

GRASS with R: An Introductory Tutorial to Open Source GIS & Statistical Computing Software for Geospatial Analysis

Anindita Dasgupta¹, Uttam Kumar², Chiranjit Mukhopadhyay³ and T. V. Ramachandra⁴

¹Centre for Ecological Sciences, Indian Institute of Science, Bangalore – 560012, India,

e-mail: anindita_dasgupta@ces.iisc.ernet.in

²Centre for Ecological Sciences, Department of Management Studies, and Centre for Sustainable Technologies, Indian Institute of Science, Bangalore – 560012, India, e-mail: uttam@ces.iisc.ernet.in

³ Department of Management Studies, Indian Institute of Science, Bangalore – 560012, India, e-mail: cm@mgmt.iisc.ernet.in

⁴Centre for Ecological Sciences, Centre for Sustainable Technologies, and Centre for *infrastructure*, Sustainable Transport and Urban Planning, Indian Institute of Science, Bangalore – 560012, India, e-mail: cestvr@ces.iisc.ernet.in

Abstract—GRASS (Geographic Resources Analysis Support System) is a free Geographic Information System (GIS) software used for geospatial data management and analysis, image processing, graphics/maps production, spatial modelling, and visualization. GRASS is being used in academic, research and development, governmental agencies and also for commercial purposes around the world. GRASS is an official project of the Open Source Geospatial (OSGeo) Foundation. OSGeo was created to support the collaborative development of open source geospatial software and promote its widespread use.

For many advanced geo-spatial statistical analysis, GRASS has an interface to the most powerful statistical package – R. R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible. R is available as Free Software under the Free Software Foundation's GNU General Public License in source code form. Both GRASS and R compile and run on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

In this communication, an introductory material for GRASS and R is presented for novice users in the form of a tutorial. It is aimed to provide basic knowledge of both these Open Source Packages, their robustness and their strengths with examples. Finally GRASS–R interface is presented to start R from within GRASS environment.

Keywords-foss; **GRASS**; **R statistical Software**; **geo-statistics**; **open source**

I. INTRODUCTION

GRASS (Geographic Resources Analysis Support System) is a Geographic Information System (GIS) used for data management, image processing, graphics production, spatial modelling, and visualization of many types of data. It is Free

Software/Open Source released under GNU General Public License (GPL). GRASS GIS is official project of the Open Source Geospatial Foundation which was created to support the collaborative development of open source geospatial software, and promote its widespread use.

GRASS is a raster/vector GIS, image processing and graphics production system. GRASS contains over 350 programs and tools to render maps and images on monitor and paper; manipulate raster, vector, and sites data; process multi spectral image data; and create, manage, and store spatial data. GRASS uses both an intuitive windows interface as well as command line syntax for ease of operations. GRASS can interface with commercial printers, plotters, digitizers, and databases to develop new data as well as manage existing data.

GRASS supports workgroups through its Location/Mapset concept which can be set up to share data over NFS (Network File System). Keeping Locations with their underlying MAPSETS on a central server, a team can simultaneously work in the same project database. For raster data analysis, GRASS uses GDAL library and OGR library is used for vector analysis. The GRASS community maintains a Wiki page which contains getting started guides, tutorials, tips and tricks, FAQs, etc.

Many Geostatistical and GeoSpatial Data Analysis which are not supported in GRASS are possible in R. R is a language and environment for statistical computing and graphics and is a GNU project. R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes an effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis and display either on-screen or on hardcopy, and a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities. R is flexible and powerful which performs

statistical analysis along with graphics and provides scripting and interfacing that can do analysis of complex data.

R appeals differently to the different fields and is easy to use for data analysis because R has built-in commands for statistical procedures such as correlation, regression, ANOVA, various graphical procedures like histogram, box-plot, scatter plot and many more [1]. The main reason for R to be in FOSS is that its developers believe that collective and cooperative participatory approach is a proper way to fast and fair development. R software can be obtained from Comprehensive R Archive network (CRAN), which may be accessed from R project website at www.r-project.org. After the installation of R it can be initiated easily by entering R through command line. R is an interactive program that mainly works on a question and answer model. When R is ready for input then it prints ">". To end the R session type (q) and this is why it is user friendly.

