Data cleaning and exploration with speciesgeocodeR

Alexander Zizka

2015-10-06

Contents

Introduction

SpeciesgeocodeR is an R-package for the exploration, cleaning and preparation of large scale species distribution data for biogeographic and macro-evolutionary analyses. The package provides functions for the automated cleaning of coordinates, the fast and reproducible classification of point occurrences to geographic areas (including elevation and minimum occurrence thresholds) with an export of the results in nexus or BioGeoBEARS format, as well functions to explore the occurrence data to facilitate data-cleaning and interpretation. This includes automated summary statistics as well as maps and the calculation and visualization of coexistence matrices, species richness patterns and species ranges. The package is designed to be user-friendly: formative output can be calculated directly from text files with on single command. SpeciesgeocodeR has particularly been tested for global scale data sets.

This tutorial provides instructions for the major functionality of the package using an example data set of Lemur distributions from the Global Biodiversity Information Facility (www.gbif.org). There is a detailed description of each function available with `?` plus function name (e.g. `?GeoClean`). If you have no experience with R you might want to have a look at Crawley (2012). Alternatively you can use the python version of SpeciesgeocodeR (Töpel et al. 2014) for some functionality of the package. SpeciesgeocodeR is mainly based on functions of the maps (Becker et al. 2013), maptools (R. Bivand and Lewin-Koh 2013), raster (Hijmans 2014), rgeos (R. Bivand and Rundel 2014) and sp (E. J. Pebesma and Bivand 2005; R. S. Bivand, Pebesma, and Gomez-Rubio 2013) packages. Please ask questions questions and report bugs to speciesgeocodeR@googlegroups.

Features of the package

- automated cleaning of geographic data sets
- classification of point occurrences to geographic areas for large datasets
- inclusion of multiple elevation thresholds and a minimum occurrence threshold into the classification
- output as nexus and BioGeoBEARS format
- summary statistics and -maps
- easy-to-use wrapper function
- coexistence matrix calculation
- species richness maps
- batch calculation of species ranges and range sizes

Example data

We will use distribution data of Madagascan Lemur species and a simplified version of the WWF biomes of Madagascar as examples in this tutorial. The data are distributed in R data format or as .txt files with the package. Use `data(package = "speciesgeocodeR")` to see all example dataset. Most functions of speciesgeocodeR can accept file paths as arguments. The following lines of code will load the example occurrences and areas, and store them in R objects called "lemurs" and "mdg_poly".

```
#occurrences
data(lemurs)

#areas
data(mdg_poly)
```

You can load your own data from text or shape files by replacing the "system.file("extdata","lemurs.txt", package ="speciesgeocodeR")" or "system.file("extdata","mdg_biomes_simple.txt", package ="speciesgeocodeR")" in the next example by the path to your input files.

```
#occurrences
occ <- read.table(system.file("extdata","lemurs.txt", package = "speciesgeocodeR"),
                  row.names = NULL)

#areas
pol <- read.table(system.file("extdata","mdg_biomes_simple.txt",
                              package = "speciesgeocodeR"), row.names = NULL)
```

Automated cleaning of geographic data

The `GeoClean` function can be used to check geographic occurrence data for coordinate integrity and to flag potential problems that are known to occur with data from public databases. Load the data from a .txt file with three obligatory columns, named "identifier" (species name), "XCOOR" (decimal longitude) and "YCOOR" (decimal latitude) and an optional "country"" column (ISO2 or ISO3 code) for more complex tests. For example

```
##           identifier    XCOOR    YCOOR country
## 1 Cheirogaleus major 47.41047 -21.2649     MDG
## 2 Cheirogaleus major 49.09200 -18.0480     MDG
## 3 Cheirogaleus major 48.46700 -18.7020     MDG
## 4 Cheirogaleus major 48.42700 -18.7920     MDG
## 5 Cheirogaleus major 48.81300 -18.6950     MDG
## 6 Cheirogaleus major 48.57800 -18.1980     MDG
```

