Diplomarbeit

zum Thema

Mapping with DAS+R

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

Johannes Löffler
Matr.-Nr.: 0027355
3552 Stratzing, Untere Haupstr. 7

Wien, am May 6, 2007 Johannes Löffler

Preface and Acknowledgements

It was very fascinating to accompany the development of a part of a software package.

I like to thank O.Univ.-Prof. Dipl.-Ing. Dr.techn. Rudolf Dutter, my supervisor, for all the effort and the help he gave to me. Moreover, I am grateful to him for the long discussions in which the program was developed.

Further I like to thank my parents for all the support they gave to me.

Johannes Löffler

Contents

List of Figures

Chapter 1

Introduction

In many applied sciences most data sets have a spatial component. For the study of these data the result must be analyzed depending on the geographical coordinates. Therefore the data are presented in maps. A method to display data on a map is to group the data and represent each group by a symbol. There are different methods to find such groups or classes, e.g. absolute class borders, classes defined by the percentiles based on the order statistics or classes based on the boxplot. Other methods for creating maps are growing dot maps and interpolated maps. In growing dot maps classes are avoided, therefore the symbol size grows continuously in relation to the data.

This thesis describes the generation of maps with R. The programs are included in the DAS+R package which offers a graphical user interface to R. (See `http://www.statistik.tuwien.ac.at/public/dutt/DASplusR/` .) The first mapping techniques need a set of symbols for representing each class. The following chapter describes how to handle symbols.

The goals of the software project DAS+R of which this thesis is part, is described in a book manuscript with tentative title "Statistical Analysis of Environmental Data", see Reiman et al. (2007).

Chapter 2

Working with Symbol Sets

To display data on a map, the values of the data are grouped and coded by symbols. A *Symbol Set* defines a collection of symbols that can be used for different methods of grouping. As default DAS+R offers six symbol sets.

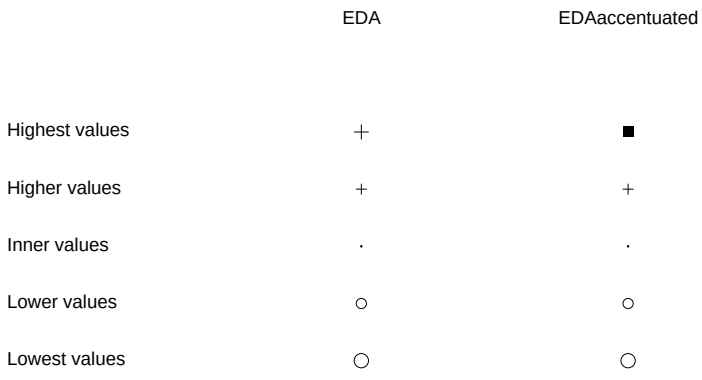

Figure 2.1: The *EDA* and *EDAaccentuated* symbol set with five symbols

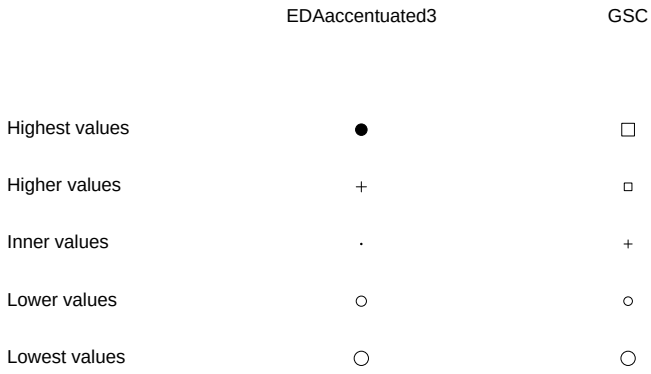

Figure 2.2: The *EDAaccentuated3* and *GSC* symbol set with five symbols

Figure 2.3: The *EDAaccentuated2* and *EDAext* symbol set with seven symbols

It is easy to edit these symbol sets or creating mew ones.

2.1 Edit a Symbol Set

To edit a symbol set, open the *Edit Symbol Sets* window. Therefore in the DAS+R window, click *Edit* → *Edit Symbol sets*.

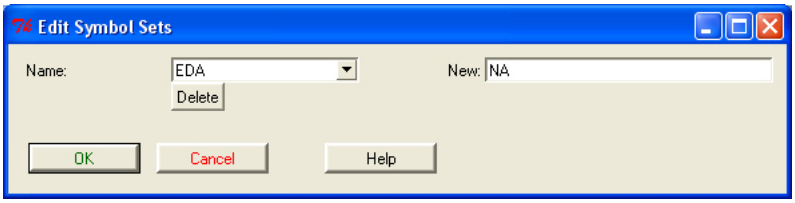

Figure 2.4: The *Edit Symbol Sets* dialog

In the list on the left site you can select an existing set.

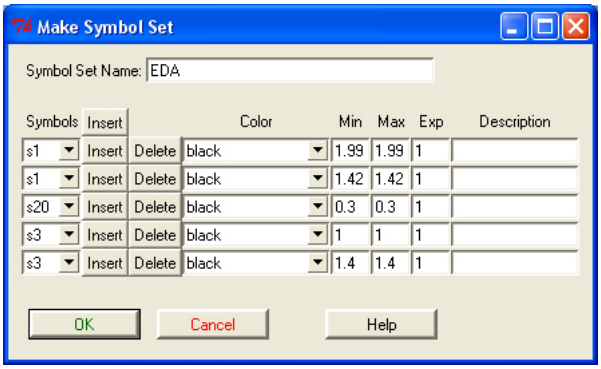

Figure 2.5: The *Make Symbol Set* dialog with the EDA symbol set

A symbol is defined by a plotting character (e.g. a character or plotting symbol), color, minimal and maximal size, exponent and a description.

The drop down box of the symbols contains the letter A-Z, a-z, the numbers 0-9, all printable ASCII symbols (a33 to a255) and the plotting characters implemented in R (see Figure 2.6).

The color drop down box offers many named and unnamed colors. The minimal and maximal sizes define the size of the symbol. The smallest element of a class is drawn with the minimal size, the largest value with the

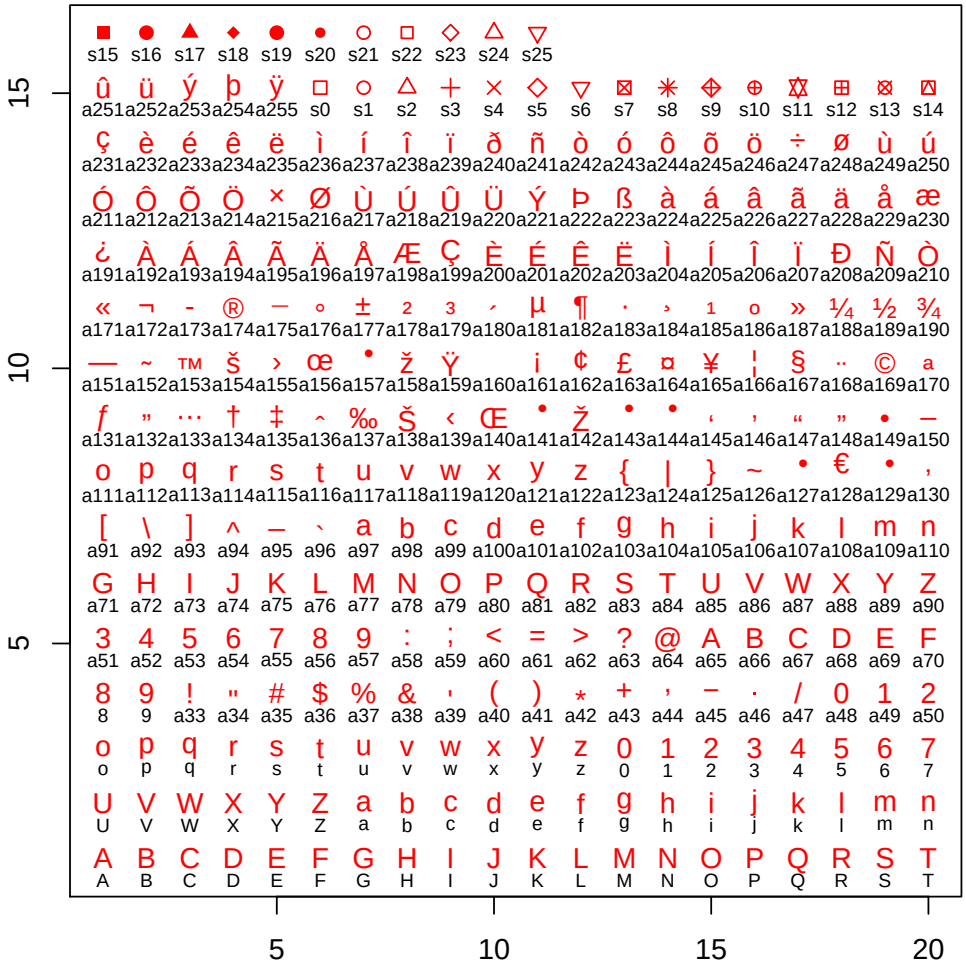

Figure 2.6: Available symbols

maximal size. If the two values are equal, all values of a class have the same size. The value of *Exponent* defines how the size grows from the minimal to the maximal size, e.g. 1 effect a linear increase, 2 a quadratic increase and so on. If the minimal and maximal sizes are equal, the exponent has no effect. The field *Description* can be used to save any additional information, but doesn't affect the plot.

To insert a new symbol to a Symbol set click the button *Insert*. The new symbol is inserted after the row in witch the button was pressed. As default the plotting character is an "A" with minimal and maximal size of 1. To

delete a symbol, press the button *Delete*. A symbol set must include at least one symbol, so the last line could not be deleted.

2.2 Create a New Symbol Set

To create a new symbol set you should enter a name on the right side in the *Edit Symbol Sets* window to the name of your choice. The *Make Symbol Set* dialog appears with one symbol. The entrance of new symbols happens the same way as described before.

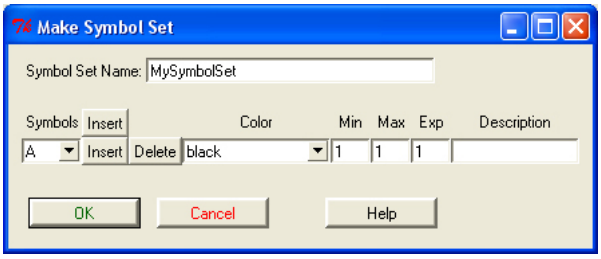

Figure 2.7: The *Make Symbol Set* dialog

2.3 Save a Symbol Set

To save a symbol set, select *File* → *Save Symbol Sets* in the DAS+R commander. Select the symbol sets which should be saved and click *OK*. Then specify the file in which the symbol sets should be stored.

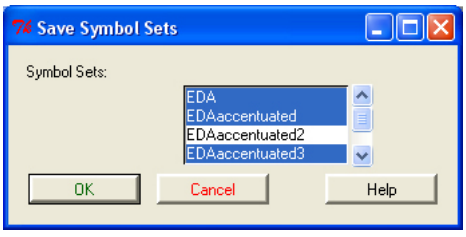

Figure 2.8: The *Save Symbol Sets* dialog

2.4 Load a Symbol Set

To load existing symbol sets select *File* → *Load symbol sets* in the DAS+R commander and specify the file which should be opened. After opening, the symbol sets can be select in the dialog which offers the opportunity to select a symbol set.

Chapter 3

Creating Maps

To create a map, select *Graphics* → *Map* from the DAS+R commander menu. The generation of a map is divided in two dialogs. In the first dialog all required information for generating the plot is specified. The second dialog defines the position of the legend, the scale bar and the north arrow.

3.1 Map Menu

After selecting the menu item the following window appears.

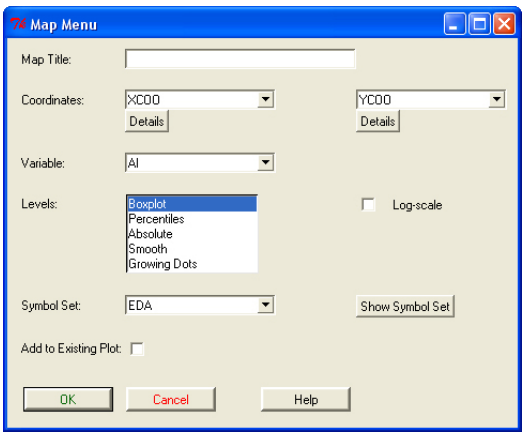

Figure 3.1: The *Map Menu* dialog

Here the variables for the coordinates and the variable displayed in the map are selected. The selection box *Levels* offer five methods of mapping. According to the selected method, the dialog on the right side of the level selection offers the possibility to change the settings for the method. The first three entries needs a symbol set, which is selected in the underneath drop-down box. The checkbox *Add to existing plot* allows to draw the map onto an existing background.

