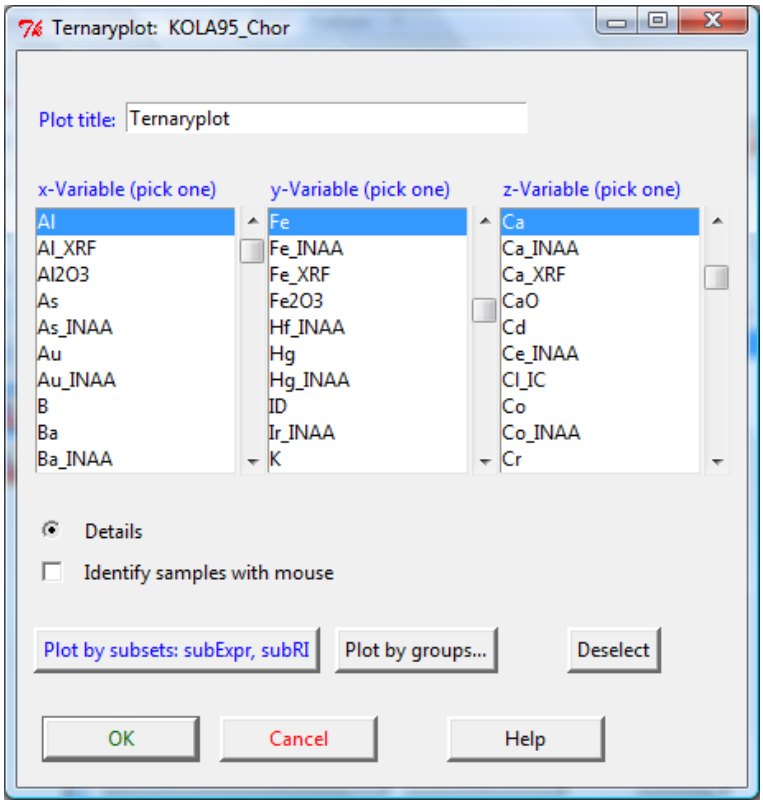


Figure 3.8: Main window for ternary plot

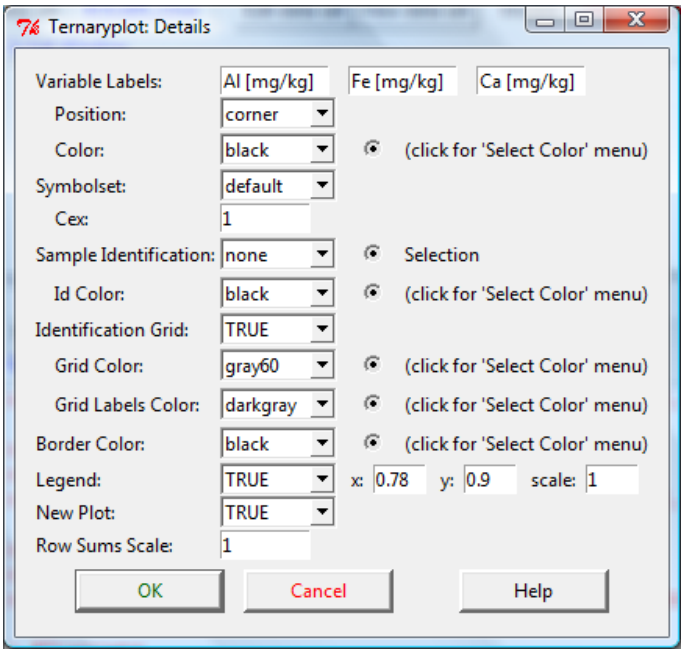


Figure 3.9: Details window for ternary plot

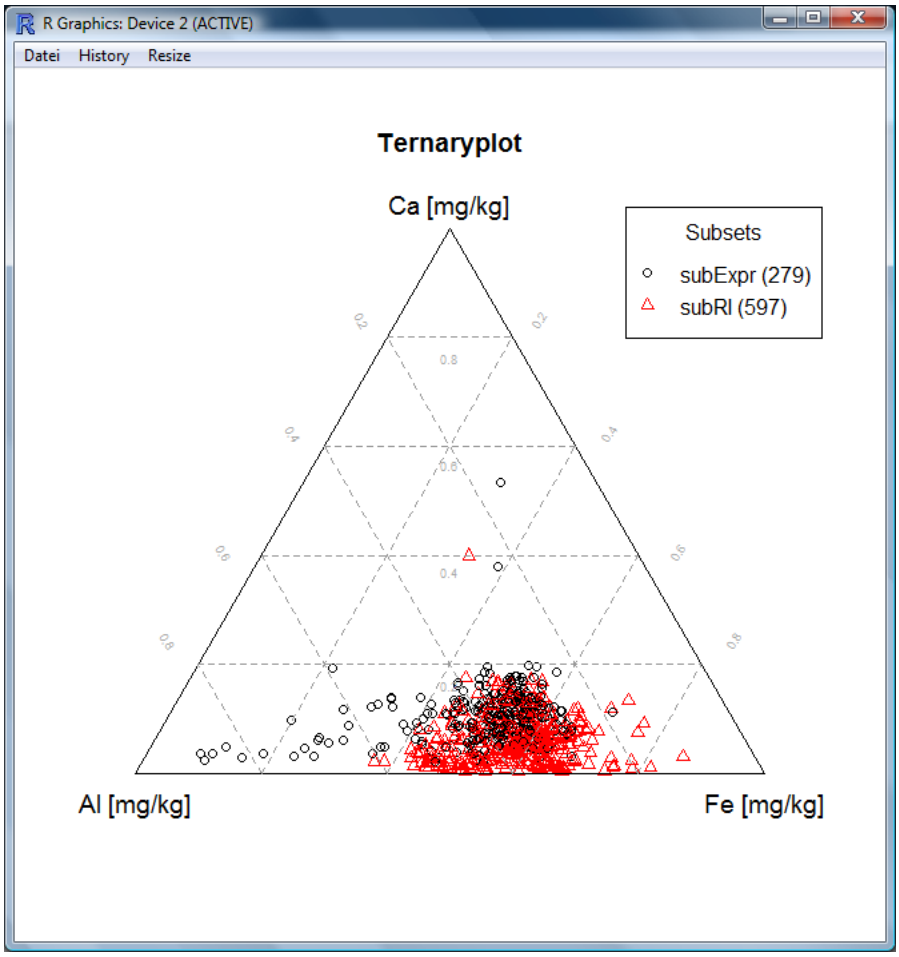


Figure 3.10: Ternary plot with two subsets

3.2.3 Xy-plot

Like for all other plots, this main window contains a field for the plot title, lists for choosing variables and the three buttons for selecting subsets or groups. The left list is for choosing the variable on the x-axis and the right list for the variable on the y-axis. Below each list there is a checkbox for plotting the x and/or y coordinates in the logarithmic scale. After the buttons for the subsets/groups there are several radiobuttons.

The first button and the second one open windows where details of the axes can be set. The next one is for the symbols used in the plot. In that window (Figure 3.12) for each subset/group element there are three fields. In the first one, the symbol or for a special symbol, the name can be entered. If the name is

unknown it can be chosen by clicking on the button next to the field. This implies the opening of a window as shown in Figure 3.13. The second field is for the color of the symbol. As for the symbol, the color can be entered in the field or chosen by clicking at the button next to the field. See Figures 3.14 and 3.15. The size of the symbol can be changed in the third field.

With the next button in the main window of xy-plot the details for the legend can be set. In this case the window Figure 3.16 opens. With the pull down menu the option of a legend to appear can be specified, and if so, whether it is positioned by entering the coordinates or by clicking at the position in the plot. In the next two fields, the coordinates of the upper left corner of the legend can be entered. The last field on this window sets the scale.

"Additionals" opens a window where more options for the plot can be set. With the checkbox, the xy-plot can be drawn in an existing graphics window or it appears in a new one.

As described in the Section before it might happen that some values from subsets are covered by other subsets, see Figure 3.17. In this example, subset `subRI` contains 251 samples but only 3 show up because they are also contained in the other two.