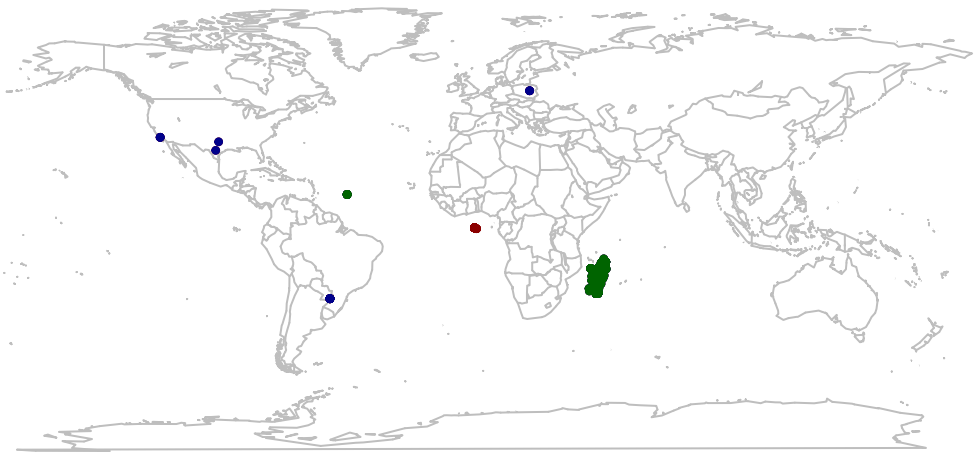
We will test the function on the lemurs_test dataset, which deliberately includes problematic coordinates. We will first perform some simple test for coordinate validity. Each argument of `GeoClean` represents one test, see `?GeoClean` for details.

```
data(lemurs_test)

#A vector, FALSE if one of the test failed
tt <- GeoClean(lemurs_test, isna = TRUE, isnumeric = TRUE,
               coordinatevalidity = TRUE, containszero = TRUE,
```

```
                zerozero = TRUE, zerozerothresh = 1,
                latequallong = TRUE)

#Alternatively a data.frame with results from each test; outp = "detailed"
tt.long <- GeoClean(lemurs_test, isna = TRUE, isnumeric = TRUE,
                    coordinatevalidity = TRUE, containszero = TRUE,
                    zerozero = TRUE, zerozerothresh = 1,
                    latequallong = TRUE, outp = "detailed")

#Or a data.frame containing only unproblematic records; outp = "cleaned"
simple.clean <- GeoClean(lemurs_test, isna = TRUE, isnumeric = TRUE,
                         coordinatevalidity = TRUE, containszero = TRUE,
                         zerozero = TRUE, zerozerothresh = 1,
                         latequallong = TRUE, outp = "cleaned")
```

If the coordinates are all valid, more complex tests can be performed. For example, if points fall within the country indicated in the country column, if the points have been assigned to country centroid or the country capital, or if the points have been assigned to the GBIF headquarters.

```
library(maptools)

data(countryref)
data(wrld_simpl)

complex.clean <- GeoClean(simple.clean, capitalcoords = TRUE, countrycheck = TRUE,
                          polygons = wrld_simpl, GBIFhead = TRUE, countrycentroid = TRUE,
                          referencecountries = countryref, capthresh = 0.5,
                          contthresh = 0.5, outp = "cleaned")

map("world", col = "grey")
points(lemurs_test$XCOOR, lemurs_test$YCOOR, col = "darkred", pch = 20, cex = .7)
points(simple.clean$XCOOR, simple.clean$YCOOR, col = "darkblue", pch = 20, cex = .7)
points(complex.clean$XCOOR, complex.clean$YCOOR, col = "darkgreen", pch = 20, cex = .7)
```



Classifying species to areas

There are two functions that can perform the occurrence to area classification : `SpeciesGeoCoder`, `SpGeoCod`.

Input data format

SpeciesgeocodeR can use different input formats.