3.1.1 Boxplot Levels

In this case the classes are computed from the boxplot. There are five classes. The lower outlier, the lower whisker, the inner box, the upper whisker and the upper outlier are coded by a symbol. In case of a very skewed distribution, this method might produce an ugly legend. Therefore there is an option to draw the boxplot in the legend from the logarithmic values by selecting *Log-scale* in the first dialog.

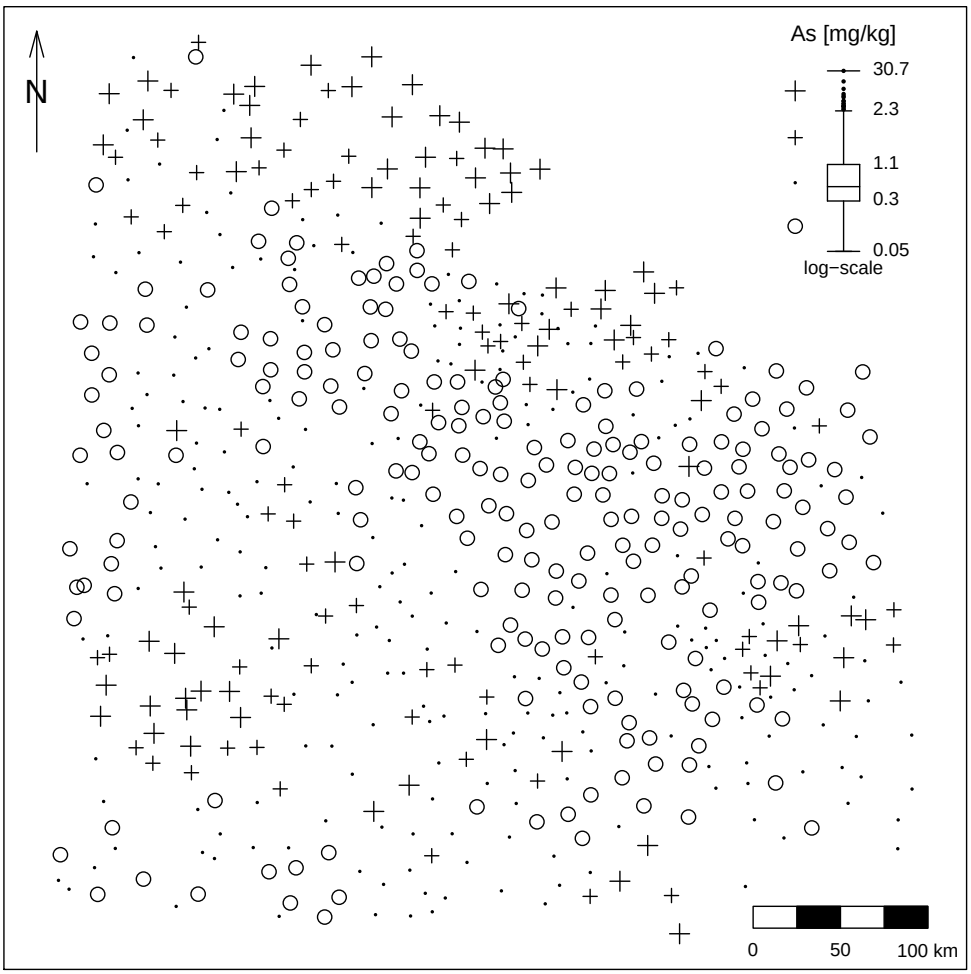

Figure 3.2: A map with boxplot levels

3.1.2 Percentile Levels

This method computes the levels from the percentiles of the considered variable. Each interval between the percentiles is coded by a symbol. In the specification box the percentiles (break points) should be entered. Values from 0 to 100 are allowed.

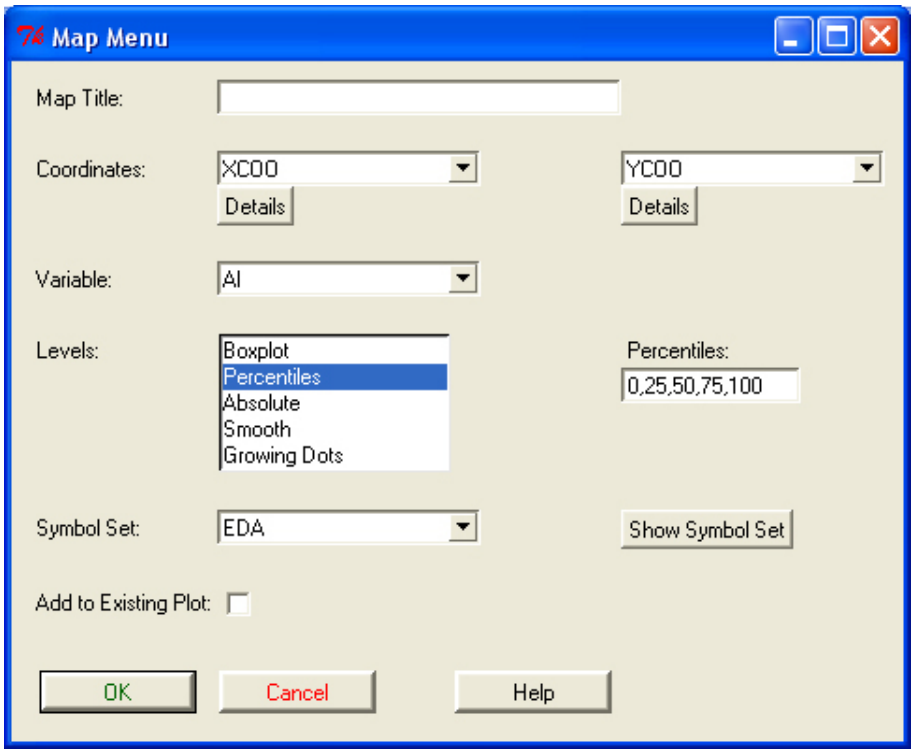

Figure 3.3: The percentile level settings

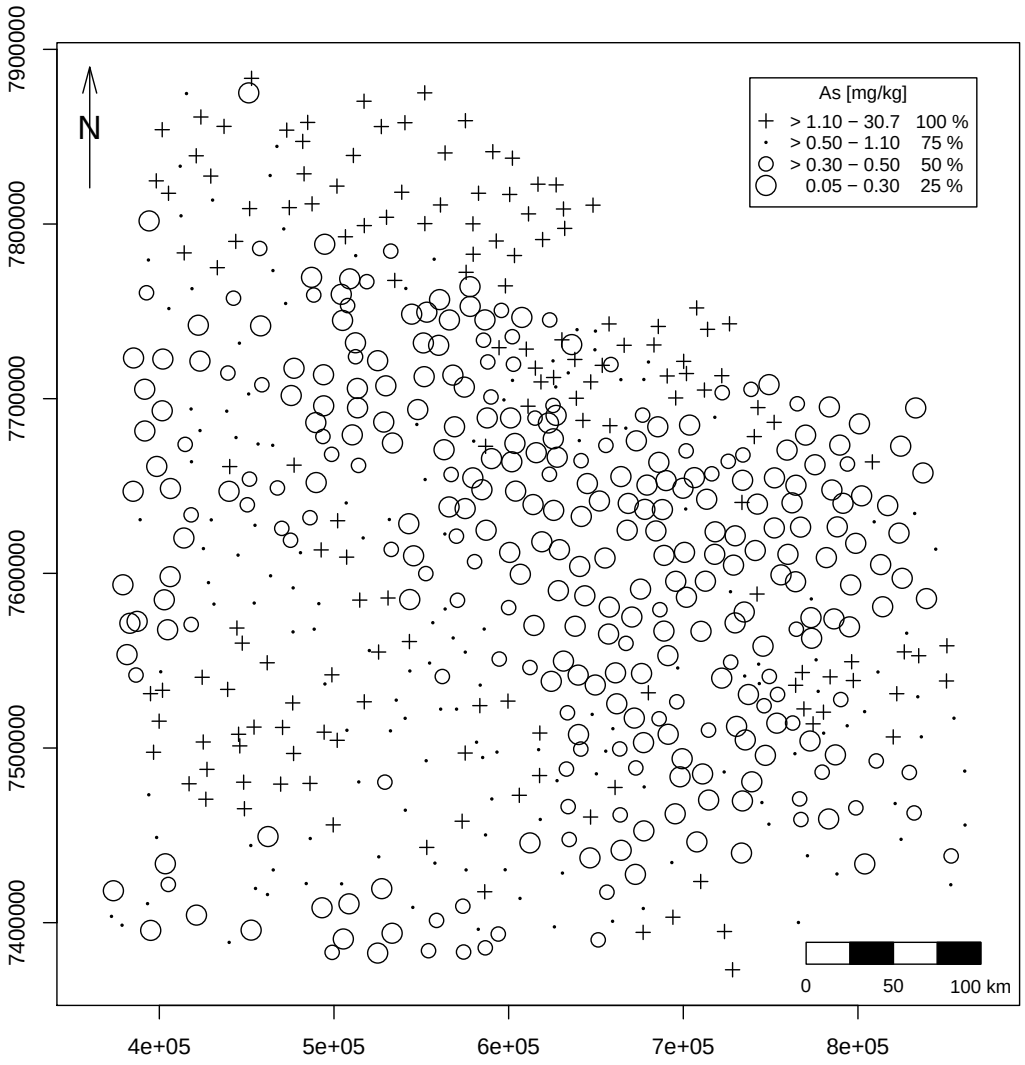

Figure 3.4: A map with percentile levels

3.1.3 Absolute Levels

In this case the levels can be chosen free by according to the values of the variable considered. The classes (-∞,$level_1$] and [$level_m$,∞) are automatically added.