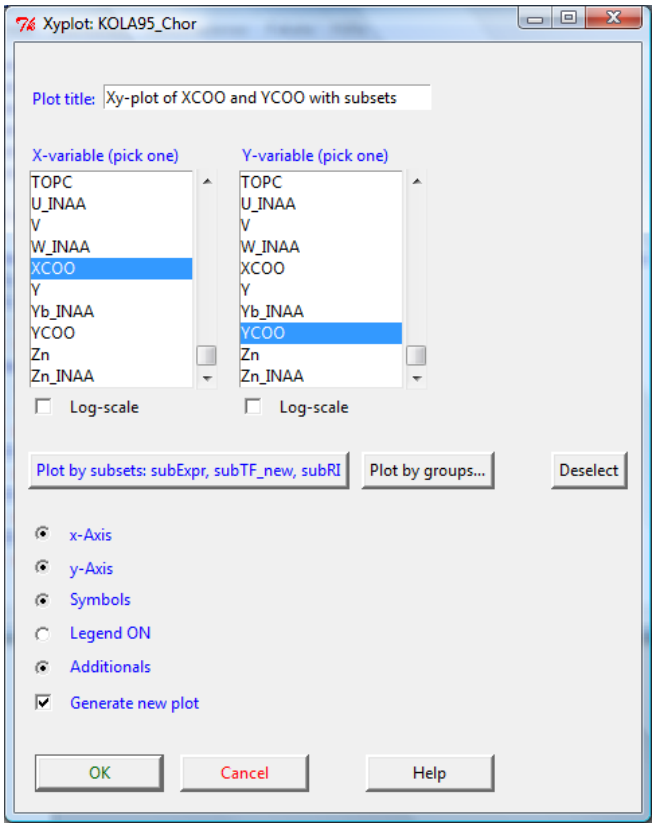


Figure 3.11: Main window for xy-plot

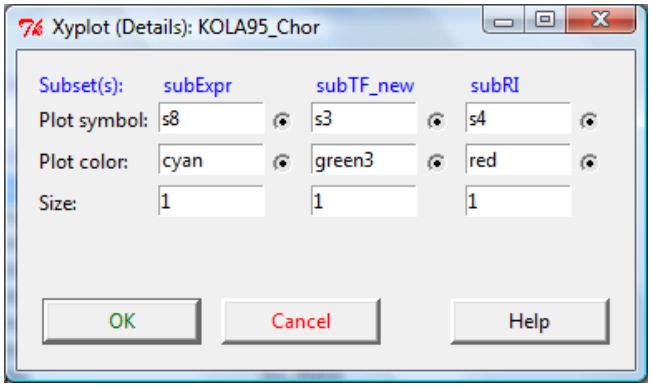


Figure 3.12: Symbols for xy-plot

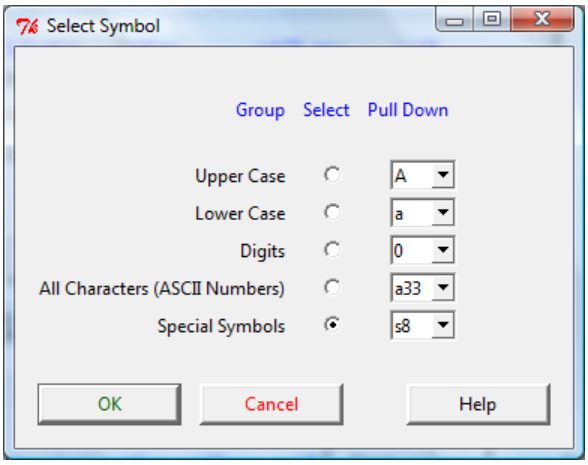


Figure 3.13: Selecting a symbol

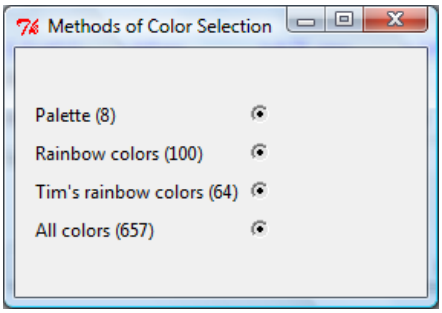


Figure 3.14: Method of color selection

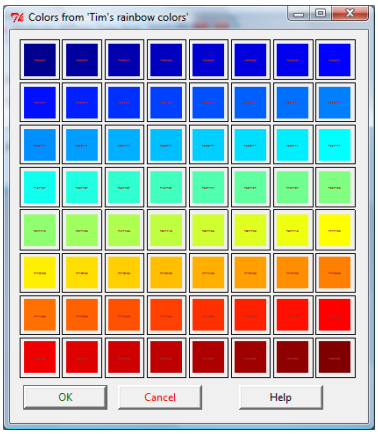


Figure 3.15: Selecting a color

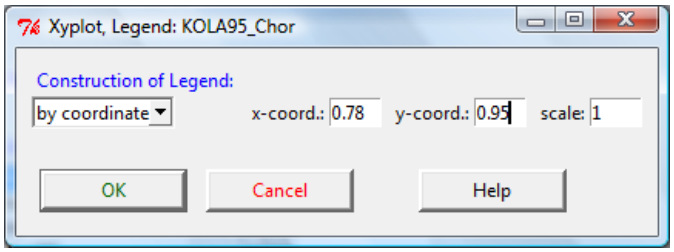


Figure 3.16: Legend details

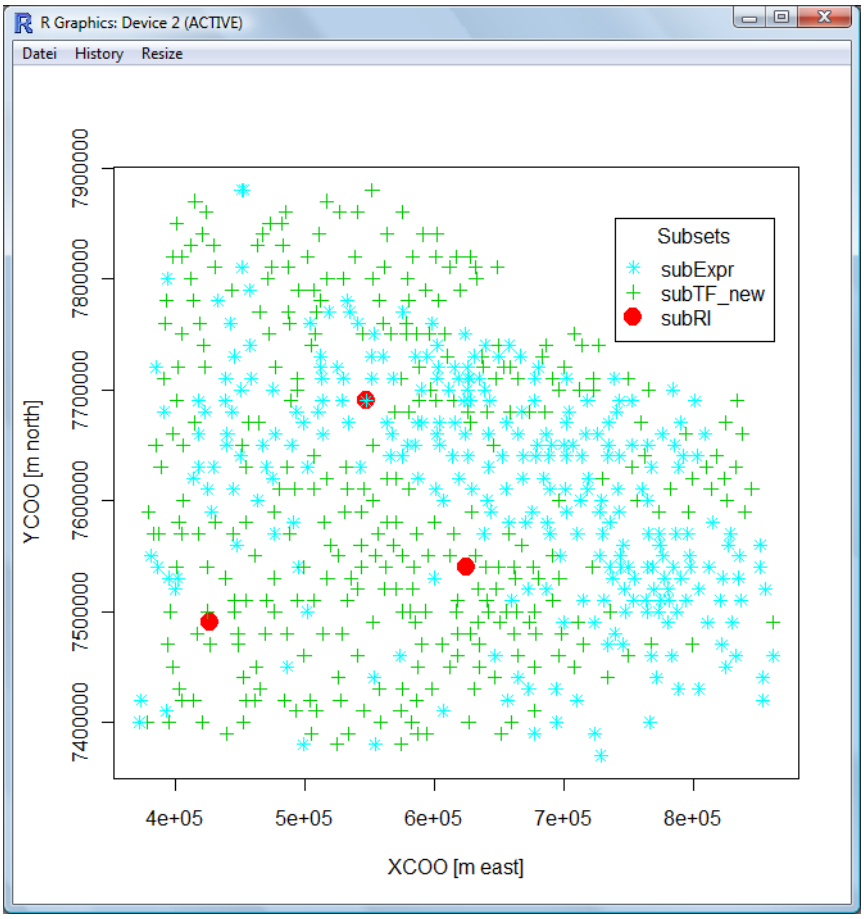


Figure 3.17: Xy-plot with three subsets

3.2.4 Density

The window for plotting densities is similar to that of the xy-plot, see Figure 3.18. The difference is that here just one list appears where several variables can be chosen. Another difference is that the third button opens a window for lines instead of symbols. In such a window the first row is for the line type of the density trace, the second for the color and the third specifies the line width of the trace. See Figure 3.19. After clicking the "OK" button the plot will be drawn. A possible result is shown in Figure 3.20.

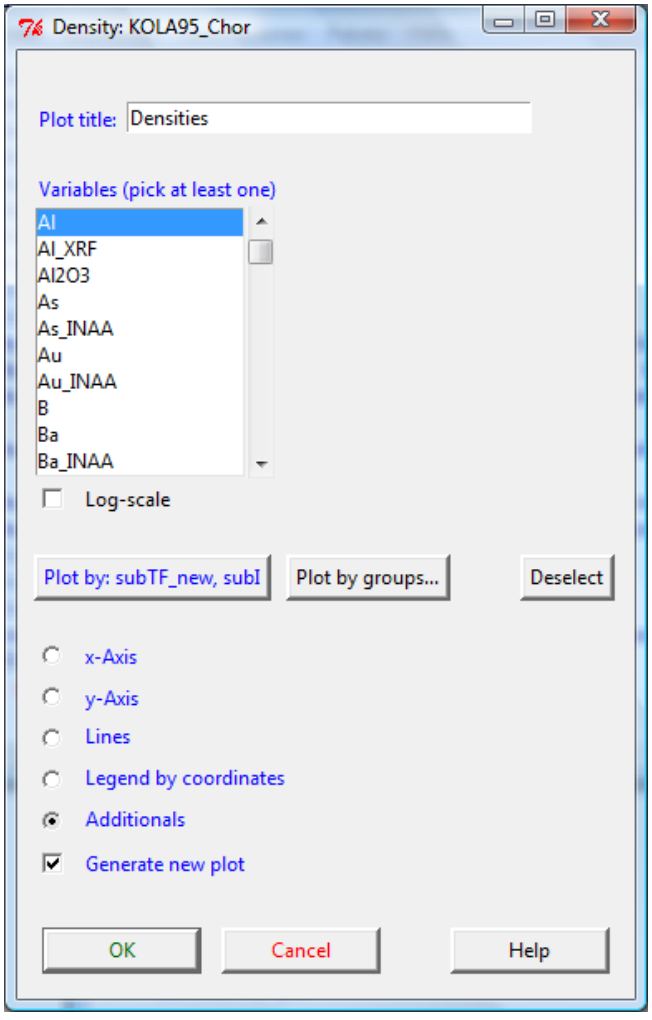


Figure 3.18: Main window for plotting densities

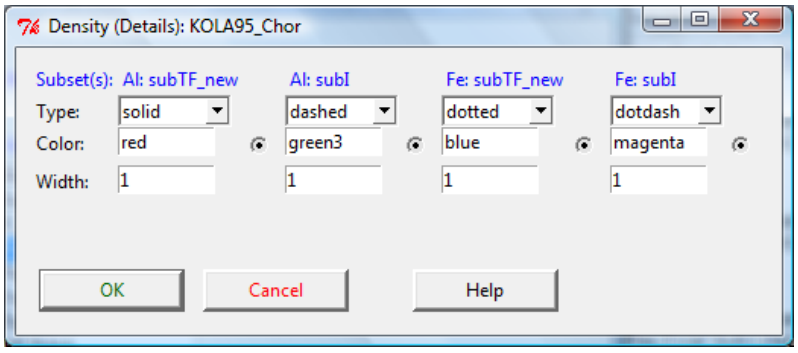


Figure 3.19: Lines for density plot

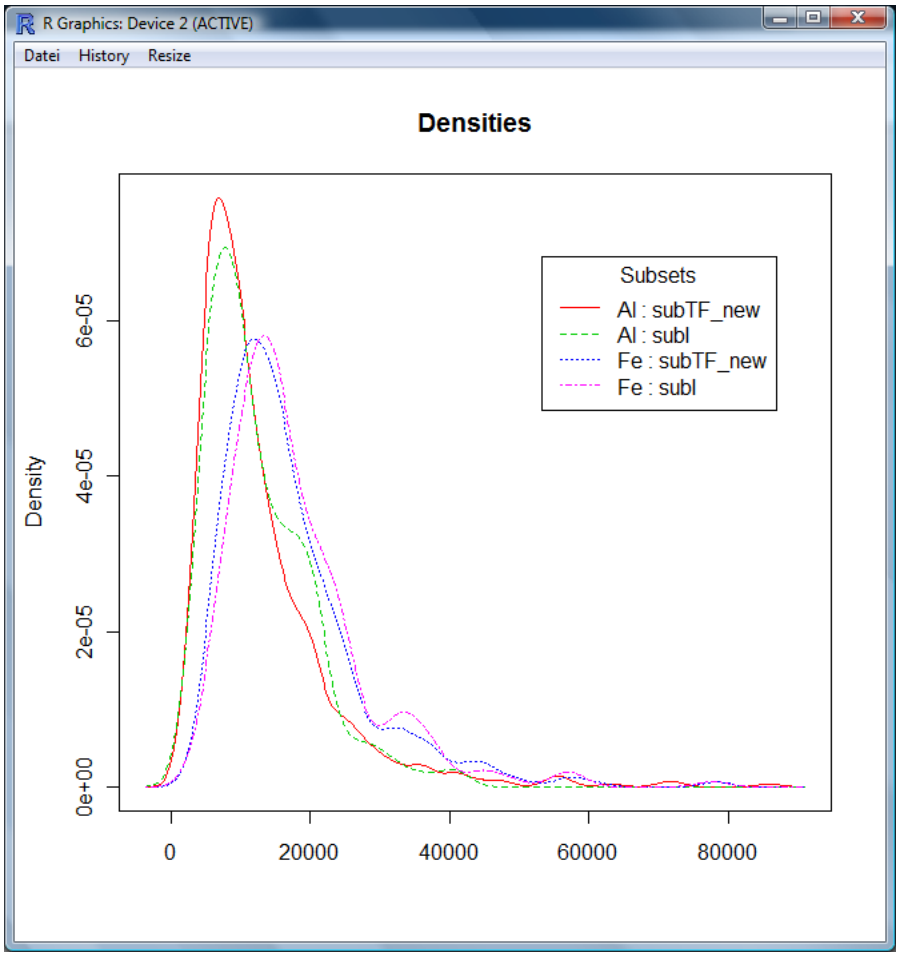


Figure 3.20: Densities of two variables and two subsets

3.2.5 Histogram

In this case we are restricted to one variable and at most one subset or group.

