The ‘rgr’ Library and Functions

Over 100 ‘rg’ functions have been written in the last twelve years at the Geological Survey of Canada (GSC) for the S-Plus proprietary statistical software to support exploration and applied geochemical survey and research activities. Most of these function scripts have been written from ‘scratch’, however, others are based on scripts shared within the S user community on S-News. The ‘rgr’ functions are a subset of the ‘rg’ functions that run under the R system. R is the open source version of the S language, it is extensively used in academia and other institutions, see http://www.r-project.org/ .   R may be downloaded from any of the CRAN sites listed there.

The ‘rgr’ functions have been prepared so that they may be used by GSC staff, and be distributed externally to other government departments (OGDs), agencies and individuals wishing to use the Applied Geochemistry Section’s graphical and other procedures. In some instances the GSC has undertaken specific geochemical compilations for OGDs of National Geochemical Reconnaissance and other geochemical survey data sets held by the GSC. Many of these are publically available as GSC Open Files. The ‘rgr’ functions may be used to process applied geochemical data to generate summary statistics, both numeric and graphical, in support of the estimation of the boundaries of ambient and natural geochemical background variations. Implicitly this includes the setting of threshold or action levels that may trigger further field activities to determine if the cause of outliers, i.e. observations with above threshold or action levels, are due to natural or anthropogenic causes. The ‘rgr’ functions described in this document fall, dominantly, under the title of univariate exploratory data analysis tools. If observation site coordinates are available, three functions are available to display simple spatial plots, ‘maps’, for data inspection (note: these do not replace the use of a Graphical Information System, GIS, for spatial data display and analysis). A further spatial function displays a concentration-area plot that assists in determining if multi-fractal patterns are present in the data that can be used to identify boundaries between data populations related to different spatial - fractal - processes; e.g., background and anomalous.

The following notes are for those as yet unfamiliar with the use of R.

> Although the help files use TRUE and FALSE in the examples, etc., the capital letters T and F, respectively, may be used when running the functions..
>
> Quotes, “   ”, are used to enclose character strings that will be printed or displayed. To obtain a Greek µ, use \265, thus to display (µg/kg) use “... (\265g/kg) ...”, if a tab is required use \t, and \n forces a new line. The available codes for special characters such as the Greek µ may be displayed with function display.ascii.o.
>
> Where justification of text is an option, adj = 0 results in left justification, adj =1 in right justification, and adj = 0.5 in centring. Defaults are always provided.
>
> NA is an explicit way that the S language, and R, has of conveying the fact that there is

no information. A blank numeric field, i.e. ', ,'(note the space between the commas) in a table entered into R by the 'read.table' command is converted to a NA, for an actual value of zero a zero has to be explicitly entered. In some geochemical data files blanks have been converted to zeros by other software packages, in others they are set to a coded value, e.g. -9999. Tools are available in functions 'ltdl.fix.df' and 'ltdl.fix' to set these zeros or coded values to NAs if that is appropriate.

Many computational tasks cannot accept NAs, therefore software, function 'remove.na', is used internally within the 'rgr' functions to remove NAs from data.

It is common practice to set geochemical results less than the detection or quantification limit (<dl) to the negative value of the detection limit. Tools, functions 'ltdl.fix.df' and 'ltdl.fix', in data frames and vectors, respectively, are provided to convert these negative values to half the positive value of the detection limit. This is essential if logarithmic scaling is to be used in plots, or calculations are to be undertaken in logarithms.

If logarithmic scaling is required for any plots or computations ensure that the parameter log = T and, if required, logx = T.

A common construct for storing data in S and R is the data frame, this includes not only the data but also the variable names (columns) and the observation identifiers (rows). The latter commonly known as the sample numbers or IDs. In this there is a difference between natural scientists and statisticians, to a natural scientist a sample is an individual 'something' that is collected, described and measured, whereas to a statistician the sample is the whole collection of individual 'somethings' and has some size N. The data for any one variable or measurement is a column vector. The data in a data frame may be made easily accessible by attaching the data frame with 'attach(dfname)'. The attached data frame may be removed by 'detach(dfname)'. The function 'dftest(dfname)' may be run to see if a dataframe is attached or present in the R work space and identify the names of the variables it contains.