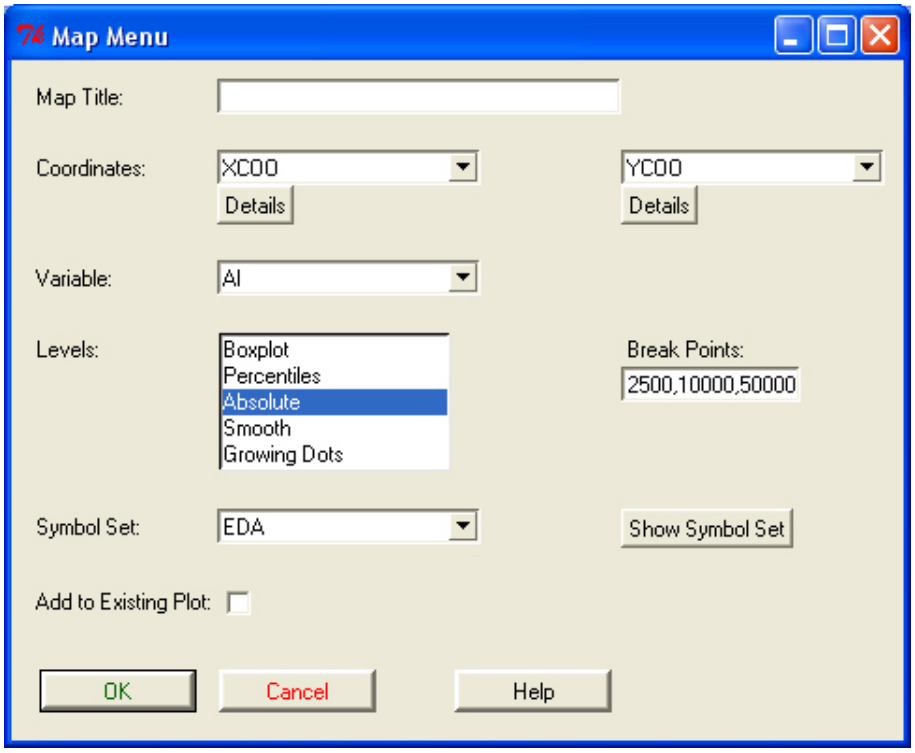

Figure 3.5: The absolute level settings

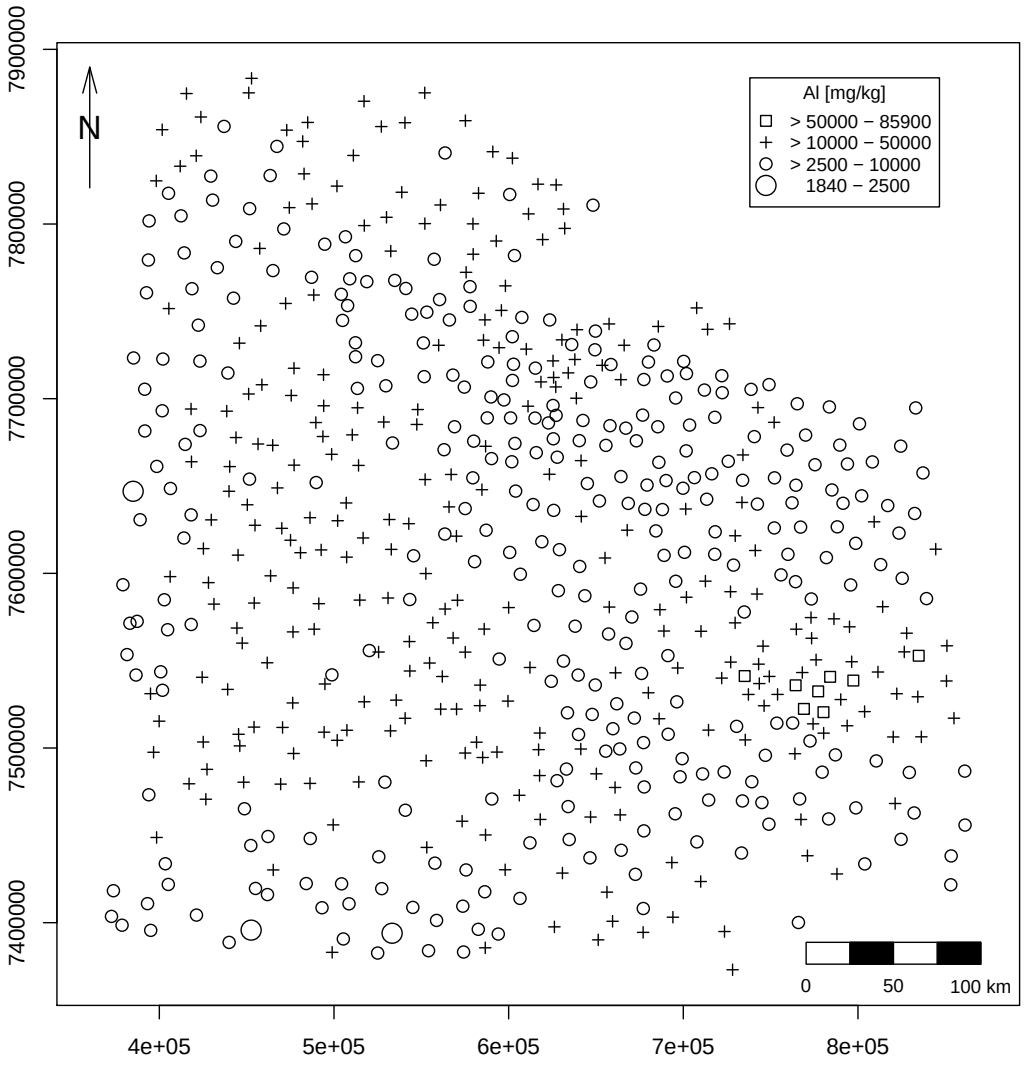

Figure 3.6: A map with absolute levels

3.1.4 Smoothing Maps

This method generates an interpolated map onto a regular grid. A click at the *Details* button opens the smoothing map settings window. The dialog

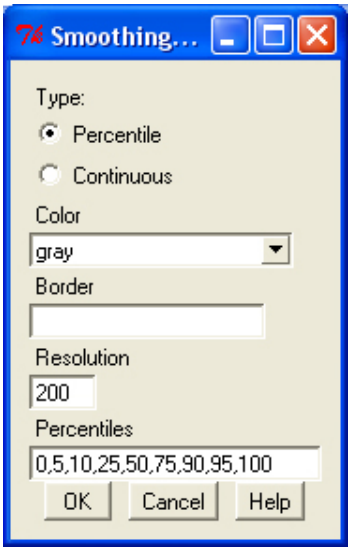

Figure 3.7: The smooth level settings

offers the following options:

Type

Percentile or Continuous - set the color breakpoints manually or use continuously the colors.

Color

Color scheme for the map. The drop-down box offers six different color schemes.

Border

Object name of a list containing the border coordinates. The list must contain two components. The first one is used for the x-coordinates, the second for the y-coordinates of the polygon presenting the border. (Remark: All smoothed values outside the defined border will be clipped.)

Resolution

The resolution of the used grid, i.e. the number of pixel in the x- respectively y-direction.

Percentiles

Only necessary if type is *Percentile*. Percentiles in which the color break points are set.

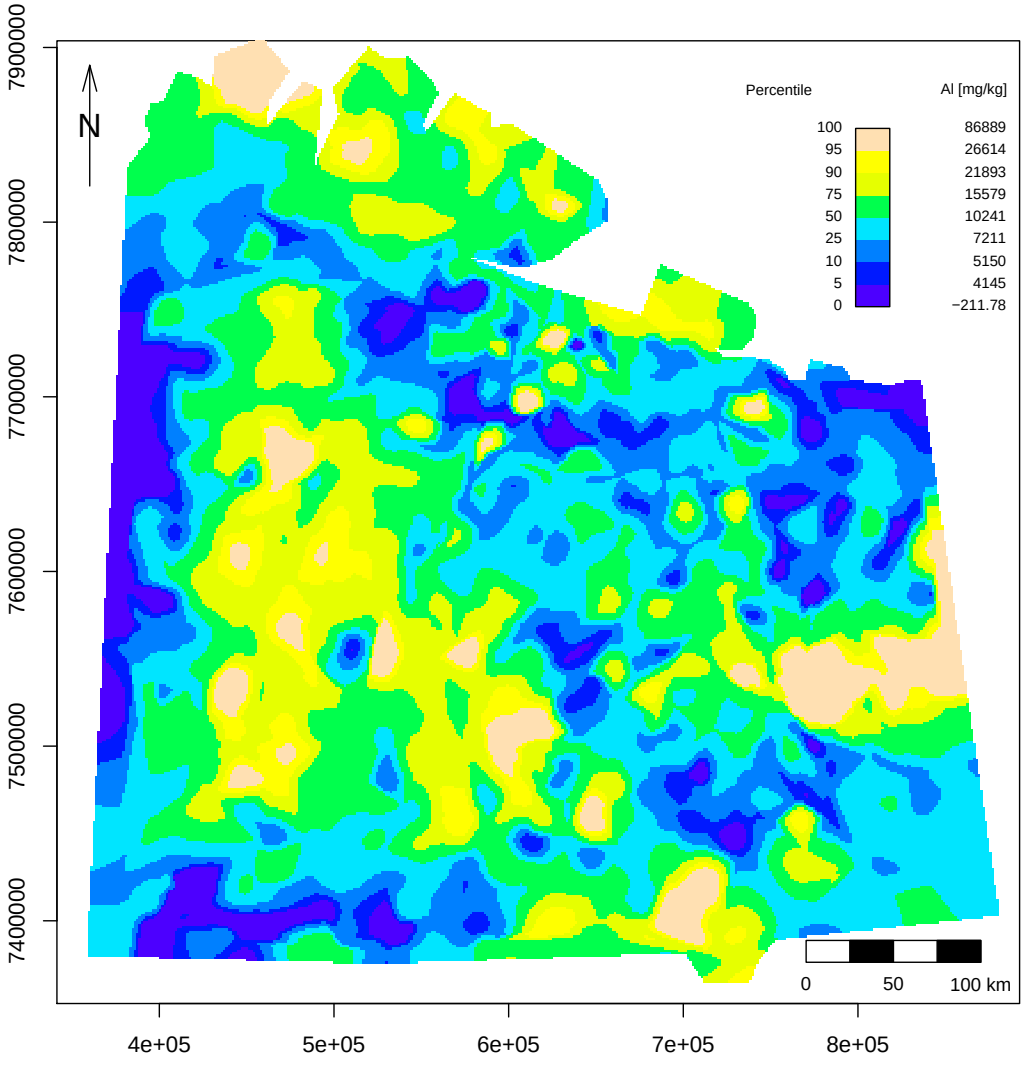

Figure 3.8: Interpolated map with manually set color break points

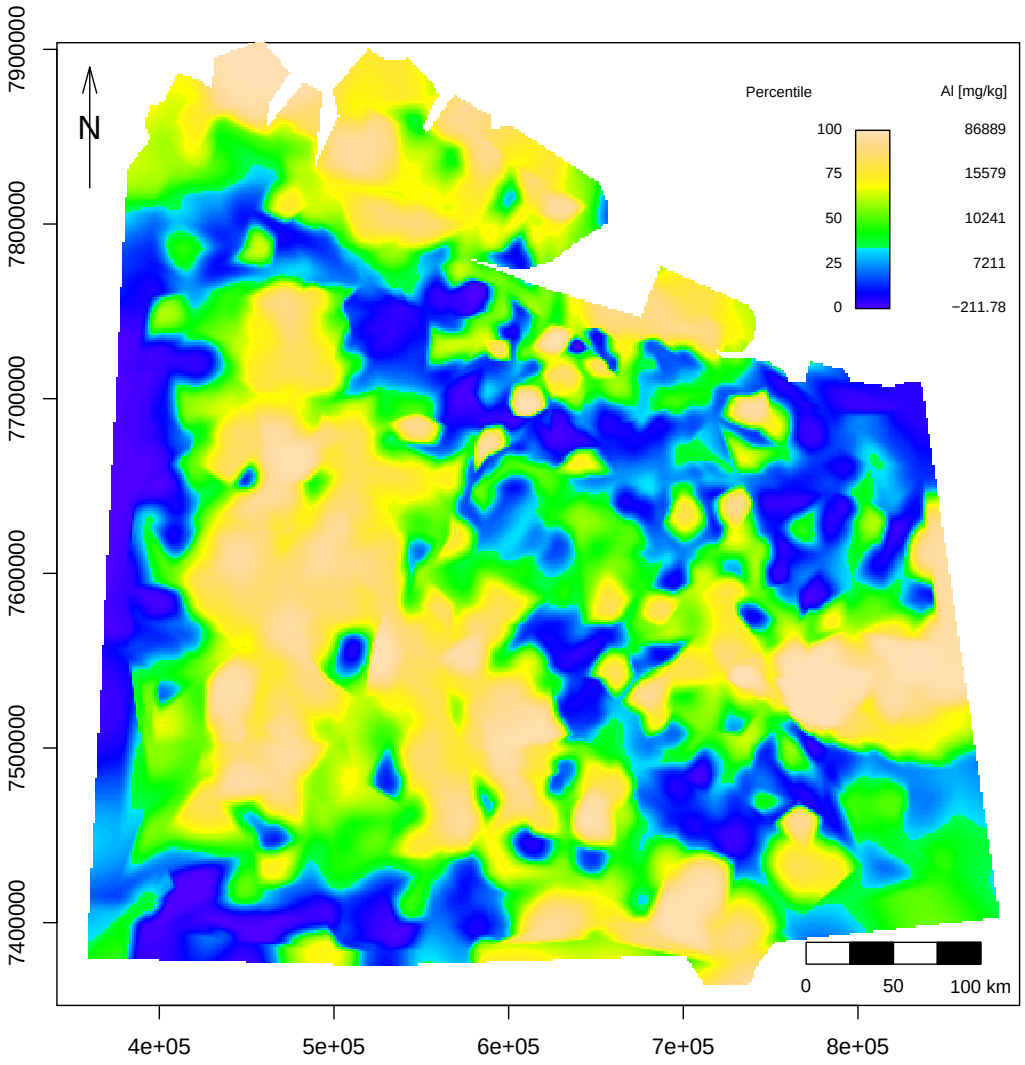

Figure 3.9: Interpolated map with continuous color break points

3.1.5 Growing Dots

This method avoids classes. It uses dots which grow according to the size of the data. In the dialog it can be chosen between two kinds of growing dots maps: linearly growing dots and exponentially growing dots. Additional settings can be specified by clicking the *Details* button.