The upper part of the main window (Figure 3.21) for plotting a histogram contains the field for the plot title, the variables list, a checkbox for logarithmic scale and the three buttons for subsets/groups. This resembles the main window for plotting densities. The next element in the histogram window is a field for the number of bins (classes of the histogram). Below this field three checkboxes for adding a density, a scatterplot and/or a boxplot are present. Next to these checkboxes, buttons for further details of the additional plots are available. Underneath there are the buttons for opening the subwindows for details of the x-axis and for additional specifications. The final checkbox sets the option for plotting in an empty graphic sheet which is filled after clicking the "OK" button, otherwise, the previous graphic is overplotted.

A possible result is presented in Figure 3.22.

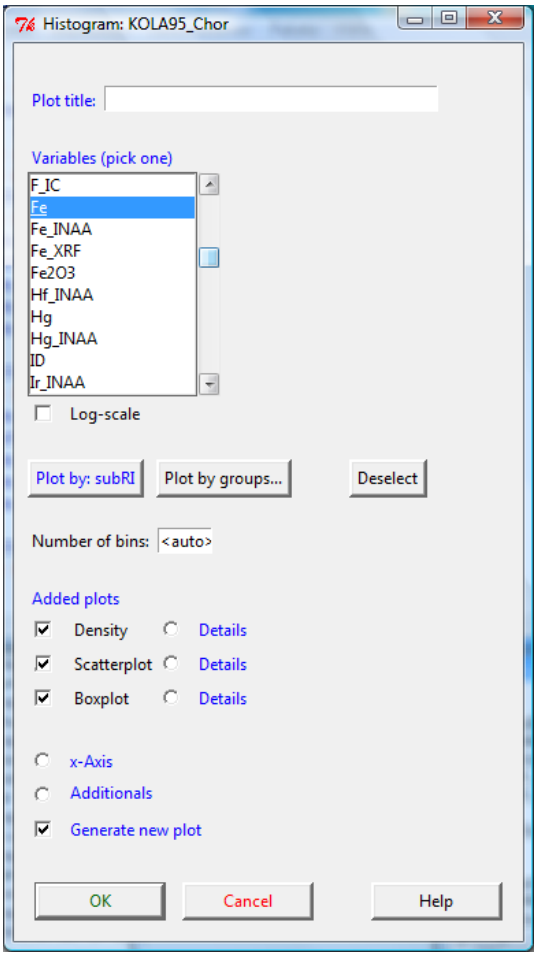


Figure 3.21: Main window for plotting a histogram

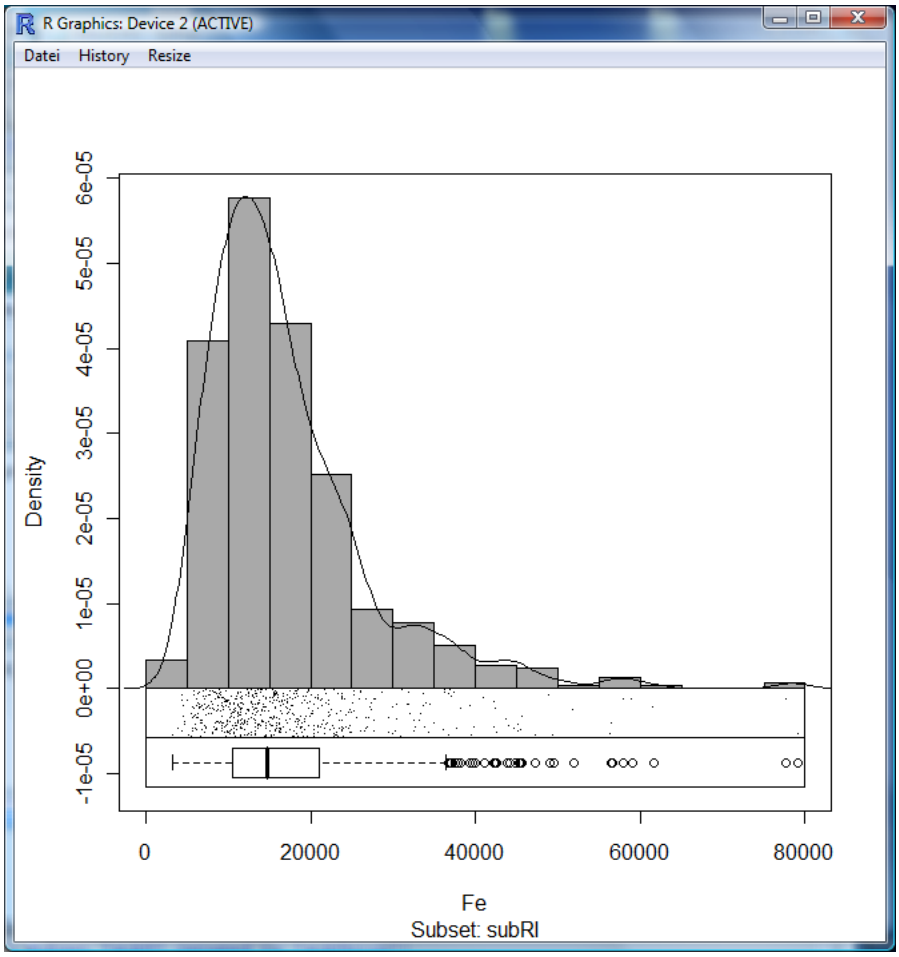


Figure 3.22: Histogram with density, scatterplot and boxplot

Chapter 4

Conclusion

In Chapter 2 we saw the different possibilities of how subsets can be defined in DASplusR, namely by a 0/1 vector, a T/F vector, a vector containing the sample names, a sample indices vector, graphically, by an expression or by set operations (complement, union, intersection or difference). This chapter also includes how subsets can be manipulated.

In Chapter 3 we found some applications of subsets in DASplusR. On the one hand the creation of a new data set was described and on the other hand we saw their usage in diagrams.

The following pages contain a reference list, the R help files for the functions and the commands used for generating the Figures 3.7, 3.10, 3.17, 3.20 and 3.22.

Bibliography

J. Fox (2004). Getting Started with the R Commander: A Basic-statistics Graphical User Interface to R. *'useR 2004' Conference*, May 20-22, 2004, Vienna University of Technology, Austria.

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

B.B. Welch and K. Jones (2003). *Practical Programming in Tcl and Tk. Prentice Hall PTR, New York.*

Appendix A

Help Files

A.1 GUI and Details Functions

These help files are for the functions which produce the GUI and the details windows where all parameters for the executable functions can be specified. These files can be opened by clicking at the "Help" button in the current windows, or simply by the -command `help(`*`function`*`)`.

`boxplotDetails` *Details for the boxplot in DASplusR*

Description

The function boxplotDetails opens a dialog window where details for the creation of a boxplot are specified.

Usage

```
boxplotDetails()
```

Dialog Elements

Color: The color of the body of the boxplot.

For all other elements, see: `boxplot`.

Details

For `width` and `at`, the entered numbers have to be separated by commas. For example: 1,2,3.

The length of `width` and `at` must be the same as the number of plotted boxplots.

Note

This function can only be used by clicking `Further details` in the GUI of Boxplot, see `BoxplotGUI`.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`boxplotDAS`, `BoxplotGUI`, `identify`, `boxplot`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),names=c("Al","Fe"))
boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),names=c("Al","Fe"),
  at=c(2,1))
boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),names=c("Al","Fe"),
  at=c(2,1),width=c(1,2))
```

`BoxplotGUI` *GUI (Graphical User Interface) for boxplots of the active data set in DASplusR*

Description

This function opens a dialog window where all settings for the creation of a boxplot are specified.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Diagrams -> Boxplot.

## R Function Usage:
  BoxplotGUI()
```

GUI Elements

Plot title: The main title of the boxplot.

Variables: Variable(s) which is(are) used for the plot.

Log: see `log` in `boxplot`.

Notch: see `notch` in `boxplot`.

Horizontal: see `horizontal` in `boxplot`.

Add plot to existing one: see `add` in `boxplot`.

Identify outliers with mouse: If checked, the outliers in the plot can be identified by placing the cursor close to a point and pressing the left mouse button. Then the ID-number or the row number of this point is plotted. Pressing the right mouse button will finish the identification process.

Further details: Clicking *Further details* a dialog box appears which allows to specify additional plotting options.

Plot by subsets...: Clicking this button a dialog box appears which allows selecting subsets.

Plot by groups...: Clicking this button a dialog box appears which allows to select a grouping variable. Clicking the OK button in this dialog box a further box appears where the levels of the grouping variable can be selected. Either subsets or groups can be used.

Deselect: By clicking this button the selected subsets or groups will be deselected

Note

This function assumes that a data set containing at least 1 numerical variable is active.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`boxplotDAS`, `boxplotDetails`, `identify`, `boxplot`

`DeleteSubsetsGUI` *GUI (Graphical User Interface) for deleting subsets of the active data set in DASplusR*

Description

This function opens a dialog window which offers the list of subsets for possible deletion.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Delete.

## R Function Usage:
  DeleteSubsetsGUI()
```

GUI Elements

Subsets list: List of existing subsets.

Delete: Button for deleting the selected subsets.

Delete all: Button for deleting all subsets.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`deleteSubsetsDAS`

DensityGUI	*GUI (Graphical User Interface) for plotting densities in DASplusR*

Description

This function opens a dialog window where all settings for the creation of a density plot are specified.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icon

  Diagrams -> Density.

## R Function Usage:
  DensityGUI()
```

GUI Elements

Plot title: The main title of the density plot.

Variables: Variable(s) which is(are) used for the plot.

Log-scale: If clicked the x- resp. y-variable should be log-transformed.

Plot by subsets...: Clicking this button a dialog box appears which allows selecting subsets. Either subsets or groups can be chosen.

Plot by groups...: Clicking this button a dialog box appears which allows to select a grouping variable. Clicking the OK button in this dialog box a further box appears where the levels of the grouping variable can be selected. Either subsets or groups can be chosen.

Deselect: By clicking this button the selected subsets or groups will be deselected.

x-Axis: Clicking this button opens a dialog box allowing to redefine specifications for the x-axis. See `GUIElements`.

y-Axis: Clicking this button opens a dialog box allowing to redefine specifications for the y-axis. See `GUIElements`.

Lines: Clicking this button opens a dialog box allowing to redefine specifications for the lines to be used. See `GUIElements`.

Legend: Clicking this button opens a dialog box allowing to redefine specifications for a possible legend. See `GUIElements`.

Additionals: Clicking this button opens a dialog box allowing to redefine specifications for additional demands like identifying points, adding lines, correlations, etc. See `GUIElements`.

Generate new plot: Checking this box implies the plotting in a new graphic windows, otherwise the previous graphic is overplotted.