While R is a general data analysis environment, it has been extensively used for modelling and simulation. The R data analysis programming language and environment is an extensible system which can be connected directly to GRASS. R consists of a base package and extensions. R comprise packages for point pattern analysis, geostatistics, exploratory spatial data analysis and spatial econometrics. The R/GRASS interface substantially improves the geospatial analysis capabilities of GRASS.

II. SPATIAL ANALYSIS WITH GRASS

GRASS supports numerous spatial analysis programs as listed in Table I.

TABLE I. GRASS SPATIAL ANALYSIS PROGRAMS.

Sl. No.	Command	Keyword
1	display commands	(d.*)
2	database commands	(db.*)
3	general commands	(g.*)
4	general3D commands	(g3.*)
5	imagery commands	(i.*)
6	paint commands	(p.*)
7	postGRASS commands	(pg.*)
8	postscript commands	(ps.*)
9	raster commands	(r.*)
10	vector commands	(v.*)
11	miscellaneous commands	(m.*)
12	site manipulation commands	(s.*)

Additionally, there is always a possibility of writing our own code as extensions. The typical applications done in GRASS are erosion, hydrologic, terrain modelling, image processing, raster analysis, landscape structure analysis, etc.

III. SPATIAL ANALYSIS WITH R

The R spatial analysis packages include spatial point processing, spatial autocorrelation, smoothing, interpolation, geostatistics, etc. The "sp" package in R offers a wide variety of geostatistical functions such as foundation classes, interface to coordinate systems, utility plotting methods, sampling methods, etc. Other R geospatial packages are listed in Table II.

TABLE II. R SPATIAL ANALYSIS PROGRAMS.

Sl. No.	Package	Purpose
1	maptools	reading and handling of shape files
2	maps	drawing of basic geographical maps
3	spgpc	polygon clipping
4	spGDAL	Geospatial Data Abstraction Library
5	spatial	spatial point pattern analysis
6	spatstat	2D point patterns multitype/marked points and spatial covariates, functions for exploratory data analysis, model-fitting, simulation, model diagnostics, and formal inference
7	splancs	space-time, emphasis on points-within-polygons
8	spdep	spatial regression, autocorrelation
9	gstat	univariate and multivariate geostatistics
10	geoR	model based geostatistics
11	geoRglm	inference in generalised linear spatial models
12	fields	curve and function fitting with an emphasis on splines, spatial data and spatial statistics, spatial covariance
13	RArcInfo	reading ArcInfo version 7 and e00 files
14	shapefiles	reading and writing ESRI shapefiles
15	RColorBrewer	color palettes optimized for thematic maps, etc.

A. R as a Calculator: R language use the usual arithmetic operators and the data type used is modes. The modes are logical, numeric and complex numbers. One of the simple possible tasks in R is to enter an arithmetic operation and receive a result. For example, if we want to add two numbers then we can type in the terminal

```
>2+2
```

```
>4
```

B. Assigning value to variables: R has symbolic variables like any other programming language that is used to represent value to the assigned variable. The operator “`<-`” is known as assignment operator and it assigns the value of the expression on right to the object on left. For example, we can assign 5 to the variable x which can be used for subsequent arithmetic expression.

```
>x<-5
```

and then type

```
>x+x
```

to get 10 as our final output

C. Methods of data entry: R can handle entire data vectors as single objects and there are various ways of inputting data in R. We can directly type the values in the command line with concatenation function “`c`”. Data can be entered one at a time from the keyboard using `scan` or “`read.table`” command.

```
w<- c(60, 72, 57, 90, 95, 72)
```

```
y<- scan()
```

```
>data<-read.table("read_my_file.txt",header=T)
```

If the data is separated by “tab” or “space” then it can be specified in the command

```
>data<-read.table("read_my_file.txt",sep="/t",header=T)
```

```
>attach (data)
```

```
>names(data)
```

D. Arithmetic operations: In R many complicated calculations can be done by using addition, subtraction, multiplication, division and exponentiation as operator.

```
>x<-3+8
```

```
>5^2-5*2
```

[1] 15 will appear as the outcome. Its better to specify the order of evaluation of expression by using parenthesis and no space is required to separate components in an arithmetic operation.

```
>1-3*3
```

[1] -8

E. Useful Builtin Function: Many calculations are already built into the program [2] and results can be easily obtained by using these functions as listed in Table III.