Linearly Growing Dots:

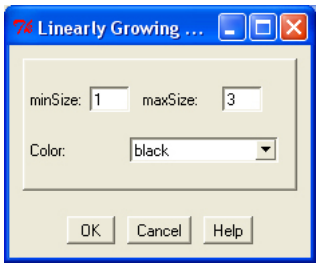

Figure 3.10: *Growing Dots Details* dialog for linearly growing dots maps

In this dialog the minimal, maximal size and color of the dots are defined.

Exponentially Growing Dots:

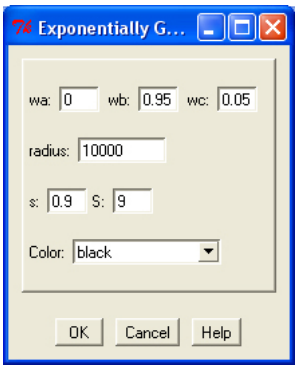

Figure 3.11: *Growing Dots Details* dialog for exponentially growing dot maps

This method uses an exponential dot size function. Therefore two percentiles (10% and 99%) from the empirical cumulative distribution function are computed. These percentiles divide the data into three segments. In this function it is necessary to choose weights wa, wb, wc for these segments. For more details see Gustavsson et al. (1997). The value of *radius* represents the radius of the largest dot in the map. The values S and s are the relative radii of the largest and smallest dot. So a dot with radius S in the exponential dot size function is drawn with a radius of *radius* in the map. The smallest dot with radius s is drawn with a radius which is proportional to the relation of s and S.

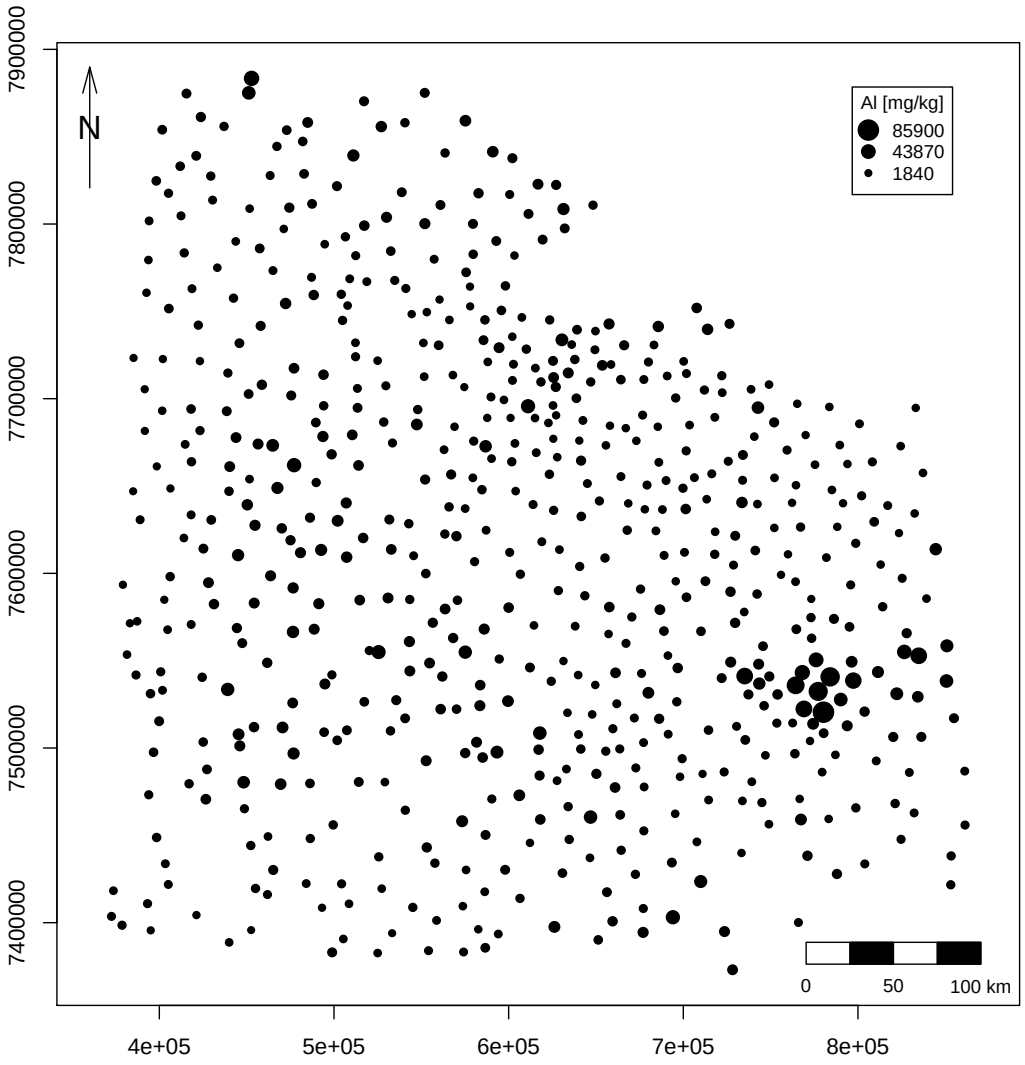

Figure 3.12: Linearly growing dots map

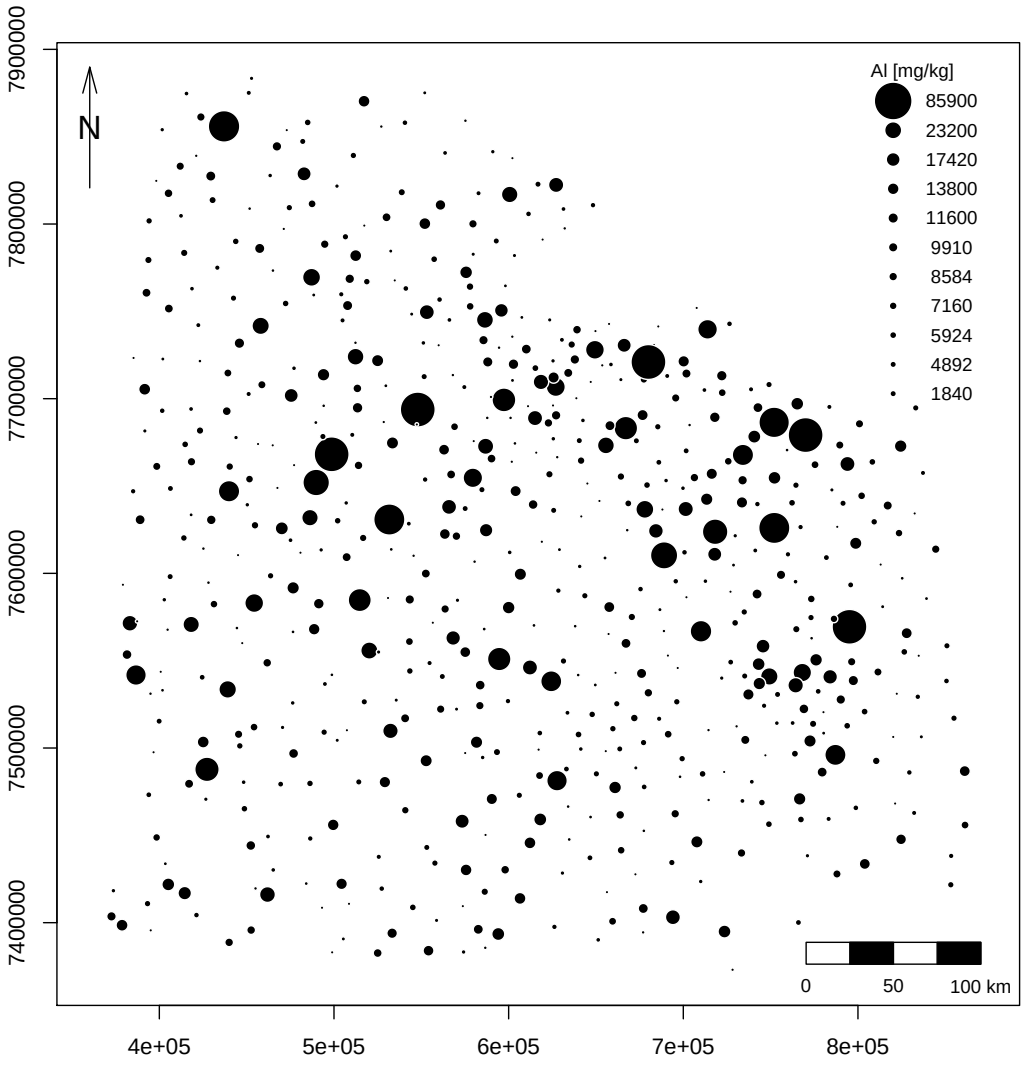

Figure 3.13: Exponentially growing dots map

3.1.6 Coordinates Details

Each coordinate variable offers the setting of display details by clicking the *Details* button below the coordinate selection field.

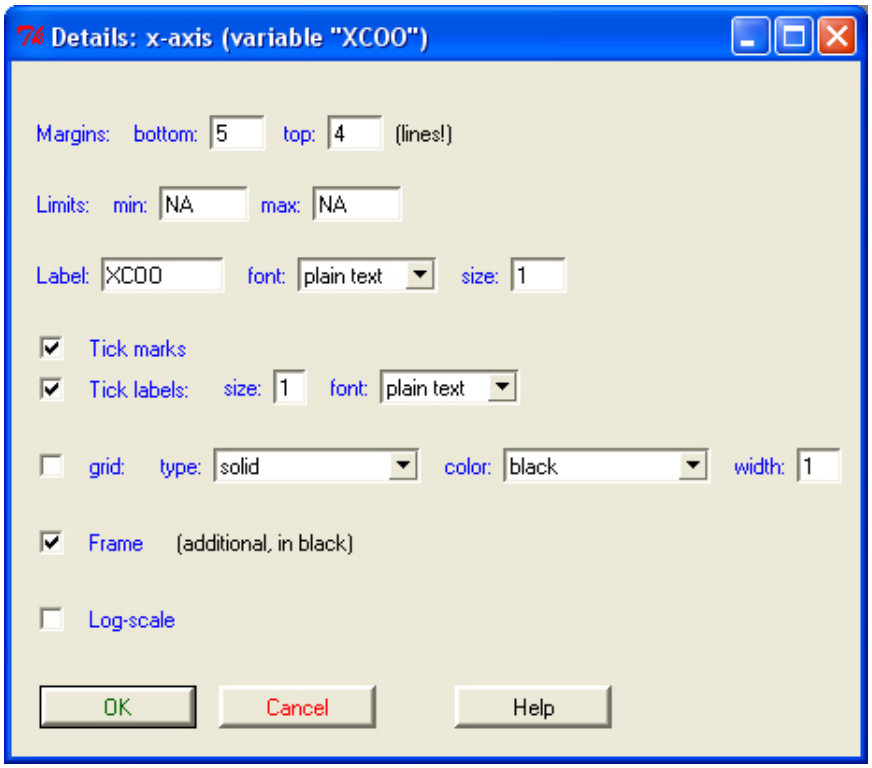

Figure 3.14: Coordinate *Details* dialog

The dialog contains following settings:

Margins
Set the margins of the plot in numbers of rows (lines). The dialog for the x-coordinate contains the settings for top and bottom margins, the dialog for the y-coordinate the settings for the left and the right margin. Nonnegative integers between zero and five are expected.

Limits
Set a lower and upper boundary of the coordinate. If no limits should be set leave the entry blank or enter *NA*.