Note

This function assumes that a data set containing at least 1 numerical variable is active.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`densityDAS`, `density`, `GUIElements`

`EditSubsetGUI` *GUI (Graphical User Interface) for editing a subset of the active data set in DASplusR*

Description

This function opens a dialog window for editing an existing subset.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Edit.

## R Function Usage:
  EditSubsetGUI()
```

GUI Elements

Subsets list: List of the existing subsets.

Selected subset: Name of the subset to be edited.

Type of definition: Type of how the subset is defined. See `makeSubset`.

Description: Description of the subset. See `makeSubset`.

Details

This function calls the function `makeSubset` and overwrites the existing subset.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`makeSubset`

`GUIElements` *Elements contained in the Graphical User Interfaces (GUI's) of the plots in DASplusR*

Description

A description of the Elements of the Graphical User Interfaces (GUI's) for plots in DASplusR.

GUI Elements: x- resp. y-axis

Margins: The page margins (in units 'lines') may be specified.

Limits: The minimum and the maximum of the points to be plotted may be specified.

Label: The label of the axis, its font and size may be specified.

Line type: The line type, width and color of the axis may be specified.

Tick marks: The box may be checked and the number of tick marks specified.

Tick labels: The box may be checked and the size and font of the labels of the tick marks specified.

Grid: A grid may be required, and its type, color and width may be specified.

Frame: A frame may be specified (this is for x and y).

Jitter points: A jittering of the points (symbols) and a factor may be specified.

GUI Elements: Symbols

The type, color and the size of each used symbol can be specified.

GUI Elements: Lines

The type, color and the width of each used line can be specified.

GUI Elements: Legend

Construction of Legend: 3 methods may be specified: 'none', 'by clicks' and 'by coordinates'. Additionally, the x- and y-coordinates of the left-upper corner and the scale (relative size) may be specified.

GUI Elements: Additionals

Identify samples: Here you may indicate identification by a list which can be specified, or via the cursor by clicking. See `identify`.

Add line: Lines can be requested: ls, robust or smooth.

Add correlation coefficient: Different kinds of correlation coefficients may be requested. The results are printed.

Define subset: A subset may be defined graphically.

Title, Size: The size of the plot title can be changed.

Frame around the graphic: A frame around the plot can be drawn.

Note

Not every element exists in all of the diagrams.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`xyplotGUI`, `DensityGUI`, `HistGUI`

HistGUI	*GUI (Graphical User Interface) for plotting a histogram of the active data set in DASplusR*

Description

This function opens a dialog window where all settings for the creation of a histogram are specified.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Diagrams -> Histogram.

## R Function Usage:
  HistGUI()
```

GUI Elements

Plot title: The main title of the histogram.

Variables: Variable which is used for the plot.

Log-scale: If clicked the x- variable should be log-transformed.

Plot by subsets...: Clicking this button a dialog box appears which allows selecting subsets. Either subsets or groups can be chosen.

Plot by groups...: Clicking this button a dialog box appears which allows to select a grouping variable. After clicking the OK button in this dialog box, a further box appears where the levels of the grouping variable can be selected. Either subsets or groups can be chosen.

Deselect: By clicking this button the selected subsets or groups will be deselected.

Number of bins: Number of cells for the histogram.

x-Axis: Clicking this button opens a dialog box allowing to redefine specifications for the x-axis. See `GUIElements`.

Added plots: By checking these boxes a boxplot, scatterplot and/or a density trace can be added to the histogram.

Additionals: Clicking this button opens a dialog box allowing to redefine specifications for additional demands like identifying points, adding lines, correlations, etc. See `GUIElements`.

Generate new plot: Checking this box implies the plotting in a new graphic window, otherwise the previous graphic is overplotted.

Note

For this function just one variable and one subset or group element can be used. If more are selected, just the first one is used.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`GUIElements`, `histDAS`,

`MakeNewDataGUI` *GUI (Graphical User Interface) for generating a new data set from a subset in DASplusR.*

Description

This function opens a dialog window for generating a new data set from a subset with the same number of samples and attributes.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> As New Data Set.

## R Function Usage:
  MakeNewDataGUI()
```

GUI Elements

Subsets list: List of the existing subsets.

New data set: Name of the new data set.

Details

This function calls the function `makeNewDataSubDAS` where the changes are stored.

The new data set has the same samples as the subset. It inherits all attributes from the original data set, also the subsets except the one the data set is created from. The subsets of course just contain samples which are included in the data set. Therefore the cardinality of the subsets changes.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`makeNewDataSubDAS`

`MakeSubsetGUI` *GUI (Graphical User Interface) for creating a subset of the active data set in DASplusR*

Description

This function opens a dialog window for creating a subset.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Make.

## R Function Usage:
  MakeSubsetGUI()
```

GUI Elements

Subsets list: List of the existing subsets.

New subset: Name of the new subset. It must start with a letter.

Description: Description of the subset. See `makeSubset`.

Type of definition: Type of how the subset is defined. See `makeSubset`.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`makeSubset`

`RenameSubsetGUI` *GUI (Graphical User Interface) for renaming a subset of the active data set in DASplusR*

Description

This function opens a dialog window for changing the name of a subset.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Rename.

## R Function Usage:
  RenameSubsetGUI()
```

GUI Elements

Subsets list: List of existing subsets.

New name: Field for the new name.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`renameSubsetDAS`

`SelectSubsetsGUI` *Function calling SelectVarSub for subsets.*

Description

This function prepares the arguments necessary for the function `SelectVarSub` which opens the window for selecting subsets.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Select.

## R Function Usage:
  SelectSubsetsGUI()
```

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`SelectVarSub`, `selectVarSub`, `SelectVariables`

`ShowSubsetsGUI` *GUI (Graphical User Interface) for showing the subsets of the active data set in DASplusR*

Description

This function opens a dialog window where the existing subsets are shown.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Show.

## R Function Usage:
  ShowSubsetsGUI()
```

GUI Elements

Subsets list: List of the existing subsets.

Selected subset: Name of the subset from which the description and the definition type are displayed.

Type of definition: Type of how the subset is defined. See `makeSubset`.

Description: Description of the subset. See `makeSubset`.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`makeSubset`

`SortSubsetsGUI`	*GUI (Graphical User Interface) for sorting the subsets of the active data set in DASplusR*

Description

This function opens a dialog window for sorting the subsets.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Sort.

## R Function Usage:
  SortSubsetsGUI()
```

GUI Elements

Subsets list: List of the existing subsets.

Sorting method: Method how the subsets are sorted. It can be alphabetically or manually. The third possibility is to reset it to the initial order.

Details

This function calls the function `sortDAS` where the changes are stored.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`sortDAS`

`SubsetOperationsGUI`

GUI (Graphical User Interface) for making a subset using existing subsets of the active data set in DASplusR

Description

This function opens a dialog window for making a new subset by the complement of a subset, or the union, intersection, or the difference of 2 subsets.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Edit -> Subsets -> Operations.

## R Function Usage:
  SubsetOperationsGUI()
```

GUI Elements

First subset: The first subset which is used for the operations.

Second subset: The second subset which is used for the operations.

New subset: The name of the new subset. It must start with a letter.

Description: A description of the new subset.

Complement: The complement of the first subset. See `setdiff`.

Union: The union of the first and the second subset. See `union`.

Intersection: The intersection of the first and the second subset. See `intersect`..

Difference: The set of elements in first subset but not in second subset. See `setdiff`.

Details

The second subset is not used for the `complement`. For the other operations both subsets are needed.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`subsetOperationsDAS`, `makeSubset`

`ternaryplotDetails`

Details for the ternaryplot of the active data set in DAS-plusR

Description

The function ternaryplotDetails opens a dialog window where details for the creation of a ternaryplot are specified.

Usage

```
ternaryplotDetails()
```

Dialog Elements

Variable Labels: Variable labels (by default, names of the three selected variables including their units if available).

Position: Position of variable labels.

Color: Color of the variable labels.

Symbolset: The used symbol set. For symbol set see `makeSymbolSet`.

Cex: A numerical value giving the amount by which plotting text and symbols should be scaled relative to the default.

Sample Identification: Optional labels to be plotted below the plot symbols.

Id Color: Color of sample identification.

Identification Grid: If `TRUE`, a grid is plotted.

Grid Color: Grid color.

Grid Labels Color: Color of the grid labels.

Border Color: Color of the triangle border.

Legend: If `TRUE`, a legend is plotted. `x` and `y` are the coordinates where the legend is plotted. `scale` is the scale of the legend.

New Plot: If `TRUE`, the plot will appear on a new graphics page.

Row Sums Scale: Row sums scale to be used (default is 1).

Details

pch, col of the plotted symbols can be changed by selecting of a symbol set. This can be done in the combo box "Symbolset", by selecting a symbol set which already contains the desired specifications.

The combo box "Legend" is effective only if subsets or groups are selected.

Sample Identification: The values of the identification variable of the data set are used. If no identification variable is specified, the row names of the data set will be used.

Note

This function can only be used by clicking `Details` in the GUI of ternaryplot, see `TernaryplotGUI`.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`ternaryplotDAS`, `TernaryplotGUI`, `identify`

`TernaryplotGUI` *GUI (Graphical User Interface) for ternaryplot of the active data set in DASplusR*

Description

This function opens a dialog window where all settings for the creation of a ternaryplot are specified.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Diagrams -> Ternary Plot.

## R Function Usage:
  TernaryplotGUI()
```

GUI Elements

Plot title: Is the main title of the ternaryplot.

x-Variable, y-Variable, z-Variable: The variables which are used for the plot.

Details: Clicking `Details`, a dialog box appears which allows to choose additional plotting options. See `ternaryplotDetails`.

Identify samples with mouse: If checked, the points in the plot can be identified. When the cursor is above a point and the left mouse button is pressed, the ID-number of this point is plotted. Pressing the right mouse button will end the identification process. For ID-numbers, see details below.

Plot by subsets...: Clicking this button a dialog box appears which allows to select subsets. Either subsets or groups can be chosen. For making subsets: see `makeSubset`.

Plot by groups...: Clicking this button a dialog box appears which allows to select a grouping variable. Clicking the OK button in this dialog box a further box appears where the levels of the grouping variable can be selected. Either subsets or groups can be chosen.

Deselect: By clicking this button the selected subsets or groups will be deselected.

Details

ID-numbers are the values of the identification variable of the data set. The identification variable can be changed in `MakeDASData`. It can be opened by clicking the DASplusR icons

`Edit -> Make DAS Data`.

Note

This function assumes a data set containing at least 3 numerical variables to be active.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`ternaryplotDAS`, `ternaryplotDetails`, `identify`

`XyplotGUI` *GUI (Graphical User Interface) for xy-plot of the active data set in DASplusR*

Description

This function opens a dialog window where all settings for the creation of an xyplot are specified.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Diagrams -> Xyplot.