1. Occurrences as `data.frame`, areas as `spatialPolygons`

The data.frame of species occurrences needs three columns named "identifier" (species name, each line one occurrence), "XCOOR" (decimal longitude) and "YCOOR" (decimal latitude). Each row is one occurrence. For the area input you need to indicate the column with the names of single polygons by the `areanames` argument. Find the respective column using the `head` function.

```
data(lemurs)
data(mdg_biomes)

head(mdg_biomes@data)
```

```
##   id         name
## 0  1 Moist Forest
## 1  2   Dry Forest
## 2  3   Shrublands
```

```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name")
```

2. Occurrences as .txt and areas as .shp

Provide occurrences as .txt file and areas as a .shp file in the working directory. The .txt file needs three columns named "identifier" (species name, each line one occurrence), "XCOOR" (decimal longitude) and "YCOOR" (decimal latitude). You can use any shape file, but you need to indicate the name of the column naming the polygons via the `areanames` argument.

```
outp <- SpGeoCod(system.file("extdata","lemurs.txt", package = "speciesgeocodeR"),
                 system.file("extdata","mdg_biomes_simple.shp", package = "speciesgeocodeR"),
                 areanames = "name")
```

3. Occurrences and areas as .txt files

Provide occurrences and areas as .txt files in the working directory. The species occurrence file needs three obligatory columns named "identifier" (species name), "XCOOR" (decimal longitude) and "YCOOR" (decimal latitude) for the occurrence file. The area file can be provided in a similar style: "identifier" (polygon name, each line is one vertex), "XCOOR" (decimal longitude) and "YCOOR" (decimal latitude). To load your data from text files replace the "system.file("extdata","lemurs.txt", package ="speciesgeocodeR")" or "system.file("extdata","mdg_biomes_simple.txt", package ="speciesgeocodeR")" in the next examples by the path to your input files.

```
#Example for occurrence input file
```

```
##      row.names identifier    XCOOR    YCOOR
## 1 Cheirogaleus      major 47.41047 -21.2649
## 2 Cheirogaleus      major 49.09200 -18.0480
## 3 Cheirogaleus      major 48.46700 -18.7020
## 4 Cheirogaleus      major 48.42700 -18.7920
## 5 Cheirogaleus      major 48.81300 -18.6950
## 6 Cheirogaleus      major 48.57800 -18.1980
```

```
#Example for area input file
```

```
##     identifier    XCOOR     YCOOR
## 1 Moist Forest 50.22007 -15.98834
## 2 Moist Forest 47.16053 -24.92900
## 3 Moist Forest 46.68460 -25.20096
## 4 Moist Forest 46.61661 -24.40208
## 5 Moist Forest 46.87157 -24.28310
## 6 Moist Forest 46.37865 -23.92615
```

```
outp <- SpGeoCod(system.file("extdata","lemurs.txt", package = "speciesgeocodeR"),
                 system.file("extdata","mdg_biomes_simple.txt", package = "speciesgeocodeR"))
```

4. Occurrences from GBIF

Use GBIF occurrence data by indicating species names as input. In this case occurrence data will automatically be downloaded from GBIF and plug in directly into the analyses (**This will download data from the Internet**). Coordinates cleaning can be included via the "cleaning"" argument.

```
outp <- SpGeoCod(c("Indri indri", "Lemur catta" ), pol, areanames = "name")
```