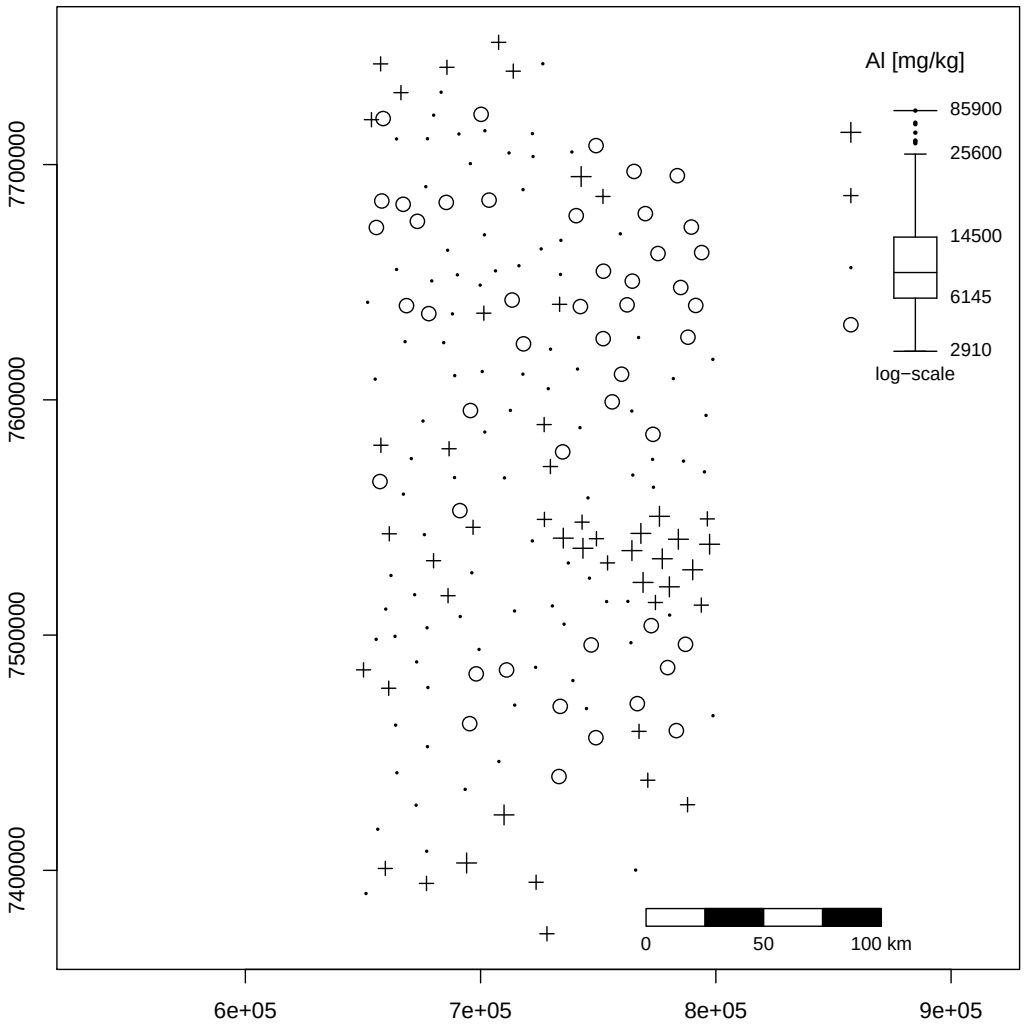

Figure 3.15: Example of a map with limited x-coordinates

Label
The label for the axis. Default is the name of the coordinate variable. Additionally the settings for the text type and the size of the label may be specified.

Tick marks
If checked, tick marks are drawn on the axis.

Tick labels
If checked, the values to the tick marks are drawn. It is also possible to select a size and the text font for the labels.

Grid
If checked, a grid is drawn with the selected line type, color and line width. Gird lines are put at the coordinates of the tick marks.

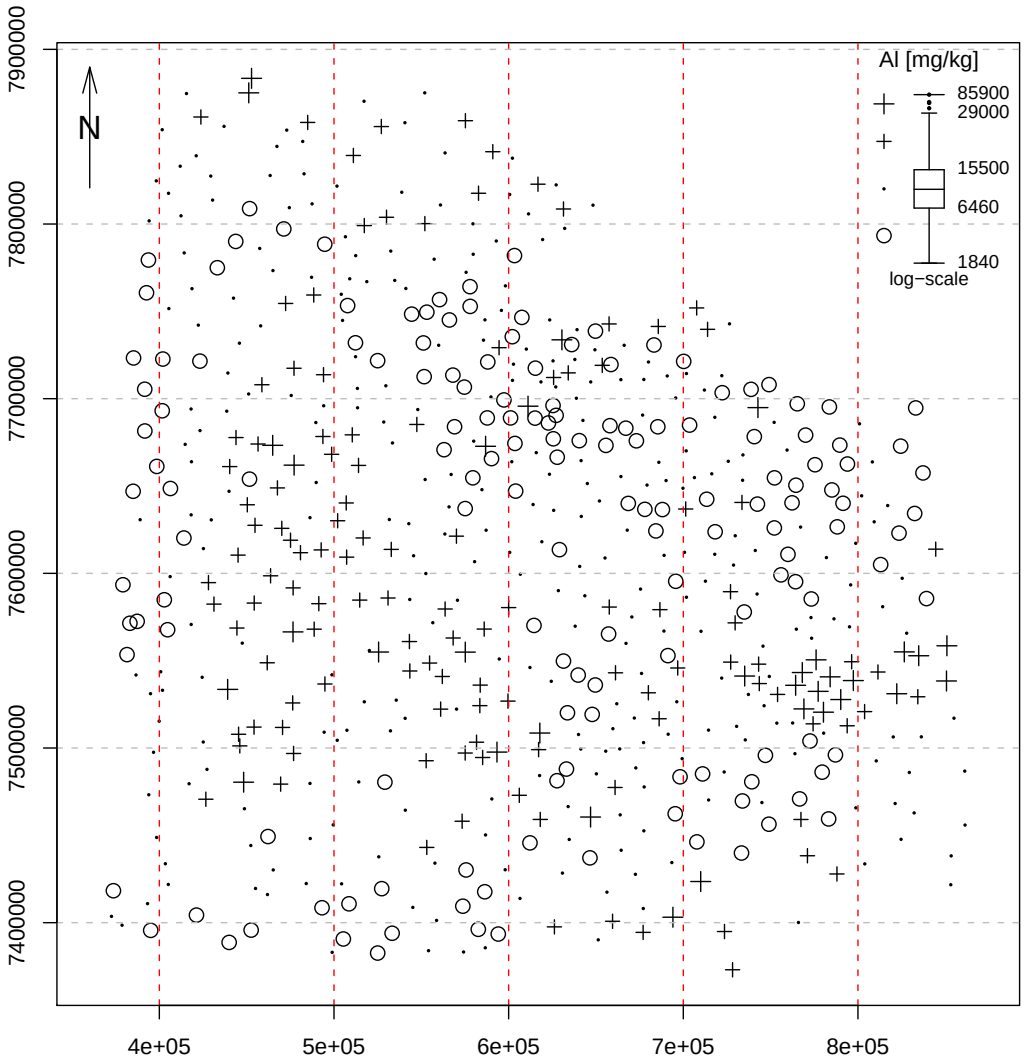

Figure 3.16: Example of a map with a grid

Frame

If selected, a frame is drawn.

Log-scale

If selected, the axis is log scaled.

3.2 Add Annotations

In the second dialog the position of the legend, the scale bar and the north arrow are defined.

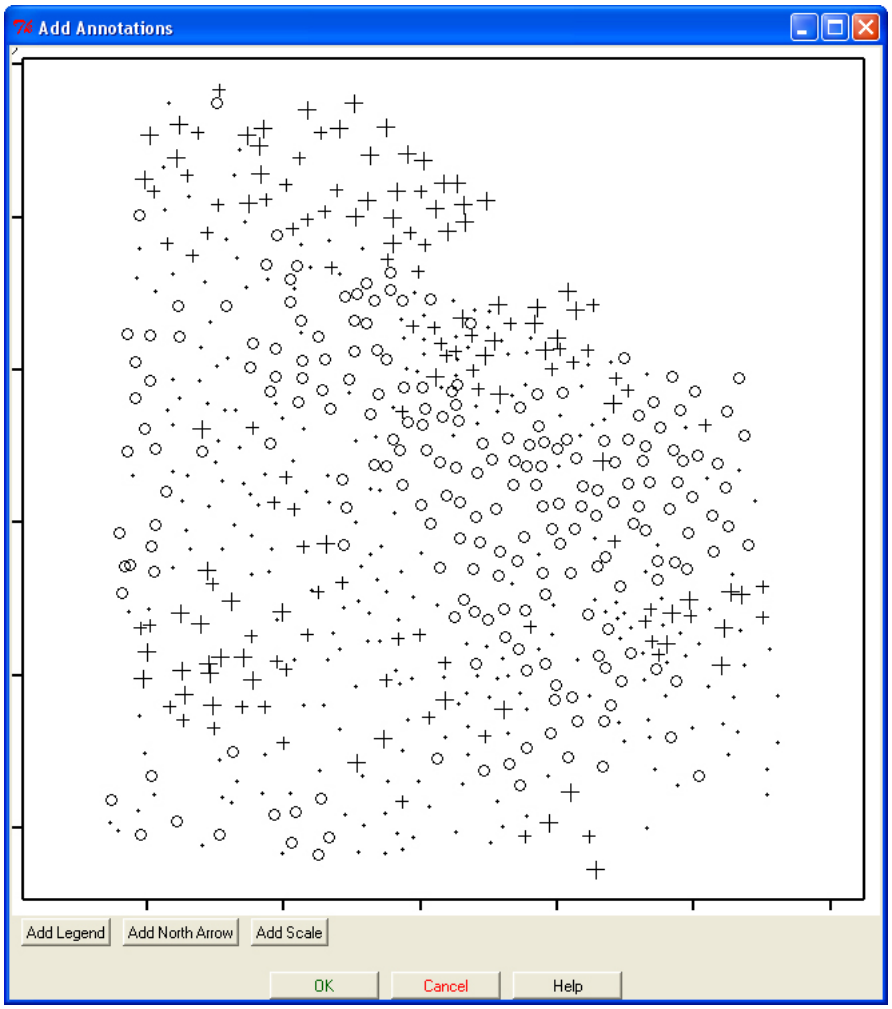

Figure 3.17: The *Add Annotations* window

To add the corresponding annotations click one of the buttons below the map. A dialog appears in which the settings are defined. All settings are saved, so if the annotations should be drawn with previous settings, click the right mouse button to avoid opening the dialog.

Legend
In the legend dialog the legend title and the size is defined. After clicking the *Add* button the legend is drawn.

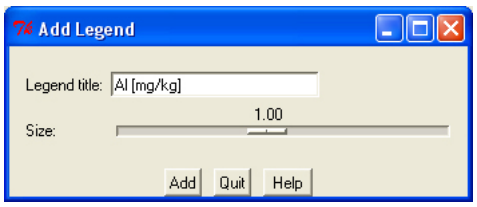

Figure 3.18: The *Add Legend* dialog

Scale
In the scale dialog a scale title and the label on the left, middle and right side of the scale are entered. After pressing *Add* the scale bar is drawn. To change the settings, repeat the procedure again.

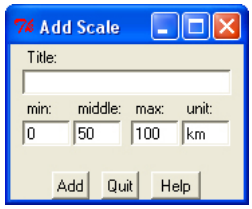

Figure 3.19: The *Add Scale* dialog

North Arrow
In this dialog the angle and the size is specified. After clicking *Add* the north arrow is drawn. To change the angle or size after drawing, press the *Add North Arrow* button again and modify the settings.

To change the position of the annotations click at the items and move them with pressed left mouse button. To move the legend, click at the temporarily added black triangle.