## R Function Usage:
  XyplotGUI()
```

GUI Elements

Plot title: The main title of the xyplot.

X-variable: Select a variable to be used for the x-axis.

Y-variable: Select a variable to be used for the y-axis.

Log-scale: If clicked the x- resp. y-variable should be log-transformed.

Plot by subsets...: Clicking this button a dialog box appears which allows selecting subsets. Either subsets or groups can be chosen.

Plot by groups...: Clicking this button a dialog box appears which allows to select a grouping variable. Clicking the OK button in this dialog box a further box appears where the levels of the grouping variable can be selected. Either subsets or groups can be chosen.

Deselect: By clicking this button the selected subsets or groups will be deselected.

x-Axis: Clicking this button opens a dialog box allowing to redefine specifications for the x-axis. See `GUIElements`.

y-Axis: Clicking this button opens a dialog box allowing to redefine specifications for the y-axis. See `GUIElements`.

Symbols: Clicking this button opens a dialog box allowing to redefine specifications for the symbols to be used. See `GUIElements`.

Legend: Clicking this button opens a dialog box allowing to redefine specifications for a possible legend. See `GUIElements`.

Additionals: Clicking this button opens a dialog box allowing to redefine specifications for additional demands like identifying points, adding lines, correlations, etc. See `GUIElements`.

Generate new plot: Checking this box implies the plotting in a new graphic windows, otherwise the previous graphic is overplotted.

Note

This function assumes that a data set containing at least 2 numerical variables, is active.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`xyplotDAS`, `xyplotDetails`, `identify`, `GUIElements`

A.2 Executable Functions

These help files can be called by typing `?DesiredHelpFile` in the R console, e.g. `?subsetOperationsDAS` These functions are usually called by the GUI functions

but they may also be executed independently which is typically used in batch execution.

`boxplotDAS` *Boxplot in DASplusR*

Description

Produce box-and-whisker plot(s) of the given (grouped) values.

Usage

```
boxplotDAS (datname, varnames, selection = NULL, levels = NULL,
  names = NULL, main = NULL, logV = FALSE, notch = NULL,
  horizontal = NULL, range = NULL, width = NULL, add = NULL,
  varwidth = NULL, outline = NULL, boxwex = NULL, staplewex = NULL,
  outwex = NULL, plot = NULL, col = NULL, at = NULL, nbp = NULL,
  id = NULL, idVal = NULL, idPos = NULL, batchmode = TRUE, ...)
```

Arguments

`datname`	Name of the data set.
`varnames`	The variable(s) from which the box plot(s) is(are) produced.
`selection`	The entry could be either the word "Subsets" or a vector of 2 elements containing the word "Groups" and the grouping variable.
`levels`	The desired level(s) of the grouping variable or the subset(s) for the box plot.
`names`	Labels which will be printed under each boxplot.
`main`	Main title of the plot.
`logV`	If TRUE the box plot from the logarithmic values of the variable(s) would be produced.
`notch`	If 'notch' is TRUE, a notch is drawn in each side of the boxes. If the notches of two plots do not overlap this is 'strong evidence' that the two medians differ (Chambers et al., 1983, p. 62). See 'boxplot.stats' for the calculations used.
`horizontal`	Logical indicating if the boxplots should be horizontal; default FALSE means vertical boxes.

`range`	This determines how far the plot whiskers extend out from the box. If 'range' is positive, the whiskers extend to the most extreme data point which is no more than 'range' times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
`width`	A vector giving the relative widths of the boxes making up the plot.
`add`	Logical, if TRUE, add boxplot to current plot.
`varwidth`	If 'varwidth' is TRUE, the boxes are drawn with widths proportional to the square-roots of the number of observations in the groups.
`outline`	If 'outline' is not true, the outliers are not drawn (as points whereas S+ uses lines).
`boxwex`	A scale factor to be applied to all boxes. When there are only a few groups, the appearance of the plot can be improved by making the boxes narrower.
`staplewex`	Staple line width expansion, proportional to box width.
`outwex`	Outlier line width expansion, proportional to box width.
`plot`	If TRUE (default) then the boxplot(s) is(are) produced. If not, the summary of the boxplot(s) is(are) based on, is stored in `summary_boxplot` on the R console.
`col`	If 'col' is non-null it is assumed to contain colors to be used to color the bodies of the box plots. By default they are in the background color.
`at`	Numeric vector giving the locations where the boxplots should be drawn, particularly when 'add = TRUE'; defaults to '1:n' where 'n' is the number of boxes.
`nbp`	A vector containing the numbers of the boxplots were the identified outliers belong to. It is the x value in function `identify`.
`id`	A vector containing the ID numbers of the identified outliers. If there are no ID's the rownames would be used. It is labels in function `identify`.
`idVal`	A vector containing the values of the identified outliers. It is the y value in function `identify`.
`idPos`	A vector containing the position of `id`. See `atpen` in function `identify`.

batchmode	If FALSE the function would be called from BoxplotGUI. If TRUE a user calls the function directly.
...	Further graphical parameters, see par.

Note

The vectors nbp, id, idVal, idPos must all have the same length.

The lengths of width and at must be the same as the number of plotted boxplots. For example: You have two variables and two levels => 4 boxplots => length of width must be 4. See the last example below.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

BoxplotGUI, boxplotDetails, boxplot, identify

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
  names=c("Al [mg/kg]","Fe [mg/kg]"))

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
  names=c("Al [mg/kg]","Fe [mg/kg]"),logV=TRUE,horizontal=TRUE)

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
  names=c("Al [mg/kg]","Fe [mg/kg]"),logV=TRUE,horizontal=TRUE,width=c(1,2),
  col="red",at=c(2,1))

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
  selection=c("Groups","COUN"),levels=c("RUS","NOR"),
  names=c("Al [mg/kg]\nRUS (290)","Al [mg/kg]\nNOR (128)",
    "Fe [mg/kg]\nRUS (290)","Fe [mg/kg]\nNOR (128)"),nbp=c(1,1,1,2,3,3,4),
  id=c("55","440","739","574","69","177","212"),
  idVal=c(63000,71000,85900,47900,61600,42600,49000),idPos=c(4,1,4,2,4,2,4))

boxplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
```

```
  selection=c("Groups","COUN"),levels=c("RUS","NOR"),
  names=c("Al [mg/kg]\nRUS (290)","Al [mg/kg]\nNOR (128)",
    "Fe [mg/kg]\nRUS (290)","Fe [mg/kg]\nNOR (128)"),width=c(1,2,1,2))
```

`deleteSubsetsDAS` *Function for deleting subsets of the active data set in DASplusR*

Description

This function deletes subsets.

Usage

```
deleteSubsetsDAS (datname, subset)
```

Arguments

`datname`	name of the data set
`subset`	name of the subset to be deleted. Writing the word `all` instead of the name, all subsets from the data set "datname" are deleted

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`DeleteSubsetsGUI`, `makeSubset`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subOne",
  ind=c(2,44,95,131,159,164,188,210,224,230,231,234,268,273,311,318,327,
  334,354,359,369,386,428,431,451,465,471,508,509,512,542,546,560,566),
  form="Al < 4000",desc=NA,overwrite=TRUE)
```

```
makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subTwo",
  ind=c(44,84,95,98,179,210,224,230,231,273,302,327,354,369,372,386,431,
  477,512,542,546,587),form="Fe < 6000",desc=NA,overwrite=TRUE)

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subThree",
  ind=c(47,58,134,151,320,346,563,569),form=" Al >50000 ",desc=NA,
  overwrite=TRUE)

deleteSubsetsDAS("KOLA95_Chor","subOne")
deleteSubsetsDAS("KOLA95_Chor","all")
```

`densityDAS` *Density plot in DASplusR*

Description

This function plots densities of the given (grouped) values.

Usage

```
densityDAS (datname, varnames, selection = NULL, levels = NULL,
    xlab = "", ylab = "Density", lty = "solid", col = "red",
    lwd = 1, log = "", xticks = TRUE, gridx = 1, gridxlty = "solid",
    gridxcol = "red", gridxwidth = 1, yticks = TRUE, gridy = 1,
    gridylty = "dashed", gridycol = "red", gridywidth = 1, xlim = NULL,
    margin = c(5.1, 4.1, 4.1, 2.1), newPlot = TRUE, legend = "none",
    legend_x = 0.78, legend_y = 0.9, legend_ex = 1, batchmode = TRUE, ...)
```

Arguments

`datname`	Character; the name of the data object.
`varnames`	Character; the variable(s) from which the density(ies) is(are) produced.
`selection`	Character; if not NULL (default) the entry could be either the word "`Subsets`" or a vector of 2 elements containing the words "`Groups`" and the grouping variable.
`levels`	Character vector of the names of the subsets or the levels of the grouping variable according to the specified 'selection' which should be plotted.

`xlab`	Label of the x-axis.
`ylab`	Label of the y-axis.
`lty`	The line type, see `par`.
`col`	Vector of colors to be used for each subset resp. grouping level.
`lwd`	The line width, see `par`.
`log`	A character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic.
`xticks`	A logical indicating whether tickmarks should be put at the x-axis, see `par`.
`gridx`	A logical indicating whether a grid should be put in x-direction, see `grid`.
`gridxlty`	Line type of the grid in x-direction, see `grid`.
`gridxcol`	Color of the grid in x-direction, see `grid`.
`gridxwidth`	Width of the lines of the grid in x-direction, see `grid`.
`yticks`	A logical indicating whether tickmarks should be put at the y-axis, see `par`.
`gridy`	A logical indicating whether a grid should be put in y-direction, see `grid`.
`gridylty`	Line type of the grid in y-direction, see `grid`.
`gridycol`	Color of the grid in y-direction, see `grid`.
`gridywidth`	Width of the lines of the grid in y-direction, see `grid`.
`xlim`	The x limits (x1, x2) of the plot. Note that x1 $>$ x2 is allowed and leads to a "reversed axis".
`margin`	Numerical vector of length 4 defining the margins of the plot, see `par`.
`newPlot`	TRUE (default) if new plot should be drawn, otherwise the previous one is overwritten.
`legend`	Character; value indicating if legend should be added, see Details.
`legend_x`	x-coordinate of left-upper corner of the legend.
`legend_y`	y-coordinate of left-upper corner of the legend.
`legend_ex`	Expansion of size of the legend (default = 1).

batchmode	If FALSE the function would be called from DensityGUI. If TRUE a user calls the function directly.
...	Further graphical parameters, see par.

Details

legend can be "none", "by coordinates" or "by clicks".

"none": No legend is plotted.

"by coordinates": The legend is plotted at the coordinates legend_x and legend_y. The coordinates define the left upper-corner of the legend.

The size of the legend can defined by the parameter legend_ex.

"by clicks": The position and size of the legend will be defined by two clicks in the graphic at two opposite corners of the frame of the legend.

The function call in the Script Window and the Output Window of the DAS+R surface, the coordinates of the left upper-corner together with legend_ex are used.

If batchmode is TRUE, the legend can only be "none" or "by coordinates".

Note

If a title for the plot is desired, just add main="desired title" to the arguments, as shown in the last two examples. (Other graphical parameters can be handled in the same way, see ...-argument.)

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

DensityGUI, density, GUIElements

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

densityDAS(datname="KOLA95_Chor",varnames=c("Al"))
```

```
densityDAS(datname="KOLA95_Chor",varnames=c("Al"),main="Density with legend",
  legend="by coordinates",legend_x=0.7,legend_y=0.9,legend_ex=2)

densityDAS(datname="KOLA95_Chor",varnames=c("Al","Fe"),
  selection=c("Groups","COUN"),levels=c("FIN","NOR"),
  lty=c("solid","dashed","dotted","dotdash"),
  col=c("red","green3","blue","magenta"), lwd=c(1,1,1,1),
  main="Density of Al and Fe with groups", cex.main=1.5, cex.sub=1,
  log="", xticks=TRUE,gridx=1,gridxlty="solid",gridxcol="red",gridxwidth=1,
  xlim=NULL, margin=c(5.1,4.1,4.1,2.1), newPlot=TRUE ,legend="by coordinates",
  legend_x=0.7,legend_y=0.9,legend_ex=1)
```

`DeselectAllSubsets`

Deselecting all subsets of the active data set in DASplusR

Description

This function deselects all variables.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Select -> Deselect All Subsets.

