AniMove Cheat Sheet
for spatial data handling, remote sensing, spatial statistics and animal movement analysis

www.animove.org
last updated: 11th May, 2014

Packages

raster	for raster data manipulation
sp	for vector data manipulation
rgdal	data import and export, projections
rgeos	geometry commands
spdep	spatial dependence
dismo	species distribution modelling
move	access and analyse movement data
bcpa	analyse movement tracks

further relevant packages:

spatstat	spatial statistics
gstat	geostatistics
geoR	geostatistical analysis
gdistance	distances on geographical grids
spsurvey	sampling functionality
trip	sp class extension for track analysis
randomForest	random Forest implementation
mgcv	GAM implementation
lme4	mixed-effects model
landsat	Landsat specific analysis
spgrass6	interaction with GRASS

visualisation packages:

maptools	handling spatial objects
maps	map display
mapproj	map projections
mapdata	supplements to maps
rasterVis	enhanced raster visualisation
ggplot2	for more fancy plots
reshape2	flexibly reshape 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 explictly spatial commands can be applied. Here we mainly list commands designed for spatial data handling.

Import and export

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

Information

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

Visualisation

plot(), plotRGB()	raster plot and RGB plot. Usefull arguments: y=bandnumber, add=TRUE (overlay multiple plots)
image(), spplot()	alternative plotting commands
levelplot()	fancy way to plot raster data information
densityplot()	raster value density plot
bwplot()	violin plot of raster data values
hovmoller()	spatio-temporal plotting options
streamplot()	plotting of streamlines

Projections

projection()	query or set projection (does NOT reproject)
projectRaster()	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.

stack()	stack different raster layers together
addLayer(); dropLayer()	add/drop a raster layer
crop()	crop raster set to smaller extent
drawExtent()	draw extent on a plot for e.g. inclusion in crop(raster,extent)
mask()	masking of background values
merge(); mosaic()	combine raster tiles to a raster with larger extent
extract()	extract values from Raster objects, using points or polygons
raster*2/raster2	any basic operation, more efficient:
calc()	apply a function to raster data and
overlay()	apply a function which uses multiple bands, e.g. to calculate NDVI
focal()	moving window operations
distance()	calculate distance to closest feature, e.g. distance to water
terrain()	calculate terrain attributes from DEM, e.g. slope
zonal()	zonal statistics, for classified raster
reclassify()	reclassify raster values
subs()	substitutes values
resample()	resampling of raster to raster
aggregate()	aggregation of cells
disaggregate()	disaggregation of cells
rasterToPoints()	converts a raster to vector points
rasterToPolygons()	converts a raster to polygons
rasterToContour()	converts raster values to contour
[[ ]]	address specific raster layer, e.g. myRaster[[1]] for first layer of myRaster
x <- raster > 50	boolean operation, output is binary
raster[raster <= 50] <- 0	replace all values smaller then 50 with 0
r1[r1==50] <- r2[r1==50]	values in r1 whose values are equal 50 are replaced by the corresponding values of r2
sampleRandom()	random sample from cell values
sampleRegular()	regular sample from cell values
sampleStratified()	stratified sample from cell values

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 reccomended 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

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
pls()	fits a partial least squares model
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.

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
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()	claculates constant variance Brownian bridges
brownian.bridge.dyn()	calculates dynamic Brownian bridges
LoCoH.k()	calculates local convex hulls using k neighbours
LoCoH.r()	calculates local convex hulls using a radius of r
LoCoH.a()	calculates local convex hulls using an adpative radius

Miscellaneous

Some useful commands which are related to spatial data analysis.

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