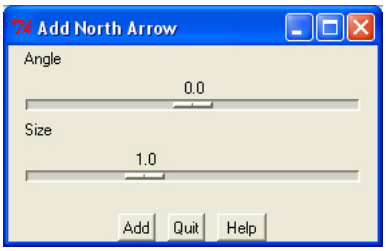

Figure 3.20: The *Add North Arrow* dialog

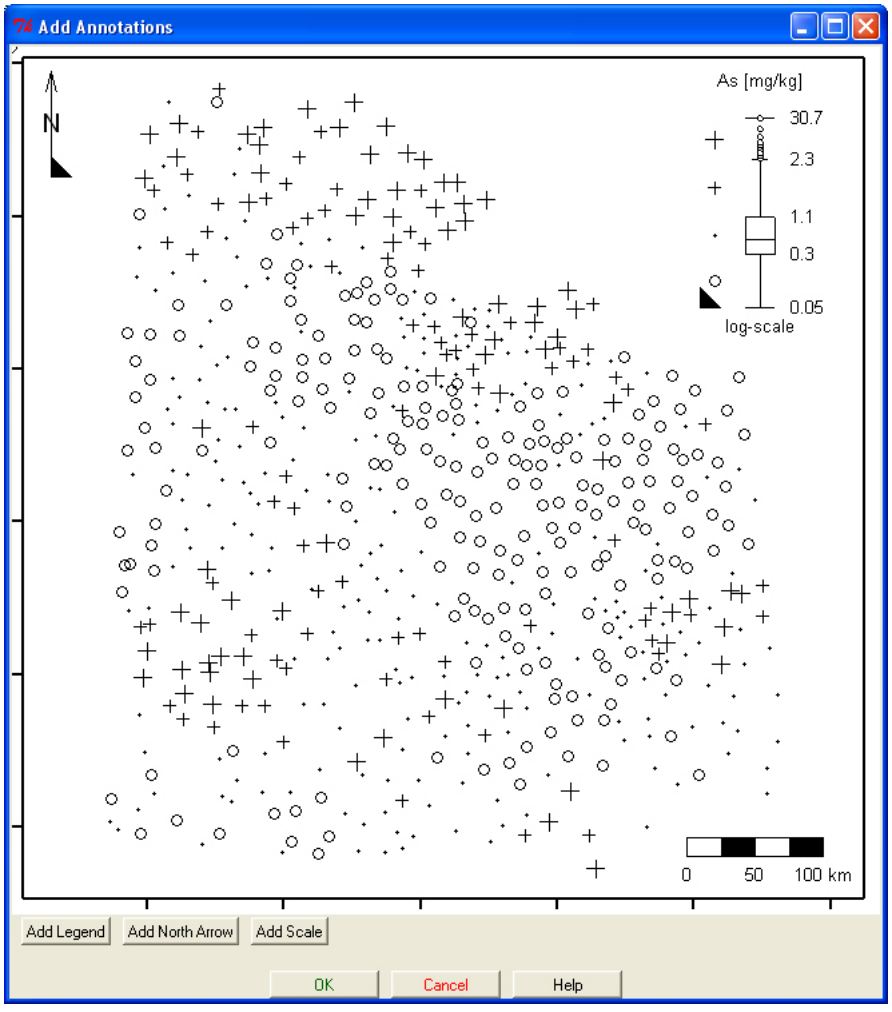

Figure 3.21: Window with added annotations

Appendix A

Helpfiles

`boxplotlegend` *Legend for boxplot levels*

Description

Adds a legend with a boxplot to an existing plot.

Usage

```
boxplotlegend(dataset,x,y,var,leg.position.x,leg.position.y,
   x.scale,y.scale,legend.title="",cex.legtit=1,
   logscale=FALSE,psymb=c(286,286,305,288,288),
   ssize=c(1.99,1.42,0.3,1,1.4),scolor=c(1,1,1,1,1),
   legend.cex=0.8,xMin=NA,xMax=NA,yMin=NA,yMax=NA,
   x.log=FALSE,y.log=FALSE)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
leg.position.x	numeric; x-coordinate of the legend
leg.position.y	numeric; y-coordinate of the legend
x.scale	numeric; half width of the box
y.scale	numeric; height of the entire boxplot
legend.title	character; legend title
cex.legtit	numeric; a numerical value giving the amount by which legend title should be scaled relative to the default

logscale	logical; if TRUE logarithmic values are used for calculating the boxplot - the values in the legend are always from the original data
psymb	a vector (length greater or equal 5) of plotting character (0-311) to be used for generating the legend
ssize	a vector (length greater or equal 5) of the size of the symbols to be used for generating the legend
scolor	a vector (length greater or equal 5) of the colors of the symbols to be used for generating the legend
legend.cex	numeric; a numerical value giving the amount by which legend text should be scaled relative to the default
xMin	numeric; value for the lower bound of the x-coordinate or NA for no boundary
xMax	numeric; value for the upper bound of the x-coordinate or NA for no boundary
yMin	numeric; value for the lower bound of the y-coordinate or NA for no boundary
yMax	numeric; value for the upper bound of the y-coordinate or NA for no boundary
x.log	logical; TRUE if logarithmic x-values
y.log	logical; TRUE if logarithmic y-values

Details

The function 'boxplotlegend()' adds a legend with symbols specified by psymb, ssize and scolor at the location specified by leg.position.x and leg.position.y.

The legend consists of a boxplot, five symbols (for the lower extreme values, lower whisker, box, upper whisker and upper extreme values) and the values of the borders of these ranges and a title. The boxplot is calculated from var in the dataset. If logscale is TRUE the log values are used. If any boundary of the coordinates given by xMin, xMax, yMin, yMax the boxplot is calculated from the corresponding values.

See Also

'plotlegend','growingDotLegend','SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",level="Boxplot",
   symbolset="EDA",title="",xlabel="", ylabel="",
   cex.xlab = "1", cex.ylab="1", font.xlab="1", font.ylab="1",
   margin=c(5,4,4,2)+0.1, xMin=NA, xMax=NA, yMin=NA, yMax=NA,
   x.grid=FALSE,x.grid.type=1, x.grid.color=1,x.grid.width=1,
   y.grid=FALSE,y.grid.type=1, y.grid.color=1,y.grid.width=1,
   x.log=FALSE,y.log=FALSE,frame=TRUE)
boxplotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
   legend.title="Al", logscale=TRUE,
   psymb=c(286,286,305,288,288,286),
   ssize=c(1.99,1.42,0.3,1,1.4,1.99),
   scolor=c("black","black","black","black","black","black"),
   leg.position.x=825048, leg.position.y=7864444,
   y.scale=137794.5,x.scale=12309, legend.cex=0.8,
   xMin=NA,xMax=NA,yMin=NA,yMax=NA)
```

`growingDot` *Growing dot map*

Description

Generates a growing dot map.

Usage

```
growingDot(dataset,x,y,var,type=c("linear","exponential"),
   color="black",radi=10000, S=9,s=0.9,wa=0,wb=0.95,
   wc=0.05,minsize=1,maxsize=3,title="",
   xlabel="",ylabel="",cex.xlab=1,cex.ylab=1,font.xlab=1,
   font.ylab=1,margin=c(5.1,4.1,4.1,2.1),
   xMin=NA,xMax=NA,yMin=NA,yMax=NA,x.grid=FALSE,
   x.grid.type=3,x.grid.color="black",x.grid.width=1,
   y.grid=FALSE,y.grid.type=3,y.grid.color="black",
   y.grid.width=1,x.log=FALSE,y.log=FALSE,
   frame=FALSE,addP=FALSE)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
type	character; "linear" or "exponential"
color	character; color which should be used for the dots
radi	numeric; radius of the lagest dot in the map
S	numeric; relative radius of the largest dot

s	numeric; relative radius of the smallest dot
wa,wb,wc	numeric; weights for 0 to 1; the weights must sum up to 1
minsize	numeric; minimal size if type is linear
maxsize	numeric; maximal size if type is linear
title	character; map title
xlabel	character; a label on the x-axis
ylabel	character; a label on the y-axis
cex.xlab	numeric; a numerical value giving the amount by which xlabel text should be scaled relatively to the default
cex.ylab	numeric; a numerical value giving the amount by which ylabel text should be scaled relatively to the default
font.xlab	numeric; font type for the xlabel
font.ylab	numeric; font type for the ylabel
margin	numerical; a vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(5, 4, 4, 2) + 0.1.
xMin	numeric; value for the lower bound of the x-coordinate or NA for no boundary
xMax	numeric; value for the upper bound of the x-coordinate or NA for no boundary
yMin	numeric; value for the lower bound of the y-coordinate or NA for no boundary
yMax	numeric; value for the upper bound of the y-coordinate or NA for no boundary
x.grid	logical; if TRUE a grid on the x-axis is drawn
x.grid.type	numeric; line type of the grid lines
x.grid.color	character; line color of the grid lines
x.grid.width	numeric; line width of the grid lines

y.grid	logical; if TRUE a grid on the y-axis is drawn
y.grid.type	numeric; line type of the grid lines
y.grid.color	character; line color of the grid lines
y.grid.width	numeric; line width of the grid lines
x.log	logical; TRUE if logarithmic x-axis
y.log	logical; TRUE if logarithmic y-axis
frame	logical; if TRUE a frame is plotted
addP	logical; if TRUE the plot is added to an existing one

Details

If type is linear the diameters of the dots grow linear with the values of var from minsize to maxsize. If type is exponential the diameters grow exponential in relation to the values of var. The exponential growing dot map needs the arguments radi, s, S, and weights wa, wb, wc. For more details see Gustavsson, N., Lampio, E. and Tarvainen, T. (1997) Visualization of geochemical data on maps at the Geological Survey of Finland. Journal of Geochemical Exploration, 59(3):197-207.

If any boundary of the coordinates given by xMin,xMax, yMin, yMax the map is generated from the corresponding values.

See Also

'plotlegend','growingDotLegend','SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
growingDot(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
```

```
   type="exponential",wa=0,wb=0.95,wc=0.05,
   radi=10000,S=9,s=0.9,xMin=NA,xMax=NA,yMin=NA,
   yMax=NA,x.grid=FALSE,x.grid.type=1,
   x.grid.color="black",x.grid.width=1,
   y.grid=FALSE,y.grid.type=1,
   y.grid.color="black",y.grid.width=1,
   x.log=FALSE,y.log=FALSE,frame=TRUE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
growingDotLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
   var="Al",type="exponential",radi=10000,S=9,s=0.9,
   title="Al [mg/kg]",leg.position.x=832864,
   leg.position.y=7882817,leg.cex=0.8,
   xMin=NA,xMax=NA,yMin=NA,yMax=NA,
   x.log=FALSE,y.log=FALSE)
```

`growingDotLegend` *Legend for growing dot maps*

Description

Adds a legend to a growing dots map.

Usage

```
growingDotLegend(dataset,x,y,var,
   type=c("linear","exponential"),radi,title,leg.cex=0.8,
   S=9,s=0.9,wa=0,wb=0.95,wc=0.05, leg.position.x,
   leg.position.y,xMin=NA,xMax=NA,yMin=NA,yMax=NA,
   minsize=1,maxsize=3,x.log=FALSE,y.log=FALSE,color="black")
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
type	character; "linear" or "exponential"
radi	numeric; radius of the lagest dot in the map
title	character; a string specifying the legend title
leg.cex	numeric; a vector of the size of the symbols to be used for generating the legend, recycled if necessary
S	numeric; relative radius of the largest dot
s	numeric; relative radius of the smallest dot
wa,wb,wc	numeric; only for type = "exponential"weights for 0 to 1; the weights must sums up to 1; more details see Gustavsson et al. (1997)

leg.position.x
: numeric; x-coordinate of the legend

leg.position.y
: numeric; y-coordinate of the legend

xMin numeric; value for the lower bound of the x-coordinate or NA for no boundary

xMax numeric; value for the upper bound of the x-coordinate or NA for no boundary

yMin numeric; value for the lower bound of the y-coordinate or NA for no boundary

yMax numeric; value for the upper bound of the y-coordinate or NA for no boundary

minsize numeric; minimal size of dots if type is linear

maxsize numeric; maximal size of dots if type is linear

x.log logical; TRUE if logarithmic x-axis

y.log logical; TRUE if logarithmic y-axis

color character; color which should be used for the dots

Details

The function 'growingDotLegend' adds a legend to a growing dot map at the location specified with leg.position.x and leg.position.y.

If any boundary of the coordinates given by xMin,xMax, yMin, yMax the legend is calculated from the corresponding values.

See Also

'plotlegend','growingDotLegend','SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
growingDot(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  type="linear",color="black",minsize=1,maxsize=3,
  title="",margin=c(2,2,2,2)+0.1,xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE,xlabel="",ylabel="",
  cex.xlab = "1",  cex.ylab="1",font.xlab="1",font.ylab="1",
  frame=TRUE,addP=FALSE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
growingDotLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",type="linear",color="black",minsize=1,
  maxsize=3,title="Al [mg/kg]",leg.position.x=796716,
  leg.position.y=7878442,leg.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

`map` *Mapping*

Description

Generates a map with percentage, absolute or boxplot levels.

Usage

```
map(dataset,x,y,var,level=c("Boxplot","Percentile","Absolute"),
   levelvalues=if(level=="Boxplot"){c(1)}, symbolset,
   title="", xlabel="", ylabel="", cex.xlab=1,
   cex.ylab=1, font.xlab=1, font.ylab=1,
   margin=c(5.1,4.1,4.1,2.1), xMin=NA,xMax=NA,yMin=NA,yMax=NA,
   x.grid=FALSE,x.grid.type=3,x.grid.color=8, x.grid.width=1,
   y.grid=FALSE,y.grid.type=3,y.grid.color=8,y.grid.width=1,
   x.log=FALSE,y.log=FALSE,frame=FALSE,addP=FALSE)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
level	character; 'Boxplot', 'Percentile' or 'Absolute'
levelvalues	numeric; if type is 'Percentile' the percentiles for the levels, if type is 'Absolute' the absolute level values
symbolset	character; the name of the Symbol Set which should be used
title	character; map title
xlabel	character; a label on the x-axis

ylabel	character; a label on the y-axis
cex.xlab	numeric; a numerical value giving the amount by which xlabel text should be scaled relative to the default
cex.ylab	numeric; a numerical value giving the amount by which ylabel text should be scaled relative to the default
font.xlab	numeric; font type for the xlabel
font.ylab	numeric; font type for the ylabel
margin	numerical; a vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(5, 4, 4, 2) + 0.1.
xMin	numeric; value for the lower bound of the x-coordinate or NA for no boundary
xMax	numeric; value for the upper bound of the x-coordinate or NA for no boundary
yMin	numeric; value for the lower bound of the y-coordinate or NA for no boundary
yMax	numeric; value for the upper bound of the y-coordinate or NA for no boundary
x.grid	logical; if TRUE a grid on the x-axis is drawn
x.grid.type	numeric; line type of the grid lines
x.grid.color	character; line color of the grid lines
x.grid.width	numeric; line width of the grid lines
y.grid	logical; if TRUE a grid on the y-axis is drawn
y.grid.type	numeric; line type of the grid lines
y.grid.color	character; line color of the grid lines
y.grid.width	numeric; line width of the grid lines
x.log	logical; TRUE if logarithmic x-axis
y.log	logical; TRUE if logarithmic y-axis

frame logical; if TRUE a frame is plotted

addP logical; if TRUE the plot is added to an existing one

Details

The function 'map()' generates a map. It plots symbols at the coordinates defined by 'x' and 'y'. The plotting symbols are defined in the Symbol Set 'symbolset'. The plotting symbol is calculated from 'var' by the method chosen via the argument 'type'.