## R Function Usage:
  DeselectAllSubsets()
```

Note

This function calls `selectVarSub` with `vars` containing no subsets (`vars=0`).

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`SelectVarSub`, `SelectSubsetsGUI`, `SelectVariables`, `SelectAllSubsets`

`histDAS` *Histogram including density trace, scatterplot, boxplot*

Description

The function 'histDAS' plots a histogram of the given data values, optionally with a density trace, a scatter- and a boxplot.

Usage

```
histDAS (datname, varname, selection = NULL, levels = NULL, scatter = TRUE,
  box = TRUE, dens = FALSE, P.plot = TRUE, P.main = "", P.cex.main = 1.2,
  P.sub = NULL, P.cex.sub = 1, P.xlab = NULL, P.ylab = "Frequency",
  P.ann = par("ann"), P.axes = TRUE, P.frame.plot = P.axes, P.xlim = NULL,
  P.margin = c(5.1, 4.1, 4.1, 2.1), B.range = 1.5, B.notch = FALSE,
  B.outline = TRUE, B.border = par("fg"), B.col = NULL, B.pch = par("pch"),
  B.cex = 1, H.breaks = "Sturges", H.freq = TRUE,
  H.include.lowest = TRUE, H.right = TRUE, H.density = NULL, H.angle = 45,
  H.col = NULL, H.border = NULL, H.labels = FALSE, S.pch = ".",
  S.col = "black", S.bg = NA, S.cex = 1, D.lty = "solid", D.col = "red",
  D.lwd = 1, batchmode = TRUE)
```

Arguments

`datname`	The name of the data object.
`varname`	The variable name from which the histogram is produced.
`selection`	If not NULL (default) the entry could be either the word "Subset" or a vector of 2 elements containing the word "Group" and the grouping variable.
`level`	Character; name of the subset or the level of the grouping variable according to the specified 'selection'.
`scatter`	Logical; if TRUE, a scatterplot is added to the histogram.
`box`	Logical; if TRUE, a boxplot is added to the histogram.
`dens`	Logical; if TRUE, a density trace is added to the histogram.

`P.plot`	Logical; if FALSE no plot is produced, just the summary is returned.
`P.main`	Main title for the plot.
`P.cex.main`	Expansion of character size of the title.
`P.sub`	Subtitle for the plot.
`P.cex.sub`	Expansion of character size of the subtitle.
`P.xlab`	A label for the x axis.
`P.ylab`	A label for the y axis.
`P.ann`	A logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot.
`P.axes`	A logical value indicating whether axes should be drawn on the plot.
`P.frame.plot`	A logical value indicating whether a box should be drawn around the plot.
`P.xlim`	The x limits (x1, x2) of the plot. Note that x1 $>$ x2 is allowed and leads to a "reversed axis".
`P.margin`	Numerical vector of length 4 defining the margins of the plot, see `par`.
`B.range`	This determines how far the plot whiskers extend out from the box. If range is positive, the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
`B.notch`	Logical; if TRUE a notch is drawn in each side of the box.
`B.outline`	If outline is not true, the boxplot lines are not drawn.
`B.border`	Color for the outlines of the boxplot.
`B.col`	Color of the body of the boxplot.
`B.pch`	Either a 'character' or an integer code for a graphic symbol, which determine the appearance of the values, which lie beyond the whiskers. See also `points`.
`B.cex`	Expansion of the character determined by 'B.pch'; a numerical vector. See also `points`.
`H.breaks`	One of:

- a vector giving the breakpoints between histogram cells,
- a single number giving the number of cells for the histogram,
- a character string naming an algorithm to compute the number of cells (see Details),
- a function to compute the number of cells.

`H.freq` Logical; if TRUE, the histogram graphic is a representation of frequencies, the counts component of the result; if FALSE, relative frequencies ('probabilities'), component density, are plotted.

`H.include.lowest`
Logical; if TRUE, an x[i] equal to the breaks value will be included in the first (or last, for H.right = FALSE) bar. This will be ignored (with a warning) unless H.breaks is a vector.

`H.right` Logical; if TRUE, the histogram cells are right-closed (left open) intervals.

`H.density` The density of shading lines, in lines per inch. The default value of NULL means that no shading lines are drawn. Non-positive values of density also inhibit the drawing of shading lines.

`H.angle` The slope of shading lines, given as an angle in degrees (counter-clockwise).

`H.col` A color to be used to fill the bars. The default of NULL yields unfilled bars.

`H.border` The color of the border around the bars. The default is to use the standard foreground color.

`H.labels` Logical or character. Additionally draw labels on top of bars, if not FALSE; see `plot.histogram`.

`S.pch` Either a 'character' or an integer code for a graphic symbol, which determine the appearance of the dots in the scatterplot. See also `points`.

`S.col` Color for the characters determined in 'S.pch', see `par`.

`S.bg` Background ("fill") color for open plot symbols determined by 'S.pch'. See also `points`.

`S.cex` Expansion of the character determined by 'S.pch'; a numerical vector. See also `points`.

`D.lty` The line type of the density, see `par`.

`D.col` The color of the density.

`D.lwd` The line width, see `par`.

`batchmode` If FALSE the function would be called from `HistGUI`. If TRUE a user calls the function directly.

Details

The definition of "histogram" differs by source (with country-specific biases). R's default with equi-spaced breaks (also the default) is to plot the counts in the cells defined by 'breaks'. Thus the height of a rectangle is proportional to the number of points falling into the cell, as is the area 'provided' the breaks are equally-spaced.

The default with non-equi-spaced breaks is to give a plot of area one, in which the 'area' of the rectangles is the fraction of the data points falling in the cells.

If 'right = TRUE' (default), the histogram cells are intervals of the form '(a, b]', i.e., they include their right-hand endpoint, but not their left one, with the exception of the first cell when 'include.lowest' is TRUE.

For 'right = FALSE', the intervals are of the form '[a, b)', and 'include.lowest' really has the meaning of 'include highest'. A numerical tolerance of 1e-7 times the range of the breaks is applied when counting entries on the edges of bins.

The default for 'breaks' is 'Sturges': see `nclass.Sturges`. Other names for which algorithms are supplied are 'Scott' and 'FD' / 'Friedman-Diaconis' (with corresponding functions 'nclass.scott' and 'nclass.FD'). Case is ignored and partial matching is used. Alternatively, a function can be supplied which will compute the intended number of breaks as a function of 'x'.

Value

An invisible list consisting of 'H', an object of class 'histogram' and a further list 'B', actually representing the values returned by 'hist' and 'boxplot'.

`H$breaks` The n+1 cell boundaries (= 'breaks' if that was a vector).

`H$counts` n integers; for each cell, the number of 'x[]' inside.

`H$density` Values $f(x[i])$, as estimated density values. If 'all(diff(breaks) == 1)', they are the relative frequencies 'counts/n' and in general satisfy sum[i; f(x[i]) (b[i+1]-b[i])] = 1, where b[i] = 'breaks[i]'.

`H$intensities`
Same as 'density'. Deprecated, but retained for compatibility.

`H$mids`	The n cell midpoints.
`H$xname`	A character string with the actual 'x' argument name.
`B$stats`	A matrix, each column contains the extreme of the lower whisker, the lower hinge, the median, the upper hinge and the extreme of the upper whisker for one group/plot.
`B$n`	A vector with the number of observations in each group.
`B$conf`	A matrix where each column contains the lower and upper extremes of the notch.
`B$out`	The values of any data points which lie beyond the extremes of the whiskers.
`B$group`	A vector of the same length as 'out' whose elements indicate which group the outlier belongs to.
`B$names`	A vector of names for the groups.

Note

For this function just one variable and one subset or group element can be used. If more are selected, just the first one is used.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Martin Riedler, Stefan Wohlmuth, Andreas Zainzinger

See Also

`GUIElements`, `HistGUI`, `hist`, `boxplot`, `scatterplot`, `density`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

histDAS(datname="KOLA95_Chor", varname=c("Fe"), selection=c("Group","COUN"),
  level=c("NOR"),dens=T,scatter=T,box=T)

histDAS (datname="KOLA95_Chor", varname=c("Fe"), scatter = TRUE,
  box = FALSE, dens = TRUE, P.plot = TRUE, P.main = "", P.cex.main = 1.2,
  P.sub = NULL, P.cex.sub = 1, P.xlab = NULL, P.ylab = "Frequency",
```

```
P.ann = par("ann"), P.axes = TRUE, P.frame.plot = P.axes, P.xlim = NULL,
P.margin = c(5.1, 4.1, 4.1, 2.1), H.breaks = "Sturges", H.freq = TRUE,
H.include.lowest = TRUE, H.right = TRUE, H.density = NULL, H.angle = 45,
H.col = NULL, H.border = NULL, H.labels = FALSE, S.pch = ".",
S.col = "black", S.bg = NA, S.cex = 1, D.lty = "solid", D.col = "red",
D.lwd = 1)
```

`makeNewDataSubDAS` *Function for generating a new data set from a subset of the active data set in DASplusR.*

Description

This function generates a new data set with the same samples as in the chosen subset.

Usage

```
makeNewDataSubDAS (dat, datname, sub, datnew)
```

Arguments

`dat`	Data object of class 'DASData' or 'data.frame'.
`datname`	Name of the data object.
`sub`	The subset used for generating a new data set.
`datnew`	Name of the new data set.

Details

The new data set has the same samples as the subset. It inherits all attributes from the original data set, also the subsets except the one the data set is created from. The subsets of course just contain samples which are included in the data set. Therefore the cardinality of the subsets changes.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`MakeNewDataGUI`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subOne",
  ind=c(2,44,95,131,159,164,188,210,224,230,231,234,268,273,311,318,327,
  334,354,359,369,386,428,431,451,465,471,508,509,512,542,546,560,566),
  form="Al < 4000",desc=NA,overwrite=TRUE)

makeNewDataSubDAS(dat=KOLA95_Chor,datname="KOLA95_Chor",sub="subOne",
  datnew="KOLA95_Chor_subOne")
```

`renameSubsetDAS` *Function for renaming a subset*

Description

This function renames a subset.

Usage

```
renameSubsetDAS (datname, subold, subnew)
```

Arguments

`datname`	name of the data set
`subold`	subset to be renamed
`subnew`	new name of the subset

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`renameSubsetGUI`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subOne",
  ind=c(2,44,95,131,159,164,188,210,224,230,231,234,268,273,311,318,327,
  334,354,359,369,386,428,431,451,465,471,508,509,512,542,546,560,566),
  form="Al < 4000",desc=NA,overwrite=TRUE)

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subTwo",
  ind=c(44,84,95,98,179,210,224,230,231,273,302,327,354,369,372,386,431,
  477,512,542,546,587),form="Fe < 6000",desc=NA,overwrite=TRUE)

renameSubsetDAS("KOLA95_Chor","subOne","subOne_new")
```

`SelectAllSubsets` *Selecting all subsets of the active data set in DASplusR*

Description

This function sets all subsets as selected.