F. Graphics with R: R provides flexible and powerful graphical analysis. There are a number of commands which helps in obtaining simple graphs through R by using the command “`plot`” and giving various specifications about the appearance of the graph in both x and y axis. Table IV lists some of the standard plot functions.

TABLE III. R BUILT IN FUNCTIONS.

Sl. No.	Function	Purpose
1	mean	to find average
2	sd	to find standard deviation
3	max	to obtain maximum element in a data set
4	min	to find minimum element in a data set
5	range	to find range of values
6	median	to find median of the data set
7	var	to find variance of the data set
8	sort	command for sorting in increasing order of magnitude
9	diff	creates a vector of differences by subtracting previous element from next element.
10	cov(a,b)	covariance between a and b
11	cor(a,b)	correlation between a and b
12	length(a)	length of a vector
13	floor(x)	round down to next lowest integer
14	ceil(x)	round up next largest integer
15	round(x)	round to nearest integer
16	anova	analysis of variance
17	summary (x)	summarises the data (x)
18	assign("x",3)	assigns the value 3 to object x
19	c()	combines or concatenates terms together
20	rep(x, each=4)	repeats x 4 times

A simple example is to plot a sine curve. First the intervals are defined along with the number of data points to support the curve.

```
>x <- seq(-2*pi, 2*pi, len = 100)
```

```
>x
```

```
>str(x)
```

```
>summary(x)
```

Then we can plot it (the type parameter specifies the line type):

```
matplot(x, sin(x), type="l")
```

To get an idea about the various "matplot()" options, run:

```
>?matplot
```

You can see some "matplot()" examples by running:

```
>example(matplot)
```

More examples follows:

```
>barplot(table(x),xlab="DigitalNumbers",ylab="Frequency",c
ol="gray70"); where x is the data which is already being inputed
in R
```

```
> plot(c,xlab="BANDS",ylab="Digital Number")
```

```
> boxplot(a,xlab="BANDS",ylab="Digital Number) where a
is the data set which R statistical software has already read.
```

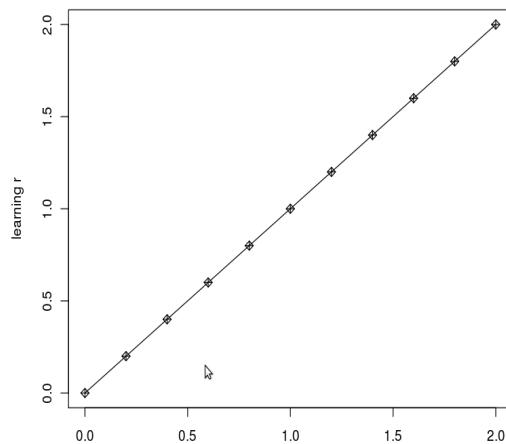


Figure 1. Line graph generated in R.

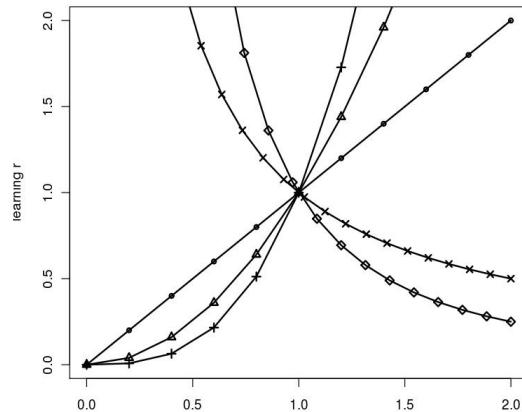


Figure 2. Quadratic line graph obtained in R.

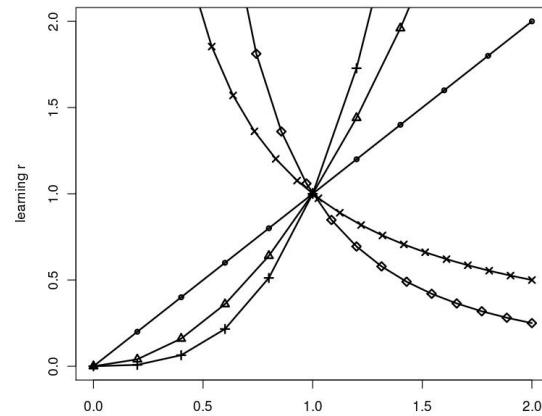


Figure 3. Line with quadratic, cubic, power -1 and power -2 generated in R.

