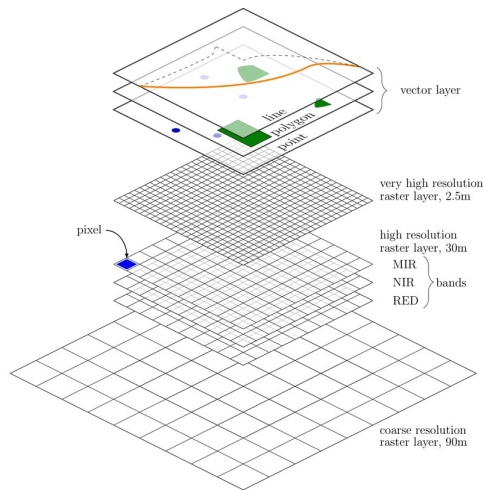


Spatial R Cheat Sheet

Remote Sensing and GIS functions



book.ecosens.org

last updated: 29th September, 2015

Packages

Packages which are used in the book are listed here, more relevant packages are however available within R

RStoolbox	various RS functions
raster	for raster data manipulation
rgdal	data import and export, projections
sp	for vector data manipulation
rgeos	geometry commands
wrspathrow	provides Landsat WRS-2 information
gfcanalysis	access to Forest Cover Change product
modis	download and analyse MODIS data
bfast	analyse time-series data
dismo	species distribution modelling
move	access and analyse movement data

More spatial R packages are listed here:
cran.r-project.org/web/views/Spatial.html

Relevant commands are listed below, actual syntax needs to be checked within the manual pages of each command.

Raster

Raster data manipulation is similar to a spreadsheet or matrix manipulation but with coordinates and projections, hence various also not explicitly spatial commands can be applied. Here we mainly list commands designed for spatial data handling.

Import and export

<code>raster()</code>	import (or generate) one raster layer
<code>brick()</code>	import raster with multiple layers
<code>writeRaster()</code>	export raster data to file
<code>writeFormats()</code>	list of supported raster file types
<code>getData()</code>	retrieves DEM and climate data directly from the web

Information

<code>print()</code>	prints raster metadata
<code>click()</code>	interactively query raster plot
<code>hist()</code>	histogram of raster values per layer
<code>cellStats()</code>	summary statistics of single layers
<code>summary()</code>	summary statistics
<code>extent()</code>	extent of raster data set
<code>ncell()</code>	number of cells (of one layer)
<code>nlayers()</code>	number of bands
<code>names()</code>	prints layer names
<code>str()</code>	print the data structure
<code>NValue()</code>	get or set background values

Visualisation

<code>ggR()</code> , <code>ggRGB()</code>	ggplot2 plotting commands implemented in RStoolbox
<code>plot()</code> , <code>plotRGB()</code>	raster plot and RGB plot. Usefull arguments: <code>y=bandnumber</code> , <code>add=TRUE</code> (overlay multiple plots)
<code>image()</code> , <code>spplot()</code>	alternative plotting commands
RasterVis package	
<code>levelplot()</code>	fancy way to plot raster data information
<code>densityplot()</code>	raster value density plot
<code>bwplot()</code>	violin plot of raster data values
<code>hovmoller()</code>	spatio-temporal plotting options

Projections

<code>projection()</code>	query or set projection (does NOT reproject)
<code>projectRaster()</code>	reprojects raster to new coordinate system

Data manipulation

Most raster commands will output a file to a chosen location, if `filename=` is specified. Otherwise it will use temp files.

<code>stack()</code>	stack different raster layers together
<code>addLayer()</code> ; <code>dropLayer()</code>	add/drop a raster layer
<code>crop()</code>	crop raster set to smaller extent
<code>drawExtent()</code>	draw extent on a plot for e.g. inclusion in <code>crop(raster,extent)</code>
<code>drawPolygon()</code>	draw polygon on a plot - alternative to <code>drawExtent()</code>
<code>mask()</code>	masking of background values
<code>merge()</code> ; <code>mosaic()</code>	combine raster tiles to a raster with larger extent
<code>extract()</code>	extract values from Raster objects, using points or polygons

Basic Operations

<code>raster*2/raster2</code>	any basic operation, more efficient: apply a function to raster data and apply a function which uses multiple bands, e.g. to calculate NDVI
<code>calc()</code>	moving window operations
<code>overlay()</code>	calculate distance to closest feature, e.g. distance to water
<code>focal()</code>	calculate terrain attributes from DEM, e.g. slope
<code>distance()</code>	zonal statistics, for classified raster
<code>terrain()</code>	reclassifies raster values
<code>zonal()</code>	substitutes values
<code>reclassify()</code>	resampling of raster to raster
<code>subs()</code>	aggregation of cells
<code>resample()</code>	disaggregation of cells
<code>aggregate()</code>	converts a raster to vector points
<code>disaggregate()</code>	converts a raster to polygons
<code>rasterToPoints()</code>	converts raster values to contour
<code>rasterToPolygons()</code>	address specific raster layer, e.g. <code>myRaster[[1]]</code> for first layer of <code>myRaster</code>
<code>rasterToContour()</code>	boolean operation, output is binary
<code>[[]]</code>	replace all values smaller then 50 with 0
<code>x <- raster > 50</code>	values in r1 whose values are equal 50 are replaced by the corresponding values of r2
<code>raster[raster <= 50] <- 0</code>	
<code>r1[r1==50] <- r2[r1==50]</code>	