The construct 'deparse(substitute(x))' is used to generate a default variable name label, the column variable name, if no more informative text is provided by the user, e.g., Cu (a column variable name) rather than a user defined text string like "Cu (mg/kg) in surface soil". The contents of the variable name label are variously defined as xlab, xname, or name, or a similar parameter name for a y or z variable, in the functions.

A construct useful in the execution time selection of a subset of the values for a variable is the use of conditioning. While Cu leads to the processing of the entire vector of data for the column variable Cu, Cu[Cu<200] would result in the processing of only those data where the Cu value was <200. Similarly, Cu[Cu>10 & Cu<200] would result in the processing of only those data with values between, exclusively, 10 and 200. The condition may be based on the values of any variable available in the data frame, thus

Cu[Zn>200] would result in only those Cu values where the Zn value exceeded 200 being processed. Similarly for a ‘factor’ (text string variable), Cu[PM ==”Till”] (note the double = signs) results in only those Cu values where PM (the soil parent material) was recorded as Till being processed. For generating more permanent subsets from a data frame or matrix the R function ‘subset’ or the rgr Library function ‘gx.subset’ may be used.

In some cases it is required to split the data for a variable, a column vector, into subsets based on the value of some classificatory variable, factor, that appears as a column variable. For example, split(Cu, GSG) would split the data for the variable Cu into subsets on the basis of the values of GSG (Great Soil Group). The values of the criterion may be either character strings on integer numbers, there will be as many subsets as there are unique values of the criterion (factor). This technique may be used with functions ‘bwplot’ and ‘tbplot’. The value of criterion also may be computed, for example, Distance%/%10 generates a truncated (integer) value of the Distance from a fixed point divided by 10. Thus all Distances between 0 and 9.99... have a value of 0, those from 10 to 19.99... have a value of 1, and so on. Thus if Distance is the distance from a point source of contaminants, e.g., a smelter stack, a Tukey boxplot display can be generated where the individual plots graphically summarize the data in 10 km units from the source.

Where options exist for the colour infill of polygons the default is grey, ‘colr = 8'. The following are the available default R colours: 1 = black; 2 = red; 3 = green; 4 = dark blue; 5= light blue; 6 = purple; 7 = yellow; and 8 = grey. Setting colr = 0 results in no infill. To display the actual colours use function ‘display.lty()’ that also displays line styles.

Some users may find it convenient to use a ‘first’ function. Such a function can be used to load the rgr Library for use along with two other R Libraries required, set certain defaults to the user’s preferences, and set the R Working Directory to one appropriate for the data under investigation. The following is an example:

```
“first” <-
function (wd = NULL)
{
if(is.null(wd)) wd = "D:\\R\\WD"
setwd(wd)
library(rgr)
par(pty = "s", pch = 3)
cat("Default plot shape set to 'square' and plot symbol to 'plus'\n")
cat("Working directory set to:", wd, "\n")
options(warn = -1)
cat("R options set to warn = -1 to suppress unwanted graphics related messages from\n
rgr multi-panel functions and functions calling eqscplot in Library MASS\n")
}
```

This assumes that a folder C:\R has been set up outside the Program Files for R where data are to be stored. If the user has set up a folder C:\R\Project3 for data and files for Project 3, then entering ‘first(“C:\\R\\Project3") at the > in the R session will result in the rgr Library being made available, the Working Directory being set to C:\R\Project3, and any other defaults the user wishes to set being implemented. Note the use of \\ to cause the correct backslash for the file name.

The last instruction in the ‘first’ function is ‘options(warn = -1)’. The use of multi-panel displays in some rgr functions and the use of function ‘eqscplot’ from the MASS Library in others causes warning messages to be displayed concerning certain graphics parameters. These are not relevant to the user and the ‘options(warn = -1)’ statement leads to their suppression.