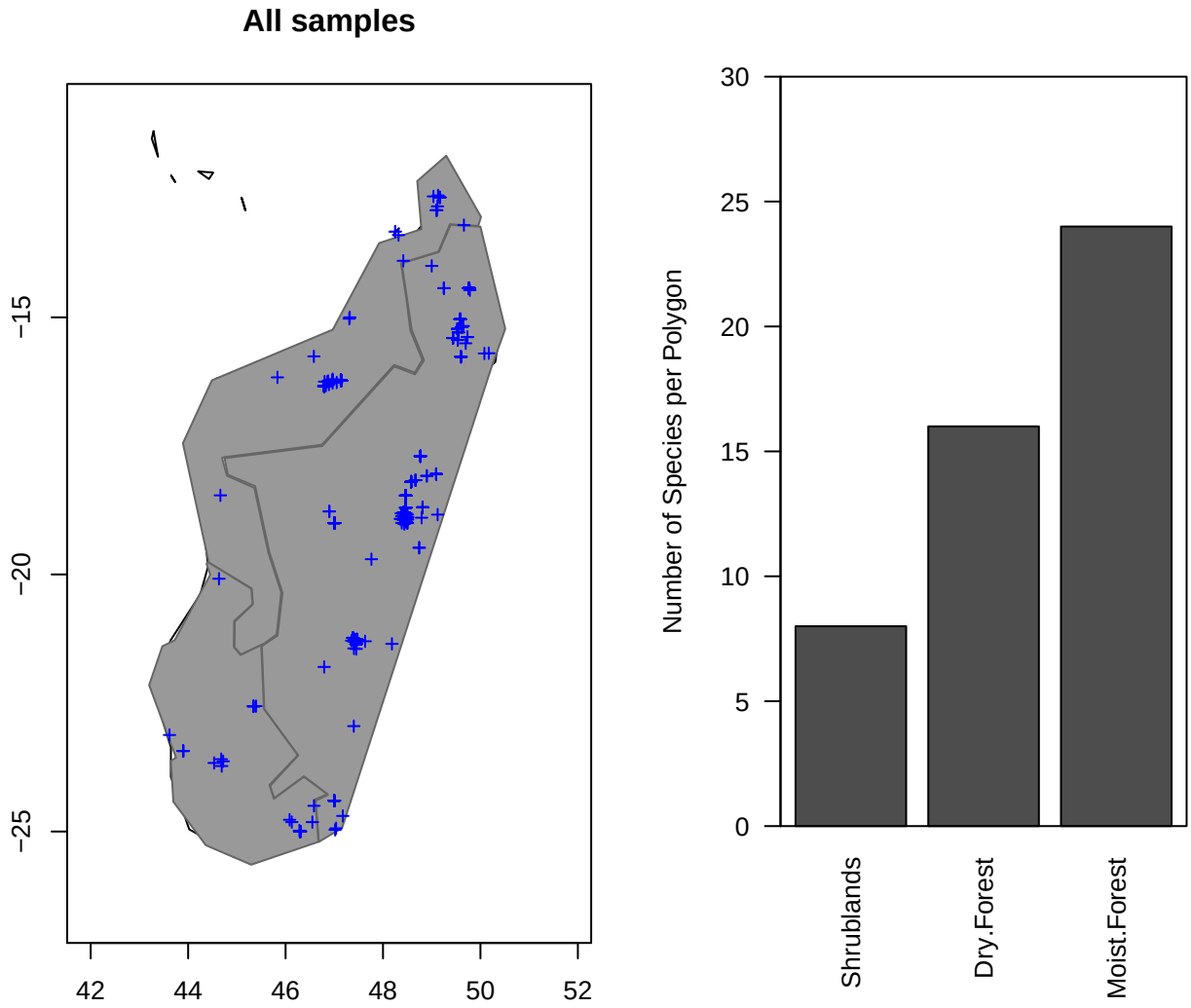
Classifying species to areas

The `SpGeoCod` function classifies all occurrences to the respective areas and calculates summary statistics. For larger data sets this might take a moment. The results can be explored with `summary` or `plot`.

```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name")

summary(outp)
```