Usage

```
## DAS+R GUI:
  It can be called by clicking the DASplusR icons

  Select -> All Subsets.

## R Function Usage:
  SelectAllSubsets()
```

Note

This function calls `selectVarSub` with `vars` contains all existing subsets.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`SelectVarSub`, `SelectSubsetsGUI`, `SelectVariables`, `DeselectAllSubsets`

`SelectVarSub` *Function for selecting elements of the active data set in DASplusR*

Description

This function opens a dialog window for selecting elements (e.g. subsets, variables, ...) therewith afterwards just a reduced amount of elements are used.

Usage

```
SelectVarSub (VarSubList = VarSubList, VarSubSelList = VarSubSelList,
  widget_title = NULL, box_title = NULL, nitems = 20)
```

Arguments

`VarSubList`	Names of all entries of variableListBox.
`VarSubSelList`	Names of initially selected entries of variableListBox.
`widget_title`	Title of the window.
`box_title`	Title of variableListBox.
`nitems`	Maximum number of entries displayed.

GUI Elements

Subsets: List of existing subsets or variables.

Select all: Button for selecting all subsets at once.

Deselect all: Button for deselecting all subsets.

Details

This function usually is called from another function which provides the necessary information about `VarSubList, VarSubSelList`, etc.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`selectVarSub`, `SelectSubsetsGUI`, `SelectVariables`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

SelectVarSub (VarSubList = colnames(KOLA95_Chor),
  VarSubSelList = c("Al","Fe","As"),
  widget_title = "Select variables for KOLA95_Chor",
  box_title = "List of Variables", nitems = 20)
```

`selectVarSub` *Selecting elements of the active data set in DASplusR*

Description

This function sets the elements (e.g. subsets,variables, ...) given by `vars` as selected.

Usage

```
selectVarSub (data, datname = NULL, vars, objects = c("variables",
   "subsets"))
```

Arguments

`data`	Data object of class 'DASData' or 'data.frame'.
`datname`	Name of the data set.
`vars`	List of elements to be selected.
`objects`	Keyword for selecting.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`SelectVarSub`, `SelectSubsetsGUI`, `SelectVariables`

`sortDAS` *Function for sorting elements of the active data set in DASplusR.*

Description

This function is for sorting elements (e.g. subsets or variables).

Usage

```
sortDAS (datname = NULL, sortmethod, sortvec, object = NULL)
```

Arguments

`datname`	Name of the data set.
`sortmethod`	The sorting method. It can be either "manually" or "alphabetically".
`sortvec`	The character vector to be sorted. In case of "manually", the already ordered vector has to be entered.
`objects`	Keyword for selecting: 'subsets' and 'variables' are possible.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

SortSubsetsGUI

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subOne",
  ind=c(2,44,95,131,159,164,188,210,224,230,231,234,268,273,311,318,327,
  334,354,359,369,386,428,431,451,465,471,508,509,512,542,546,560,566),
  form="Al < 4000",desc=NA,overwrite=TRUE)

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subTwo",
  ind=c(44,84,95,98,179,210,224,230,231,273,302,327,354,369,372,386,431,
  477,512,542,546,587),form="Fe < 6000",desc=NA,overwrite=TRUE)

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subThree",
  ind=c(47,58,134,151,320,346,563,569),form=" Al >50000 ",desc=NA,
  overwrite=TRUE)

sortDAS(datname="KOLA95_Chor",sortmethod="alphabetically",
  sortvec=c("subOne","subTwo","subThree"),object="subsets")

sortDAS(datname="KOLA95_Chor",sortmethod="manually",
  sortvec=c("subThree","subTwo","subOne"),object="subsets")
```

---

subsetOperationsDAS

Function for making a new subset from existing subsets of the active data set in DASplusR

Description

This function produces a subset by different subset operations as building the complement of a subset, or the union, intersection, or the difference of 2 subsets.

Usage

```
subsetOperationsDAS (datname, sub1, sub2 = NULL, newsub, operation,
  desc = NA)
```

Dialog Elements

datname: Name of the data set.

sub1: First subset used for the operations.

sub2: Second subset used for the operations.

newsub: Name of the new subset. It must start with a letter.

operation: Operation used for creating a new subset.

desc: Description of the new subset.

Details

`sub2` is only necessary for the operations: "union", "intersection", "difference".
`operation` can be: "complement", "union", "intersection" or "difference".
`complement` ... complement of sub1.
`union` ... union of first and the second subset. See `union`.
`intersection` ... intersection of the first and the second subset. See `intersect`.
`difference` ... the set of elements in sub1, but not in sub2. See `setdiff`.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`SubsetOperationsGUI`, `makeSubset`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subOne",
  ind=c(2,44,95,131,159,164,188,210,224,230,231,234,268,273,311,318,327,
  334,354,359,369,386,428,431,451,465,471,508,509,512,542,546,560,566),
  form="Al < 4000",desc=NA,overwrite=TRUE)
```

```
makeSubsetDAS(KOLA95_Chor,datname="KOLA95_Chor",subname="subTwo",
  ind=c(44,84,95,98,179,210,224,230,231,273,302,327,354,369,372,386,431,
  477,512,542,546,587),form="Fe < 6000",desc=NA,overwrite=TRUE)

subsetOperationsDAS(datname="KOLA95_Chor",sub1="subOne",newsub="subC",
  operation="complement")
subsetOperationsDAS(datname="KOLA95_Chor",sub1="subOne",sub2="subTwo",
  newsub="subU",operation="union")
subsetOperationsDAS(datname="KOLA95_Chor",sub1="subOne",sub2="subTwo",
  newsub="subI",operation="intersection")
subsetOperationsDAS(datname="KOLA95_Chor",sub1="subOne",sub2="subTwo",
  newsub="subD",operation="difference")
subsetOperationsDAS(datname="KOLA95_Chor",sub1="subTwo",sub2="subOne",
  newsub="subD2",operation="difference")
```

`ternaryplotDAS` *Ternaryplot in DASplusR*

Description

Produce a ternaryplot of the given (grouped) samples.

Usage

```
ternaryplotDAS (datname, varnames, main = NULL, Selection = NULL,
  levels = NULL, names = NA, variableLabel = NA, position = "corner",
  color = "black", symbolset = "default", cex = 1,
  sampleIdentification = "none", idColor = "black",
  identificationGrid = TRUE, gridColor = "gray60",
  gridLabelsColor = "darkgray", borderColor = "black", legend = TRUE,
  legend_x = 0.78, legend_y = 0.9, legend_ex = par("cex"),
  newPlot = TRUE, rowSumsScale = 1, id = NULL, idPos = NULL,
  rowPos = NULL, batchmode = TRUE, ...)
```

Arguments

`datname`	Name of the data set.
`varnames`	The three variables from which the ternaryplot is produced.
`main`	Main title of the plot.

`Selection` The entry could be either the word "Subsets", or a vector of two elements containing the word "Groups" and the grouping variable.

`levels` The desired level(s) of the grouping variable or the subset(s) for the plot.

`names` Names of the plotted groups/subsets of the data set.

`variableLabel` Variable labels (default the names of the selected variables including the unit of the variables).

`position` Position of variable labels.

`color` Color of the variable labels.

`symbolset` The used symbol set. for symbol set see: `makeSymbolSet`.

`cex` A numerical value giving the amount by which plotting text and symbols should be scaled relatively to the default.

`sampleIdentification` Optional labels to be plotted below the plot symbols.

`idColor` Color of sample identification.

`identificationGrid` If `TRUE`, a grid is plotted.

`gridColor` Grid color.

`gridLabelsColor` Color of the grid labels.

`borderColor` Color of the triangle border.

`legend` If `TRUE` a legend is plotted.

`legend_x` x-coordinate of the left-upper corner of the legend.

`legend_y` y-coordinate of the left-upper corner of the legend.

`legend_ex` The scale of the legend (relative to the default).

`newPlot` If `TRUE` the plot will appear on a new graphics page.

`rowSumsScale` Row sums scale to be used.

`id` A vector containing the ID's of the identified values. If there are no ID's, the row names would be used. They correspond to the labels in function `identify`.

`idPos` A vector containing the position of `id`. See `atpen` in function `identify`.

`rowPos`	Is the row position of the identified values in the data matrix.
`batchmode`	If FALSE the function would be called from `TernaryplotGUI`. If TRUE a user calls the function directly.
`...`	Further graphical parameters, see `par`.

Details

`position` can be `"corner"`, `"edge"` or `"none"`.

`sampleIdentification`: The values of the identification variable of the data set are used. If no identification variable is specified, the row names of the data set will be used.

`Legend` only appears if subsets or groups are selected.

The vectors `id` and `idPos` must be of the same length.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`TernaryplotGUI`, `ternaryplotDetails`, `ternary`, `identify`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

ternaryplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe","Mg"),
  variablesLabel=c("Al [mg/kg]","Fe [mg/kg]","Mg [mg/kg]"))

ternaryplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe","Mg"),
  main="main title",variablesLabel=c("Al [mg/kg]","Fe [mg/kg]","Mg [mg/kg]"),
  position="edge",color="green")

ternaryplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe","Mg"),
  variablesLabel=c("Al [mg/kg]","Fe [mg/kg]","Mg [mg/kg]"),
  sampleIdentification="ID Number",idColor="black",id=c("2","9","34"),
  idPos=c(1,1,1),rowPos=c(2,8,28))

ternaryplotDAS(datname="KOLA95_Chor",varnames=c("Al","Fe","Mg"),
```

```
Selection=c("Groups","COUN"),levels=c("RUS","NOR"),
names=c("RUS (290)","NOR (128)"),
variablesLabel=c("Al [mg/kg]","Fe [mg/kg]","Mg [mg/kg]"),
position="corner",color="black",symbolset="default",cex=1,
sampleIdentification="none",idColor="black",identificationGrid=TRUE,
gridColor="gray60",gridLabelsColor="darkgray",borderColor="black",
legend=TRUE,legend_x=0.78,legend_y=0.9,legend_ex=1,newPlot=TRUE,
rowSumsScale=1)
```

`xyplotDAS` *Xy-plot in DASplusR*

Description

This function plots a xy-plot by assembling all kind of specifications from different menus and by calling 'scatterplot'.