<code>sampleRandom()</code>	random sample from cell values
<code>sampleRegular()</code>	regular sample from cell values
<code>sampleStratified()</code>	stratified sample from cell values

Remote Sensing Operations

<code>superClass()</code>	supervised classification
<code>unsuperClass()</code>	unsupervised classification
<code>rasterCVA()</code>	change vector analysis for change detection
<code>rasterPCA()</code>	principal component analysis
<code>tasseledCap()</code>	tasseled cap transformation
<code>spectralIndices()</code>	computation of spectral indices
<code>fCover()</code>	analysis of fractional cover
<code>cloudMask()</code>	cloud masking

Vector

Vector data often come in shp format including a variety of auxiliary files. All of them are relevant and are needed for further analysis. Note that readShapePoly() etc. from package maptools do NOT automatically read projection information from shapefiles. It is recommended to use readOGR() instead.

Import and Export

readOGR()	import vector file
writeOGR()	export vector file
ogrDrivers()	list supported file formats

Information

plot()	vector plot. add=TRUE overlays multiple plots, e.g. combine with raster data
summary()	metadata and data summary
extent()	extent/bounding box of vector data
coordinates()	sets spatial coordinates to create spatial data, or retrieves spatial coordinates

Projections

projection()	query or set projection (does NOT reproject)
spTransform()	reproject vector data to new coordinate system

Data Manipulation

Check out the functions in the rgeos package, which provides most of the classical vector GIS operations such as buffers etc.

subset()	subset spatial data, based on a condition, e.g. keep only certain points
merge()	Merge a Spatial object having a data.frame (i.e. merging of non-spatial attributes)
over()	spatial overlay for points, grids and polygons
rasterize()	Rasterize points, lines, or polygons
distanceFromPoints()	computes the distance to points, output is a raster
extract()	extracts raster values behind points, lines or polygons
gIntersection()	intersection of vector data sets
gBuffer()	Buffer Geometry
gUnion()	...

Spatial Modeling

kfold()	partitioning of data set for training/validation purpose
evaluate()	cross-validation of models with presence/absence data
randomForest()	fits a randomForest model
maxent()	executes Maxent from R
gam()	fits a GAM
predict()	predicts statistical model into space (raster)

Movement Analysis

For most of the following commands the data sets need to be converted to a specific format. The commands are mainly provided in the “move” package but same names might exist in other packages. Use move::spTransform() to address the move command. Please consider checking the AniMove R cheat sheet (www.animove.org).

show()	summary of the move object
as()	coerce movement between object types
angle()	extracts turning angles from a move object
speed()	extracts speed from a move object
distance()	extracts distance between locations from a move object
time.lag()	extracts time lag between locations from a move object
spTransform()	changes the projection of a move object to a default of Azimuthal Equidistance
mcp()	calculates minimum convex polygons for SpPdf
kernelUD()	calculates a kernel density surface for SpPdf
brownian.bridge()	calculates constant variance Brownian bridges
brownian.bridge.dyn()	calculates dynamic Brownian bridges
move()	import of movement data sets from movebank.org
moveStack()	stacks multiple animal tracks
split()	splits stack into single move objects
movebankLogin()	stores movebank.org credentials
searchMovebankStudies()	reports the studies in movebank.org matching search criteria
getMovebankData()	import tracks directly from movebank.org

Miscellaneous

Some useful commands which are related to spatial data analysis.

gmap()	get google maps for your plot
geocode()	geocoding in R
complete.cases()	returns only cases with no missing values
gridSample()	sample point from a grid e.g. just one point per pixel
function(...){.. return(...) if (...) {...} else{...} for (...) {...} while (...) { ...}	generates a defined functions returns the output of a function if else statement for loop while statement

Further Packages

rNOMADS	data retrieval from NOAA, global and regional weather models
...	access and analyse movement data
bcpa	analyse movement tracks

compiled by:

Martin Wegmann (martin.wegmann@ecosens.org) and Benjamin Leutner (benjamin.leutner@ecosens.org)
2015

Compiled for the book “Remote and GIS for Ecologists - Using Open Source Software” book.ecosens.org