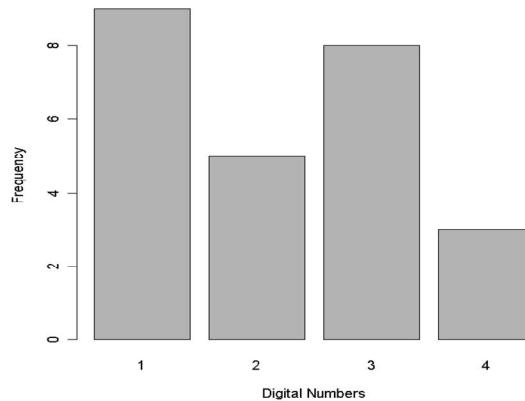


Figure 4. Bar plot.

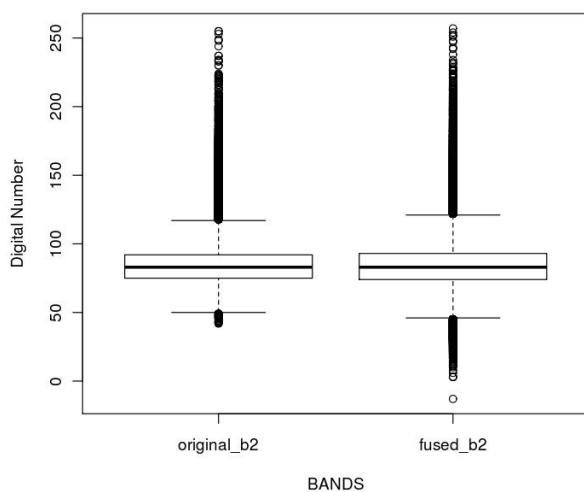


Figure 5. Boxplot.

TABLE IV. STANDARD R PLOT FUNCTIONS.

Sl. No.	Function	Name of the plot
1	plot()	Scatterplot
2	hist()	Histogram
3	boxplot()	Box-and-whiskers plot
4	stripchart()	Stripchart
5	barplot()	Bar diagram
6	stem()	Stem-and-leaf display

G. *Saving, Storing and Retrieving Work in R*: When we quit R session, we should type “yes” to “save workspace image”. When we restart R all the data and variables from the previous session are available.

H. *Getting help in R*: To get help in R use “help.search” function with the query in double quotes “ ”, for example help.search “data input”.

I. *Add-on (CRAN) Packages in R*: R has numerous addon packages for different applications. Packages can be downloaded for the CRAN Mirror site (<http://cran.r-project.org/>) and installed in R. Table V presents some of the R packages. More packages can be found on <http://cran.r-project.org/>. If we want to install packages then the simple command is “install.packages” or from the Linux Terminal type: \$R CMD INSTALL package_name.tar.gz. For example, “R CMD INSTALL spgrass6_0.3-7.tar.gz”.

TABLE V. STANDARD PACKAGES IN R.

Sl. No.	Package	Purpose
1	northeast	Normality test

2	randomforest	Classifier
3	cluster	Clustering tool
4	rgdal	Geospatial abstraction data library in R
5	raster	Geographic analysis and modelling with raster data
6	kohonen	Supervised and unsupervised self-organising maps
7	tseries	Time-series data analysis

IV. GRASS / R INTERFACE

The GRASS/R interface substantially improves the geospatial analysis capabilities of GRASS. The “spgrass6” R addon package (<http://cran.r-project.org/web/packages/spgrass6/>) provides the R to GRASS interface for raster and vector. It is an interface between GRASS GIS and R based on starting R from within the GRASS environment, or running free-standing R in a temporary GRASS location; the package provides facilities for using all GRASS commands from the R command line. When R is used in GRASS GIS, it has two meanings:

1. R is run “on top of” GRASS, transferring GRASS data to R to run statistical functions on the imported data as R objects in memory, and possibly transfer the results back to GRASS.
2. The data are stored in GRASS and R is used as a scripting language “on top of” GRASS with execGRASS() function. In this case, little data is moved to R, so memory constraints can be ignored and R functionalities are available.

