

# AniMove Cheat Sheet

## for spatial data handling, remote sensing, spatial statistics and animal movement analysis



[www.animove.org](http://www.animove.org)  
last updated: 11<sup>th</sup> May, 2014

### Packages

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

further relevant packages:

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

visualisation packages:

<b>maptools</b>	handling spatial objects
<b>maps</b>	map display
<b>mapproj</b>	map projections
<b>mapdata</b>	supplements to maps
<b>rasterVis</b>	enhanced raster visualisation
<b>ggplot2</b>	for more fancy plots
<b>reshape2</b>	flexibly reshape data

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

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

#### Information

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

#### Visualisation

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

#### Projections

<b>projection()</b>	query or set projection (does NOT reproject)
<b>projectRaster()</b>	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.

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

## 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()	calculates 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 adaptive 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

compiled by: Wegmann, Leutner, Bevanda, Horning & Safi 2014

<http://www.animove.org>

