R Glossary and Helpsheet

This is a glossary and helpsheet for users using R as a GIS. This is a work in progress, so if you have any comments or commands that you think should be included, please let me know – nick@geospatialtrainingsolutions.co.uk.

Commands & Functions

?	shows the help file for that command, for example, `?help` or `?head`
??	will search through the help files for any reference to the word you type, `??dataframe`
#	used to precede a comment, `#this is a comment`
<-	assigns a value or output from a function to a variable
$	used to refer to columns within a data frame, `dataframe$column`
@	used to refer to a *slotNames* within a spatial data frame, `spatialdataframe@data$column`
[,]	square brackets are used to refer to specific elements in a list or data frame. `pop2011[1,]` will show the first row and `pop2011[,1]` will show the first column.
abline()	adds vertical lines to a histogram, used to show classification breaks `abline(v = breaks$brks, col = "red")`
as.character()	converts a value to a string/text (e.g. from a number) *see also as.numeric()*
as.numeric()	converts a value to a number (e.g. from a string/text) *see also as.character()*
brewer.pal()	function to set the colours used and number of colours, `brewer.pal(6, "YlOrRd")`
c(,)	used to create a list, either numbers `c(1,2,3)` or strings (text) `c("Thomas","Richard","Harriet")`
cex	used to scale items in a plot, `legend(x=412971, y=439516, legend=levels(OA@data$oac_group), fill=my_colour, bty="n", cex=.5, ncol=2)`
cbind()	sticks two R data frames together, like *merge* but doesn't use a common attribute to match the rows
classIntervals()	function to set the data classification breaks, number of groups and classification method `classIntervals(LSOA@data$Age0to14pc,n=6,style="fisher")`
col	used within *`plot()`* function to set colours
colnames()	shows the names and numbers of the columns in the specified data set, `colnames(hp.data)`
data.frame()	function used to create a new *data frame*, particularly used with *match*.
dev.off()	used to stop PDF output, *see `pdf()`*
file.choose()	open a window to choose files interactively, `sthel <- readShapeSpatial(file.choose())`
findInterval()	function using breaks (from *classIntervals*) to set which data point is in which category
fix(dataframe)	edits a data frame in a new window, make sure you close this window before continuing
for()	begins a loop to make R repeat a command a set number of times, `for (i in 1:length(mapvariables))`
gBuffer()	function to create a buffer around a point object with the specified radius, `gBuffer(schools_SP_Leeds, width=1608, byid=TRUE)`
gCentriod()	function to calculate the *centroid* of a polygon, `gCentroid(OA, byid=TRUE)`
getwd()	shows the current working directory, see also *`setwd()`*
head()	used to show the first six rows of the data frame, `head(hp.data)`
header = TRUE	parameter used in `read.csv` to tell R to read the first line of the CSV file as the column headers (specifying `header = FALSE` will do the opposite)
hist()	Shows a histogram of the specified data, `hist(LSOA@data[,"Age0to4pc"])`
install.packages()	allows the user to install *packages* (also known as *libraries*) which is required the first time they are used on a computer, `install.packages("rgdal")`