Usage

```
xyplotDAS (datname, varnames, selection = NULL, levels = NULL,
  symb = 8, col = "red", size = 1, cex.main = par("cex.main"),
  sub = FALSE, cex.sub = par("cex.sub"), margin = par("mar"),
  xlab = deparse(substitute(x)), xlty = "solid", xlwd = 1,
  xcol = NULL, xlas = par("las"), xticks = TRUE,
  ylab = deparse(substitute(y)), ylty = "solid", ylwd = 1, ycol = NULL,
  ylas = par("las"), yticks = TRUE, gridx = FALSE,
  gridxcol = "lightgray", gridxlty = "dotted", gridxwidth = 1,
  gridy = FALSE, gridycol = "lightgray", gridylty = "dotted",
  gridywidth = 1, lwd = 1, cex.lab = par("cex.lab"), cex.font = NA,
  jitter = list(), frame.plot = TRUE, asp = NA, log = NULL,
  ellipse = FALSE, robust = FALSE, reg.line = NULL, smooth = NULL,
  span = 0.5, labels = FALSE, legend = FALSE, legend_x = 0.78,
  legend_y = 0.9, legend_ex = 1, reset.par = TRUE, newPlot = TRUE,
  batchmode = TRUE, ...)
```

Arguments

`datname`	The name of the data object.
`varnames`	Character vector of the names of the two variables to be plotted.

`selection`	If not NULL (default) the entry could be either the word "Subsets", or a vector of two elements containing the word "Groups" and the grouping variable.
`levels`	Character vector of the names of the subsets or the levels of the grouping variable according to the specified 'selection', which should be plotted.
`symb`	Vector of integers specifying the symbols to be used for each subset resp. grouping level.
`col`	Vector of colors to be used for each subset resp. grouping level.
`size`	Vector of numerical values specifying the size to be used for each subset resp. grouping level.
`cex_main`	Expansion of character size of the title (default = 1).
`sub`	Character string to be used as subtitle (see also `plot.default`).
`cex_sub`	Expansion of character size of the subtitle (default = 1).
`margin`	Numerical vector of length 4 defining the margins of the plot, see `par`.
`xlab`	Label of the x-axis.
`xlty`	Line type of the x-axis, see `par`.
`xlwd`	Line width of the x-axis, see `par`.
`xcol`	Color of the x-axis, see `par`.
`xlas`	Label style of the x-axis, see `par`.
`xticks`	A logical indicating whether tickmarks should be put at the x-axis, see `par`.
`ylab`	Label of the y-axis.
`ylty`	Line type of the y-axis, see `par`.
`ylwd`	Line width of the y-axis, see `par`.
`ycol`	Color of the y-axis, see `par`.
`ylas`	Label style of the y-axis, see `par`.
`yticks`	A logical indicating whether tickmarks should be put at the y-axis, see `par`.
`gridx`	A logical indicating whether a grid should be put in x-direction, see `grid`.
`gridxcol`	Color of the grid in x-direction, see `grid`.

`gridxlty`	Line type of the grid in x-direction, see `grid`.
`gridxwidth`	Width of the lines of the grid in x-direction, see `grid`.
`gridy`	A logical indicating whether a grid should be put in y-direction, see `grid`.
`gridycol`	Color of the grid in y-direction, see `grid`.
`gridylty`	Line type of the grid in y-direction, see `grid`.
`gridywidth`	Width of the lines of the grid in y-direction, see `grid`.
`lwd`	The line width, see `par`.
`cex.lab`	Expansion of character size of the labels (default = 1).
`cex.font`	Font used for the labels.
`jitter`	List of 2 values containing the size of jittering for the x- and y-values, resp., default is 0.
`frame.plot`	A logical indicating whether a box should be drawn around the plot.
`asp`	The y/x aspect ratio, see `plot.window`.
`log`	A character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic.
`ellipse`	If TRUE a tolerance ellipse is drawn.
`robust`	If TRUE use the cov.trob function in the MASS package to calculate the center and covariance matrix for the data ellipse.
`reg.line`	If TRUE a regression line is added to the plot.
`smooth`	If TRUE a lowess nonparametric regression line is is added to the plot.
`span`	Span for the lowess smooth.
`labels`	If not FALSE, a vector of point labels to be used to identify points on the plot.
`legend`	Character; value indicating if legend should be added, see Details.
`legend_x`	x-coordinate of left-upper corner of legend.
`legend_y`	y-coordinate of left-upper corner of legend.
`legend_ex`	Expansion of size of the legend (default = 1).
`reset.par`	Reset the graphics parameter.

`newPlot`	TRUE (default) if new plot should be drawn, otherwise the previous one is overwritten.
`batchmode:`	If FALSE the function would be called from `XyplotGUI`. If TRUE a user calls the function directly.
`...:`	Further graphical parameters, see `par`.

Details

`legend` can be `"none"`, `"by coordinates"` or `"by clicks"`.

`"none"`: No legend is plotted.

`"by coordinates"`: The legend is plotted at the coordinates `legend_x` and `legend_y`. The coordinates define the left upper-corner of the legend.

The size of the legend can defined by the parameter `legend_ex`.

`"by clicks"`: The position and size of the legend will be defined by two clicks in the graphic at two opposite corners of the frame of the legend.

The function call in the Script Window and the Output Window of the DAS+R surface, the coordinates of the left upper-corner together with `legend_ex` are used.

If batchmode is `TRUE`, the legend can only be "none" or "by coordinates".

Value

The function returns TRUE or FALSE regarding the success of the workflow.

Note

If a title for the plot is desired, just add `main="desired title"` to the arguments as shown in the last two examples.

Author(s)

Rudolf Dutter ⟨R.Dutter@tuwien.ac.at⟩, Andreas Zainzinger

See Also

`XyplotGUI`, `GUIElements`, `scatterplot`

Examples

```
require(DASplusR)
data("KOLA95_Chor")
activeDataSet("KOLA95_Chor")

xyplotDAS(datname="KOLA95_Chor",varnames=c("Al","CaO"), symb=c(8),
  col=c("red"), cex=c(1), legend=FALSE, cex.main=1.20000004768372,
  cex.sub=1, xlab="Al [mg/kg]", ylab="CaO [wt.-
  font.lab=1,log="")

xyplotDAS(datname="KOLA95_Chor",varnames=c("XCOO","YCOO"),
  selection=c("Groups","COUN"),levels=c("NOR","FIN","RUS"),
  symb=c(8,3,4),col=c("red","green3","blue"), cex=c(.5,1,2),
  main="groups selected", xlab="Al [mg/kg]", ylab="CaO [wt.-
  legend="by coordinates",legend_x=0.78,legend_y=0.97,legend_ex=1)
```

Appendix B

Commands for Generating the Subsets and Diagrams

B.1 Examples for Generating Subsets

The command below generates the subset "subEx" defined by an expression.

```
makeSubsetDAS(KOLA95_Chor, datname="KOLA95_Chor", subname="subEx",
  ind=c(47,58,134,151,320,346,563,569), form=" Al > 50000 ", desc=
  "Al bigger 50000", overwrite=TRUE)
```

For "sub01", the command is:

```
makeSubsetDAS(KOLA95_Chor, datname="KOLA95_Chor", subname="sub01",
  ind=c(3,5,9,10,11,13,18,20,21,22,23,24,25,26,27,42,43,44,45,58,
  60,61,62,63,81,82,83,84,85,86,87,88,95,96,97,98,99,100,101,102,
  104,105,119,120,121,122,123,124,125,126,127,128,129,130,131,136,
  137,138,139,140,159,160,161,162,175,176,177,178,179,180,181,182,
  183,184,185,186,192,193,194,195,196,197,198,199,200,201,202,203,
  228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,
  244,245,246,247,248,249,258,259,260,261,262,263, 264,265,266,267,
  351,352,353,354,355,356,357,358,359,360,361,362,363,364,380,381,
  382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,
  398,399,400,401,402,403,404,405,406,407,408,409,424,425,426,427,
  428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,
  460,461,462,463,464,465,466,467,468,469,487,488,489,490,491,504,
  505,506,535,536,572,573,574,575,576,577,578,579,580,581,582,583,
  584,599,600,601,602,603,604,605,606), desc="vector0/1", form=
  "indicator",overwrite=TRUE)
```

B.2 Boxplot

A boxplot is created by the following command:

```
boxplotDAS(datname="KOLA95_Chor", varnames=c("Al","Fe"), selection=
  "Subsets", levels=c("subExpr","subRI"), names=c("Al [mg/kg]\n
  subExpr (279)","Al [mg/kg]\nsubRI (596)","Fe [mg/kg]\nsubExpr (279)",
  "Fe [mg/kg]\nsubRI (596)"),main="Boxplot of Al and Fe",notch=TRUE,
  range=1.5, width=c(NULL), varwidth=FALSE, outline=TRUE, boxwex=0.8,
  staplewex=0.5, outwex=0.5, plot=TRUE, col="white", at=c(NULL))
```

B.3 Ternary Plot

An example of a generating command for a ternary plot:

```
ternaryplotDAS(datname="KOLA95_Chor", varnames=c("Al","Fe","Ca"),
  main="Ternaryplot", Selection="Subsets",levels=c("subExpr","subRI"),
  names=c("subExpr (279)","subRI (597)"), variablesLabel=
  c("Al [mg/kg]","Fe [mg/kg]","Ca [mg/kg]"), position="corner", color=
  "black", symbolset="default", cex=1,sampleIdentification="none",
  idColor="black", identificationGrid=TRUE, gridColor="gray60",
  gridLabelsColor="darkgray", borderColor="black", legend=TRUE,
  legend_x=0.78, legend_y=0.9, legend_ex=1, newPlot=TRUE,
  rowSumsScale=1)
```

B.4 Xy-plot

Create an xy-plot with:

```
xyplotDAS(datname="KOLA95_Chor", varnames=c("XCOO","YCOO"),
  selection="Subsets", levels=c("subExpr","subTF_new","subRI"), symb=
  c(8,3,19), col=c("cyan","green3","red"), cex=c(1,1,2), legend=
  "by coordinates", legend_x=0.75, legend_y=0.95, legend_ex=1,
  main="", cex.main=1.2, cex.sub=1, xlab="XCOO [m east]",
  ylab="YCOO [m north]", cex.lab=1, font.lab=1, log="",
  xticks=TRUE, yticks=TRUE, jitter=list(), xlim=NULL, ylim=NULL,
  mar=c(5.1,4.1,4.1,2.1), newPlot=1)
```

B.5 Density

A command for the density is:

```
densityDAS(datname="KOLA95_Chor", varnames=c("Al","Fe"), selection=
  "Subsets", levels=c("subTF_new","subI"), lty=c("solid","dashed",
  "dotted","dotdash"), col=c("red","green3","blue","magenta"),lwd=
  c(1,1,1,1), main="Densities", cex.main=1.2, cex.sub=1, log="",
  xticks=TRUE, xlim=NULL, margin=c(5.1,4.1,4.1,2.1),
  newPlot=TRUE, legend="by coordinates", legend_x=0.6, legend_y=0.9,
  legend_ex=1 )
```

B.6 Histogram

A histogram is created with the command:

```
histDAS(datname="KOLA95_Chor", varname=c("Fe"), selection="Subset",
  level=c("subRI"),dens=T,scatter=T,box=T)
```