Type 'Boxplot' has five classes defines through the lower extreme values, lower whisker, box, upper whisker and the upper extreme values from the boxplot of 'var'.

Type 'Percentiles' calculates the classes as the quantile from the vector 'levelvalues'.

Type 'absolute' defines the classes from the values given by the argument 'qutiles'.

Each class is represented by a symbol.

If any boundary of the coordinates given by xMin,xMax, yMin, yMax the map is generated from the corresponding values.

See Also

'plotlegend','growingDotLegend','SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",level="Boxplot",
   symbolset="EDA",title="",xlabel="", ylabel="", cex.xlab = "1",
   cex.ylab="1", font.xlab="1", font.ylab="1", margin=c(5,4,4,2)+0.1,
   xMin=NA, xMax=NA, yMin=NA, yMax=NA,x.grid=FALSE,x.grid.type=1,
```

```
x.grid.color=1,x.grid.width=1,y.grid=FALSE,y.grid.type=1,
y.grid.color=1,y.grid.width=1,x.log=FALSE,y.log=FALSE,frame=TRUE)
```

`Northarrow` *North arrow*

Description

Draw a north arrow.

Usage

```
Northarrow(Xbottom,Ybottom,Xtop,Ytop,Xtext,Ytext,
   Alength=0.2,Aangle=15,Alwd=1,Tcex=1.5)
```

Arguments

Xbottom	numeric; the x coordinate of the arrow shaft
Ybottom	numeric; the y coordinate of the arrow shaft
Xtop	numeric; the x coordinate of the arrow head
Ytop	numeric; the y coordinate of the arrow head
Xtext	numeric; the x coordinate of the 'N'-label
Ytext	numeric; the y coordinate of the 'N'-label
Alength	numeric; length of the edges of the arrow head (in inches)
Aangle	angle from the shaft of the arrow to the edge of the arrow head
Tcex	numeric; a numerical value giving the amount by which 'N'-label should be scaled relative to the default

Details

This function draws a north arrow from the point defined by Xbottom/Ybottom to the point defined by Xtop/Ytop. It is also drawn the letter N at the position defined by Xtext/Ytext.

See Also

'plotlegend', 'growingDotLegend', 'SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="As",
  level="Boxplot",symbolset="EDA",title="",xlabel="",
  ylabel="", cex.xlab="1",cex.ylab="1",font.xlab="1",
  font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=NA,xMax=NA,
  yMin=NA, yMax=NA,x.grid=FALSE,x.grid.type=1,
  x.grid.color=1,x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color=1,y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE)
boxplotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="As",legend.title="As [mg/kg]",logscale=TRUE,
  psymb=c(286,286,305,288,288,286),
  ssize=c(1.99,1.42,0.3,1,1.4,1.99),
  scolor=c("black","black","black","black","black","black"),
  leg.position.x=822118,leg.position.y=7867069,
  y.scale=103345.875,x.scale=9232.29016644467,
  legend.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

`plotlegend` *Legend for maps with relative or absolute levels.*

Description

The function 'plotlegend()' adds a legend to an existing plot.

Usage

```
plotlegend(dataset,x,y,var,type=c("percentile","absolute"),
   qutiles,legend.title="",psymb=c(1,1,16,3,3),
   ssize=c(1.3,0.9,0.5,0.9,1.3),scolor=c(1,1,1,1,1),
   leg.position.x,leg.position.y,
   leg.title.cex=0.8, leg.round=2,
   leg.just="right",xMin=NA,xMax=NA,yMin=NA,yMax=NA)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the occupied variable
type	character; "absolute" or "percentile"
qutiles	numeric; if type is 'relative' the percentiles for the levels, if type is 'absolute' the level values
legend.title	character; a string specifying the legend title
psymb	numeric; a vector of plotting character (0-311) to be used for generating the legend, recycled if necessary
ssize	numeric; a vector of the size of the symbols to be used for generating the legend, recycled if necessary

scolor numeric; a vector of the colors of the symbols to be used for generating the legend, recycled if necessary

leg.position.x
numeric; x-coordinate of the legend

leg.position.y
numeric; y-coordinate of the legend

leg.title.cex numeric; a numerical value giving the amount by which legend title should be scaled relative to the default

leg.round numeric; number of digits of the legend values

leg.just character; "left", "right" or "center" specifying the justification

xMin numeric; value for the lower bound of the x-coordinate or NA for no boundary

xMax numeric; value for the upper bound of the x-coordinate or NA for no boundary

yMin numeric; value for the lower bound of the y-coordinate or NA for no boundary

yMax numeric; value for the upper bound of the y-coordinate or NA for no boundary

Details

The function 'plotlegend()' adds a legend with symbols specified through psymb, ssize and scolor at the location specified with leg.position.x and leg.position.y.

If type is 'absolute' the borders of the classes are given by the vector 'qutiles' including the class less then the first value and the class greater then the last value.

If type is "relative" the classes are calculated from the percentages given by qutiles.

If any boundary of the coordinates given by xMin,xMax, yMin, yMax the legend is calculated from the remaining values.

See Also

'plotlegend','growingDotLegend','SmoothLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)

map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",level="Percentile",
    levelvalues=c(0,50,75,100),symbolset="EDA",title="",xlabel="",
    ylabel="", cex.xlab = "1",cex.ylab="1", font.xlab="1",
    font.ylab="1", margin=c(5,4,4,2)+0.1, xMin=NA, xMax=NA,
    yMin=NA, yMax=NA,x.grid=FALSE, x.grid.type=1,x.grid.color=1,
    x.grid.width=1,y.grid=FALSE,y.grid.type=1, y.grid.color=1,
    y.grid.width=1, x.log=FALSE, y.log=FALSE, frame=TRUE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
plotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
    type="percentile", qutiles=c(0,0.5,0.75,1),
    legend.title="Al [mg/kg]",psymb=c(286,286,305,288),
    ssize=c(1.99,1.42,0.3,1),
    scolor=c("black","black","black","black"),
    leg.position.x=725398,leg.position.y=7876693,leg.title.cex=0.8,
    xMin=NA,xMax=NA,yMin=NA,yMax=NA)
```

`scalebar` *Scale bar*

Description

The function 'scalebar()' adds a scale bar to an existing plot.

Usage

```
scalebar(X,Y,min,middle,max,unit,title)
```

Arguments

X	numeric; the x-coordinate of the left bottom corner
Y	numeric; the y-coordinate of the left bottom corner
min	character; the label on the left side of the scalebar
middle	character; the label in the middle of the scalebar
max	character; the label on the right side of the scalebar
unit	character; the label next to 'max'
title	character; title of the scale bar

Details

The function 'scalebar()' adds a scale bar at the position defined by X,Y to an existing plot. The scale text is defined by the 'min', 'middle', 'max' and 'unit' argument.

See Also

'plotlegend', 'growingDotLegend', 'SmoothLegend', 'Northarrow'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",level="Boxplot",
    symbolset="EDA",title="",xlabel="", ylabel="", cex.xlab = "1",
    cex.ylab="1", font.xlab="1", font.ylab="1",
    margin=c(5,4,4,2)+0.1, xMin=NA, xMax=NA, yMin=NA, yMax=NA,
    x.grid=FALSE,x.grid.type=1, x.grid.color=1,x.grid.width=1,
    y.grid=FALSE,y.grid.type=1, y.grid.color=1,y.grid.width=1,
    x.log=FALSE,y.log=FALSE,frame=TRUE)
scalebar(784016,7391131,min="0", middle="50", max="100",
    unit="km", title="Scale title")
```

`smoothingMap` *Interpolated maps*

Description

Create a two-dimensional interpolation map.

Usage

```
smoothingMap(dataset,x,y,var,title="",xlabel="",ylabel="",
   cex.xlab=1,cex.ylab=1,font.xlab=1,font.ylab=1,
   margin=c(5.1,4.1,4.1,2.1),xMin=NA,xMax=NA,
   yMin=NA,yMax=NA,x.grid=FALSE,x.grid.type=3,
   x.grid.color="black",x.grid.width=1,y.grid=FALSE,
   y.grid.type=3,y.grid.color="black",y.grid.width=1,
   x.log=FALSE,y.log=FALSE,resol=200,
   type=c("contin","percentile"),
   whichcol=c("gray","rainbow","rainbow.trunc",
   "rainbow.inv","terrain","topo"),
   qutiles=c(0,0.05,0.1,0.25,0.50,0.75,0.90,0.95,1),
   frame=FALSE,border=NULL)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
title	character; map title
xlabel	character; a label on the x-axis

ylabel character; a label on the y-axis

cex.xlab numeric; a numerical value giving the amount by which xlabel text should be scaled relatively to the default

cex.ylab numeric; a numerical value giving the amount by which ylabel text should be scaled relatively to the default

font.xlab numeric; font for the xlabel

font.ylab numeric; font for the xlabel

margin numerical; a vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(5, 4, 4, 2) + 0.1.

xMin numeric; value for the lower bound of the x-coordinate or NA for no boundary

xMax numeric; value for the upper bound of the x-coordinate or NA for no boundary

yMin numeric; value for the lower bound of the y-coordinate or NA for no boundary

yMax numeric; value for the upper bound of the y-coordinate or NA for no boundary

x.grid logical; if TRUE a grid on the x-axis is drawn

x.grid.type numeric; line type of the grid lines

x.grid.color character; line color of the grid lines

x.grid.width numeric; line width of the grid lines

y.grid logical; if TRUE a grid on the y-axis is drawn

y.grid.type numeric; line type of the grid lines

y.grid.color character; line color of the grid lines

y.grid.width numeric; line width of the grid lines

x.log logical; TRUE if logarithmic x-axis

y.log logical; TRUE if logarithmic y-axis

resol	numeric; the resolution of the grid for the interpolation
type	character; 'percentile' or 'contin'
whichcol	character; color scheme - one out of: 'gray', 'rainbow', 'rainbow.trunc', 'rainbow.inv', 'terrain', 'topo'
qutiles	numeric; vector of the quantiles in which breakpoints for the colors are taken
frame	logical; if TRUE a frame is plotted
border	a data object for the border: a list with 2 elements, a vector of the x-values and a vector of the y-values of the polygon line

Details

The function 'smoothingMap' generates a bivariate interpolation onto a grid for irregularly spaced input data. The resolution of grid is given by 'resol'. If 'type' is 'percentile' the breakpoints of the color are given by the quantiles given by 'qutiles' If 'type' is 'conti' the breakpoints are somewhat continous.

See Also

'map', 'growingDot'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
smoothingMap(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  title="",xlabel="",ylabel="",cex.xlab = "1",cex.ylab="1",
```

```
  font.xlab="1",font.ylab="1",margin=c(5,4,4,2)+0.1,
  xMin=NA,xMax=,yMin=NA,yMax=NA,x.grid=FALSE,x.grid.type=1,
  x.grid.color="black",x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color="black",y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE,type="contin",
  whichcol="terrain",border=bor,
  qutiles=c(0,0.05,0.1,0.25,0.5,0.75,0.9,0.95,1),resol=400)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
smoothLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",whichcol="terrain",resol=400,
  qutiles=c(0,0.05,0.1,0.25,0.5,0.75,0.9,0.95,1),
  type="contin",leg.title="Al [mg/kg]",
  leg.position.x=806486,leg.position.y=7755958,
  size=1,leg.title.cex=0.7,xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(798670,7378008,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(328752,7822450,328752,7891566,328752,7857445)
```

`smoothLegend` *Legend for two-dimensional interpolated maps*

Description

Adds a legend to an existing smoothing map.