For the integration of R with GRASS, you need to run R from the GRASS shell environment. The interface dynamically loads compiled GIS library functions into the R environment. The GRASS metadata about the LOCATION’s regional extent and raster resolution are transferred to R. These libraries can be called in R as (i) library(spgrass6), and (ii) library(GRASS). Chapter 10 (Using GRASS with other Open Source Tools) in Open Source GIS A GRASS GIS Approach Book written by Neteler and Mitasova [3] discusses these spatial functions in detail.

V. CONCLUSION

In this introductory tutorial to GRASS GIS and R interface, commands for geospatial data analysis available in GRASS and R were presented. This tutorial presents a noncomprehensive list of the numerous functionalities (commands) for spatial data and point pattern analysis that GRASS and R Interface provides. For further in depth discussion, the readers are directed to read materials listed in the references.

REFERENCES

[1] Sudha G. Purohit, Sharad D. Gore, Shailaja R. Deshmukh, “Statistics Using R”, Narosa Publishing House, New Delhi, 2008.

- [2] Peter Dalgaard, "Introductory Statistics with R", Springer, New York, 2002.
- [3] Neteler, M., and Mitasova, H., Open Source GIS A GRASS GIS Approach, Third Edition, Springer, New York, 2008, ISBN-13:978-0-387-35767-6, eISBN-13:978-0-387-68574-8.
- [4] <http://wgbis.ces.iisc.ernet.in/grass>
- [5] <http://grass.osgeo.org>
- [6] http://grass.osgeo.org/wiki/R_statistics
- [7] <http://sal.uiuc.edu/csis/Rgeo/>
- [8] <http://r-spatial.sourceforge.net/>
- [9] <http://cran.r-project.org/src/contrib/Views/Spatial.html>
- [10] http://grass.fbk.eu/statsgrass/grass_geostats.html
- [11] http://grass.fbk.eu/statsgrass/grass_geostats.html
- [12] http://grass.itc.it/grass61/manuals/html61_user/full_index.html
- [13] <http://grass.itc.it/gdp/manuals.php>
- [14] <http://grass.gdf-hannover.de/wiki/Documents>
- [15] <http://www.gstat.org>
- [16] <http://gnuplot.sourceforge.net>
- [17] <http://r-spatial.sourceforge.net>



Ramachandra T. V. obtained his Ph.D. from Indian Institute of Science (IISc), Bangalore. Currently he is the Co-ordinator of Energy & Wetlands Research Group at the Centre for Ecological Sciences (CES), Associate Faculty at Centre for Sustainable Technologies (CST) and Centre for *infrastructure*, Sustainable Transport and Urban Planning (CiSTUP) at IISc, Bangalore. His area of research includes remote sensing, digital image processing, urban sprawl: pattern recognition, modelling, energy systems, renewable systems, energy planning, energy conservation, environmental engineering education, etc. He is member of many national and internationally recognized professional bodies and is a recipient of the prestigious Satish Dhawan Young Engineer Award, 2007 of Karnataka State Government, India.

AUTHORS PROFILE



Anindita Dasgupta is with the Centre for Ecological Sciences, Indian Institute of Science, Bangalore. She has a Master's Degree in BioTechnology from Bangalore University, India. Her research interests include Remote Sensing, Image Processing, Geographical Information Systems, Spatial Data Analysis and Urban Sprawl.



Uttam Kumar has an Engineering Degree in Computer Science from VTU, Belgaum, India., Master's Degree in Geoinformation Science from University of Twente, The Netherlands and a Ph.D. from the Indian Institute of Science, Bangalore. His areas of research are developing algorithms for multi-satellite sensor spatial temporal data analysis. His research interests are Pattern recognition, Remote sensing, Data mining and Image processing using Free and Open Source Software (FOSS).



Chiranjit Mukhopadhyay earned a Ph.D. in Statistics from University of Missouri, Columbia in 1992, after obtaining M-Stat and B-Stat from Indian Statistical Institute, Kolkata. Currently he is Associate Professor of Statistics in the Department of Management Studies, Indian Institute of Science, Bangalore. He has served in the faculty of several Universities around the globe like The Ohio State University, Case Western Reserve University, Bilkent University etc. His current research interests include Bayesian Statistics, Reliability Theory, Statistical Quality Control, Empirical Finance and Financial Econometrics.