In practice it is important to appropriately handle any -ve values due to the presence of <dl data and any zeros or coded values indicating missing data prior to undertaking any plotting or computations. The easiest way to achieve this is by use of the ‘ltdl.fix.df’ function. Thus, if the data are in a data frame ‘dfname' the command ‘dfname.fixed <- ltdl.fix.df(dfname)’ is executed. The newly created object ‘dfname.fixed’ is attached, ‘attach(dfname.fixed)’ so that its column vectors are directly accessible. Any resulting NAs are handled appropriately in each ‘rgr’ function.

The following list describes the functions available to the user, functions that are only called internally are marked with an asterisk, together with the test data sets used in the examples. Full details are available through the on-line help files in the rgr Library.

anova1	Duplicate Sample Analysis of Variance (ANOVA)
anova2	Duplicate Sample Analysis of Variance (ANOVA), alternate input
bwplot	Plot Vertical Box-and-Whisker Plots
bwplot.by.var	Plot Vertical Box-and-Whisker Plots for Variables
bxplot	Plot a Horizontal Boxplot or Box-and-Whisker Plot
caplot	Prepare a Concentration-Area (C-A) Plot
cat2list *	Divides Data into Subsets by Factor
cnpplt	Cumulative Normal Percentage Probability (CPP) Plot
cutter *	Function to Identify into which Interval a Value Falls
dftest	Check for the Existence of a Data Frame
display.alts	Display Alt Codes
display.ascii.d	Display the ASCII Character Set by Decimal Number
display.ascii.o	Display the Windows Latin 1 Font Octal Table
display.lty	Display Available Line Styles and Colour Codes
display.marks	Display Available Plotting Marks
display.rainbow	Display the Colours of the Rainbow(36) Pallette
edamap	Plot a Map of Data using Proportional Symbols
edamap7	Plot a Symbol Map of Data Based on the Tukey Boxplot

edamap8	Plot a Symbol Map of Data Based on their Percentiles
fences	Generate and Display Fence Values
fences.summary	Generate and Save Fence Values for Data Subsets
fix.test	Test Data for Function ltdl.fix.df
framework.stats	Compute Framework/Subset Summary Statistics
framework.summary	Generate and Save Framework/Subset Summary Statistics
gx.ecdf	Plot an Emprical Cumulative Distribution Function (ECDF)
gx.hist	Plot a Histogram
gx.stats	Compute Summary Statistics
gx.subset	Extract a Subset of Rows from a Data Frame
inset	An EDA Graphical and Statistical Summary
inset.exporter	Saves an EDA Graphical and Statistical Summary
kola.c	Kola Project C-horizon Soil Data
kola.o	Kola Project O-horizon Soil Data
ltdl.fix	Replace Negative Values Representing Less Than Detects for a Vector
ltdl.fix.df	Replace Negative Values Representing Less Than Detects for a Data Frame
ms.data1	Measurement Variability Test Data
ms.data2	Measurement Variability Test Data
ms.data3	Measurement Variability Test Data
remove.na *	Remove and Count NAs
shape	An EDA Graphical Summary
syms *	Function to Compute the Diameters of Proportional Symbols
syms.pfunc	Function to Demonstrate the Effect of Different Values of p
tbplot	Plot Vertical Tukey Boxplots
tbplot.by.var	Plot Vertical Tukey Boxplots for Variables
thplot1	Display a Thompson-Howarth Plot of Duplicate Measurements
thplot2	Display a Thompson-Howarth Plot of Duplicate Measurements, alternate input
var2fact *	Rearranges Data for Variables as Factors

The ‘rgr’ Library was built at the Geological Survey of Canada, Ottawa, by Yiwen Chen working in collaboration with the author.

Robert G. Garrett
Emeritus Scientist
Geological Survey of Canada
Earth Sciences Sector
Natural Resources Canada
Ottawa, Ontario K1A 0E8

E_mail: garrett@NRCan.gc.ca
Web_site: http://cgc.nrcan.gc.ca/dir/index_e.php?id=4961

March 5, 2007