Usage

```
SmoothLegend(dataset,x,y,var,resol=200,
   type=c("contin","percentile"),
   whichcol=c("gray","rainbow","rainbow.trunc",
   "rainbow.inv","terrain","topo")
   qutiles=c(0,0.05,0.25,0.50,0.75,0.90,0.95,1),
   leg.title="",leg.title.cex=0.7,leg.numb.cex=0.7,leg.round=2,
   leg.position.x,leg.position.y,size,xMin=NA,xMax=NA,
   yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
```

Arguments

dataset	a data object of class 'DASData' or 'data.frame'
x	character; the object name of the x-coordinate
y	character; the object name of the y-coordinate
var	character; the object name of the spatial variable
resol	numeric; the resolution of the grid for the interpolation
type	character; 'percentile' or 'contin'
whichcol	character; color scheme - one of the following: 'gray', 'rainbow', 'rainbow.trunc', 'rainbow.inv', 'terrain', 'topo'
qutiles	numeric; the percentiles (divided by 100) in which breakpoints for the colors are taken
leg.title	character; a string specifying the legend title

leg.title.cex numeric; a numerical value giving the amount by which legend title should be scaled relative to the default

leg.numb.cex
numeric; a numerical value giving the amount by which legend values should be scaled relative to the default

leg.round numeric; number of digits of the legend values

leg.position.x
numeric; x-coordinate of the legend

leg.position.y
numeric; y-coordinate of the legend

size numeric; a factor giving the amount by which the legend should be scaled relative to the default

xMax numeric; value for the lower bound of the x-coordinate or NA for no boundary

xMax numeric; value for the upper bound of the x-coordinate or NA for no boundary

yMin numeric; value for the lower bound of the y-coordinate or NA for no boundary

yMax numeric; value for the upper bound of the y-coordinate or NA for no boundary

x.log logical; TRUE if logarithmic x-axis

y.log logical; TRUE if logarithmic y-axis

Details

The function 'SmoothLegend' adds a legend to an interpolated map created with 'smoothingMap' at the location specified with leg.position.x and leg.position.y.

If 'type' is 'percentile' the break points of the color are given by the percentings given by 'qutiles'. If 'type' is conti the break points are somewhat continous.

See Also

'plotlegend', 'growingDotLegend'

Examples

```
require(DASplusR7)
DefaultSymbolSets()
data(KOLA95_C2MM)
smoothingMap(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",title="",xlabel="",ylabel="",cex.xlab = "1",
  cex.ylab="1",font.xlab="1",font.ylab="1",
  margin=c(2,2,2,2)+0.1,xMin=NA,xMax=,yMin=NA,yMax=NA,
  x.log=FALSE,y.log=FALSE,frame=TRUE,type="percentile",
  whichcol="topo",border=bor,
  qutiles=c(0,0.05,0.25,0.5,0.75,0.9,0.95,1),resol=400)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
smoothLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",whichcol="topo",resol=400,
  qutiles=c(0,0.05,0.25,0.5,0.75,0.9,0.95,1),
  type="percentile",leg.title="Al [mg/kg]",
  leg.position.x=798670,leg.position.y=7751584,
  size=1,leg.title.cex=0.7,xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

Appendix B

Commandos to Produce the Figures

B.1 Figure 3.2

```
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="As",
  level="Boxplot",symbolset="EDA",title="",xlabel="",
  ylabel="", cex.xlab="1",cex.ylab="1",font.xlab="1",
  font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=NA,xMax=NA,
  yMin=NA, yMax=NA,x.grid=FALSE,x.grid.type=1,
  x.grid.color="black",x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color="black",y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE)
boxplotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="As",legend.title="As [mg/kg]",logscale=TRUE,
  psymb=c(286,286,305,288,288,286),
  ssize=c(1.99,1.42,0.3,1,1.4,1.99),
  scolor=c("black","black","black","black","black","black"),
  leg.position.x=822118,leg.position.y=7867069,
  y.scale=103345.875,x.scale=9232.29016644467,
  legend.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.2 Figure 3.4

```
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="As",
  level="Percentile",levelvalues=c(0,25,50,75,100),
  symbolset="EDAaccentuated",title="",xlabel="",
  ylabel="", cex.xlab="1",cex.ylab="1",font.xlab="1",
  font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=NA, xMax=NA,
  yMin=NA, yMax=NA,x.grid=FALSE,x.grid.type=1,
  x.grid.color="black",x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color="black",y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
plotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="As",
  type="percentile",qutiles=c(0,0.25,0.5,0.75,1),
  legend.title="As [mg/kg]",psymb=c(286,286,305,288,300),
  ssize=c(1.99,1.42,0.3,1,1.4),
  scolor=c("black","black","black","black","black"),
  leg.position.x=738099,leg.position.y=7883692,
  leg.title.cex=0.8,xMin=NA,xMax=NA,yMin=NA,yMax=NA)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.3 Figure 3.6

```
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  level="Absolute",  levelvalues=c(2500,10000,50000),
  symbolset="GSC",title="",xlabel="", ylabel="",
  cex.xlab = "1",cex.ylab="1", font.xlab="1",
  font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=NA,xMax=NA,
  yMin=NA, yMax=NA,x.grid=FALSE, x.grid.type=1,
  x.grid.color="black",x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color="black",y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
plotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  type="absolute", qutiles=c(2500,10000,50000),
  legend.title="Al [mg/kg]", psymb=c(286,286,288,307,307),
  ssize=c(1.99,1.2,0.8,1.2,1.99),
  scolor=c("black","black","black","black","black"),
  leg.position.x=738099,leg.position.y=7883692,leg.title.cex=0.8,
  xMin=NA,xMax=NA,yMin=NA,yMax=NA)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.4 Figure 3.8

```
smoothingMap(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",title="",xlabel="",ylabel="",cex.xlab = "1",
  cex.ylab="1",font.xlab="1",font.ylab="1",
  margin=c(2,2,2,2)+0.1,xMin=NA,xMax=,yMin=NA,yMax=NA,
  x.log=FALSE,y.log=FALSE,frame=TRUE,type="percentile",
  whichcol="topo",border=KOLA95_border,
  qutiles=c(0,0.05,0.1,0.25,0.5,0.75,0.9,0.95,1),resol=400)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
smoothLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",whichcol="topo",resol=400,
  qutiles=c(0,0.05,0.1,0.25,0.5,0.75,0.9,0.95,1),
  border=KOLA95_border,type="percentile",
  leg.title="Al [mg/kg]",leg.position.x=798670,
  leg.position.y=7751584,size=1,leg.title.cex=0.7,
  xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.5 Figure 3.9

```
smoothingMap(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",title="",xlabel="",ylabel="",cex.xlab = "1",
  cex.ylab="1",font.xlab="1",font.ylab="1",
  margin=c(2,2,2,2)+0.1,xMin=NA,xMax=,yMin=NA,yMax=NA,
  x.log=FALSE,y.log=FALSE,frame=TRUE,type="contin",
  whichcol="topo",border=KOLA95_border,resol=400)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
smoothLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  whichcol="topo",resol=400,border=KOLA95_border,
  type="contin",leg.title="Al [mg/kg]",
  leg.position.x=798670,leg.position.y=7751584,
  size=1,leg.title.cex=0.7,xMin=NA,
  xMax=NA,yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.6 Figure 3.12

```
growingDot(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  type="linear",color="black",minsize=1,maxsize=3,
  title="",margin=c(2,2,2,2)+0.1,xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE,xlabel="",ylabel="",
  cex.xlab = "1",  cex.ylab="1",font.xlab="1",font.ylab="1",
  frame=TRUE,addP=FALSE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
growingDotLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",type="linear",color="black",minsize=1,
  maxsize=3,title="Al [mg/kg]",leg.position.x=796716,
  leg.position.y=7878442,leg.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.7 Figure 3.13

```
growingDot(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  type="exponential",color="black",wa=0,wb=0.95,wc=0.05,
  radi=10000,S=9,s=0.9,title="",margin=c(2,2,2,2)+0.1,
  xMin=NA,xMax=NA,yMin=NA,
  yMax=NA,x.log=FALSE,y.log=FALSE,xlabel="",ylabel="",
  cex.xlab = "1",cex.ylab="1",font.xlab="1",font.ylab="1",
  frame=TRUE,addP=FALSE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
growingDotLegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",type="exponential",color="black",radi=10000,
  S=9,s=0.9,title="Al [mg/kg]",leg.position.x=820164,
  leg.position.y=7887191,leg.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.8 Figure 3.15

```
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  level="Boxplot",symbolset="EDA",title="",xlabel="",
  ylabel="", cex.xlab="1",cex.ylab="1",font.xlab="1",
  font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=650000,
  xMax=800000,yMin=NA,yMax=NA,x.grid=FALSE,x.grid.type=1,
  x.grid.color="black",x.grid.width=1,y.grid=FALSE,
  y.grid.type=1,y.grid.color="black",y.grid.width=1,
  x.log=FALSE,y.log=FALSE,frame=TRUE,addP=FALSE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
boxplotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",legend.title="Al [mg/kg]",logscale=TRUE,
  psymb=c(286,286,305,288,288,286),
  ssize=c(1.99,1.42,0.3,1,1.4,1.99),
  scolor=c("black","black","black","black","black","black"),
  leg.position.x=884812,leg.position.y=7722946,
  y.scale=102306,x.scale=9139,legend.cex=0.8,xMin=650000,
  xMax=8e+05,yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
scalebar(770333,7376258,min="0",middle="50",max="100",
  unit="km",title="")
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

B.9 Figure 3.16

```
map(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",var="Al",
  level="Boxplot",symbolset="EDA",title="",
  xlabel="", ylabel="", cex.xlab = "1",cex.ylab="1",
  font.xlab="1",font.ylab="1",margin=c(2,2,2,2)+0.1,xMin=NA,
  xMax=NA,yMin=NA,yMax=NA,x.grid=TRUE,x.grid.type=8,
  x.grid.color="red",x.grid.width=1,y.grid=TRUE,
  y.grid.type=8,y.grid.color="green3",y.grid.width=1,x.log=FALSE,
  y.log=FALSE,frame=TRUE,addP=FALSE)
axis(1,labels=TRUE,font=1,cex.axis=1)
axis(2,labels=TRUE,font=1,cex.axis=1)
boxplotlegend(dataset=KOLA95_C2MM,x="XCOO",y="YCOO",
  var="Al",legend.title="Al [mg/kg]",logscale=TRUE,
  psymb=c(286,286,305,288,288,286),
  ssize=c(1.99,1.42,0.3,1,1.4,1.99),
  scolor=c("black","black","black","black","black","black"),
  leg.position.x=840749,leg.position.y=7874068,
  y.scale=96456,x.scale=8617,legend.cex=0.8,xMin=NA,xMax=NA,
  yMin=NA,yMax=NA,x.log=FALSE,y.log=FALSE)
Northarrow(360157,7820700,360157,7889816,360157,7855695)
```

Bibliography

N. Gustavsson, E. Lampio, and T. Tarvainen. Visualization of geochemical data on maps at the Geological Survey of Finland. *Journal of Geochemical Exploration*, 59:197–207, 1997.

C. Reiman, P. Filzmoser, R.G. Garrett, and Dutter R. *Statistical Analysis of Environmental Data (tentative).* submitted to Wiley & Sons, New York, 2007.

C. Reimann, M. Äyräs, V. Chekushin, I. Bogatyrev, R. Boyd, P. de Caritat, R. Dutter, T.E. Finne, J.H. Halleraker, Ø. Jæger, G. Kashulina, O. Letho, H. Niskavaara, V. Pavlov, M.L. Räisänen, T. Strand, and T. Volden. *Environmental Geochemical Atlas of the Central Parts of the Barents Region.* Geological Survey of Norway, Trondheim, 1998.

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