is.na()	tests whether a value is listed as `NA` (not applicable) `schools_SP[!is.na(schools_SP@data$label),]` `!` inverts this (so `!is.na` keeps everything that is not `NA`)
lapply()	applies a function to each item in a list
legend()	adds a legend to an existing plot, `legend(x = 357000, y = 392000, legend = leglabs(breaks$b), fill = breaks$c, bty = "n", cex = 0.5)`
library()	loads the specified library, `library(rgdal)` (see also *`install.packages()`*)
locator()	tool to select coordinates on plot window, use mouse to select points, click Finish to end and the coordinates will appear in the *console*. Can specify number of points required (e.g. `locator(1)`) and the you do not need to click Finish
ls()	function to list all of the *variables* in the *environment*
match()	similar to *merge*, used to join two data frames together using a common attribute, particularly useful for spatial data, `data.frame(OA@data, pop2013[match(OA@data[,"code"], pop2013[,"OA11CD"]),])`
merge()	joins two data frames together using a common attribute or ID, `merge(sthel@data,hp.data,by.x="SP_ID",by.y="ID", all.x=TRUE)`
ncol()	lists the number of columns in a data frame, `ncol(hp.data)`
nrow()	lists the number of rows in a data frame, `nrow(hp.data)`
order()	reorders a data frame by the specified variable, `schools[order(schools$Easting,decreasing = TRUE),]`
over()	function to perform a point in polygon GIS analysis, `over(schools_SP,OA)`
par()	function to change the background colour of the plot window, `par(bg = "#696969")`
pdf()	used to output plot commands to a PDF file, always ends with `dev.off()`, `pdf(file="image.pdf")`
plot()	creates a map from a Spatial data frame, `plot(sthel)`
read.csv()	used to read CSV files (often converted from Excel) into R, `hp.data <- read.csv("hpdata.csv")`
readShapeSpatial()	reads in a shape file from the specified location, `sthel <- readShapeSpatial("sthel")`
round()	formatting function, used to round numbers, e.g. text in a legend, `round(breaks$brks, 1)`
rowSums()	adds up the values in the specified data frame rows, `rowSums(pop2011[,c(20,22)])`
rm()	used to delete specific variables, ***Warning*** *there is no "are you sure?" prompt,* `rm(price)`
setwd()	sets the working directory, `setwd("c:\folder")`
skip = n	parameter used in `read.csv` to tell R to skip the first n lines of the CSV file
slotNames()	returns the different types of slots within a SpatialDataFrame, `slotNames(LSOA)`
SpatialPolygonsRescale()	function to draw north arrow and scale bar on plot
spTransform()	changes a data set from one projection to another, `crime.pts <- spTransform(crime.pts, CRS(bng))`
substr()	function to extract characters from a string (text), `substr(OA@data$oac_group,1,1)`
title()	adds a title to an existing plot, `title('Burglary Rates per 10,000 Homes in St. Helens')`
text()	function to add text to a plot, `text(335379,380606,"0km", cex=.8)`
unzip()	unzips the specified zip file into the current working folder, `unzip("sthel.zip")`
View()	opens the dataframe in a new tab in RStudio, `View(sthel)`
which()	selects out data that match criteria, `which(OA@data$oac_group == "1")`
writeOGR()	saves a shapefile object including the projection, `writeOGR(crime.pts, "crime2.shp", crimes, driver = "ESRI Shapefile")`
writeSpatialShape()	saves a shapefile object, but does not include projection, *see writeOGR*

Glossary

Buffer	a circle around a point, where the radius of the circle is the buffer distance
Centroid	the centre point of a polygon
Console	window where you can type in commands for R to run, clears whenever you reopen R or run more than 1000 lines of commands *see also scripts*
Data frame	a format of storing spatial and non-spatial data in R
Environment	the area where the *variables* are stored, called *Workspace* by R, and can be shown by the command `ls()` can be saved
History	record of every command you have typed into R, can be saved
Join	the process of linking two data frames (usually an attribute data frame and a spatial data frame by a common attribute or ID)
Library	a set of commands that can be loaded and used in R (also known as *package*)
Package	a set of commands that can be loaded and used in R (also known as *library*)
Script	a series of R commands that can be run on demand (filename usually ends with .R) useful for rerunning commands
R	the main program used to run R commands, see also *RStudio*
RStudio	an interface that runs on top of R, allowing easier management of *variables*, *scripts* and *plots*
Shapefile	a type vector of spatial data file, consisting of one of points, lines OR polygons; consists of multiple files (between 4 and 6 files, with extensions of .shp, .dbf, .shx, .prj)
Variable	the way R stores values and data, assigned using the `<-` command
Workspace	the area where the variables are stored, called *Environment* by *RStudio* and shown in the top right hand corner

Error Messages

Error messages are commonly caused by incorrect spellings or missing something small from the code. Below are some error messages that may occur.

`Error in fix.by(by.y, y) : 'by' must specify a uniquely valid column`	Common with the `merge()` function, where R can't find the fields/columns you have specified
`Error: could not find function "img"`	Incorrect spelling should be `png`

`Error in plot(LSOA) :`
`  error in evaluating the argument 'x' in selecting a method for function 'plot':`
`Error: object 'LSOA' not found`
	R can't find the variable `LSOA`, check the spelling

By Nick Bearman, 19/04/2017