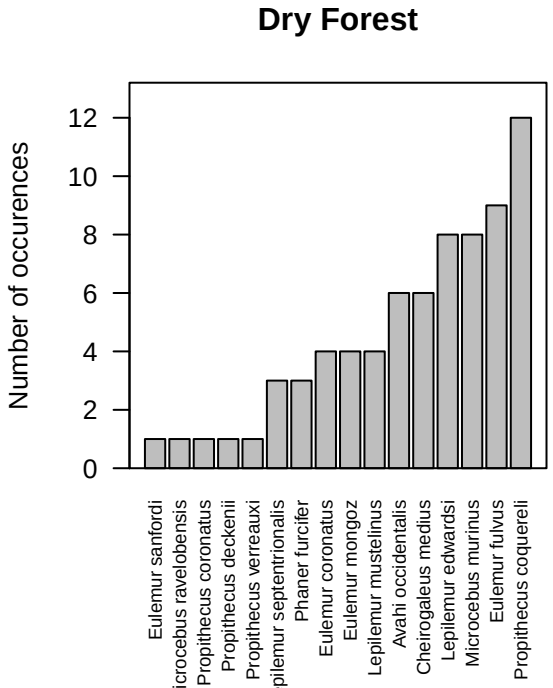
```
## $overall
## [1] "39 species with 403 occurrence points and 3 input polygons."
##
## $species_coordinates
##      XCOOR           YCOOR
##  Min.   :43.62   Min.   :-25.00
##  1st Qu.:47.00   1st Qu.:-19.59
##  Median :48.25   Median :-18.47
##  Mean   :47.98   Mean   :-18.25
##  3rd Qu.:49.09   3rd Qu.:-15.70
##  Max.   :50.17   Max.   :-12.63
##
## $polygons
## [1] "Moist Forest" "Dry Forest"   "Shrublands"
##
## $species_number_per_polygon
##        [,1]
## Mean     16
## Median   16
## Max      24
## Min       8
##
## $not_classified_samples
## [1] "4 occurrences did not fall in any polygon"
```

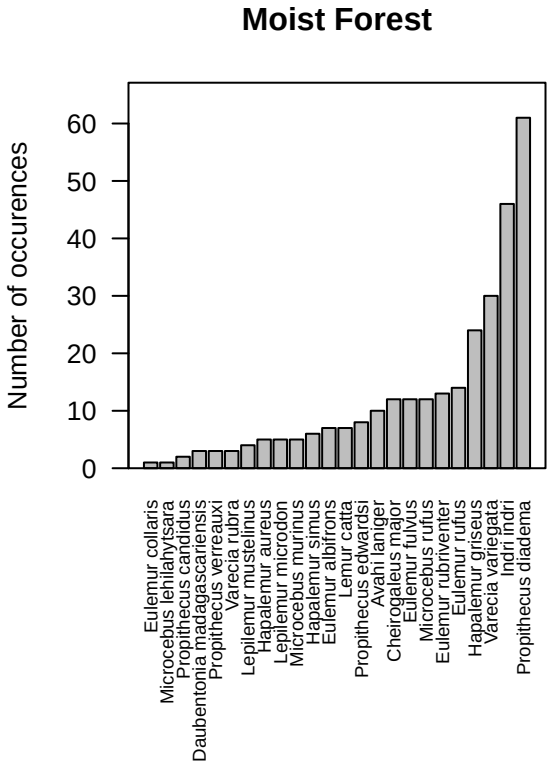
```
plot(outp, mar = c(8,4,4,4))
```

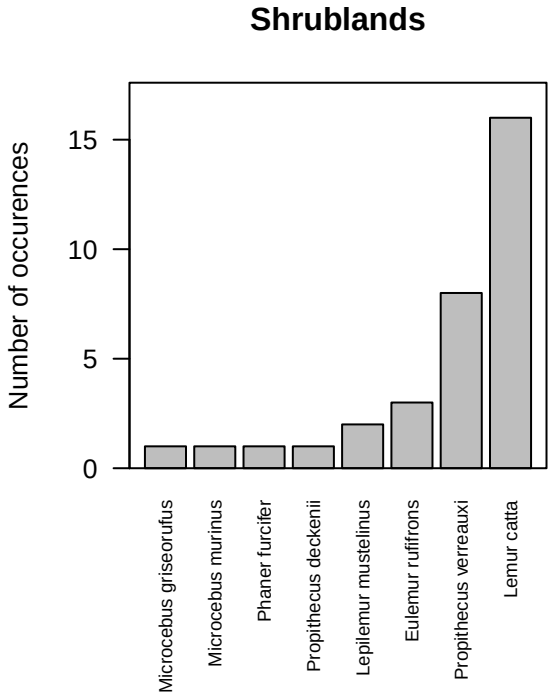


The `plot` function can visualize multiple results, depending on the `plottype` argument, including species number per area, occurrences per species per area, occurrences per area per species or maps for each species and each area. See `?plot.spgeoOUT` for details. For example species occurrences per polygon:

```
par(mfrow = c(2,2), mar = c(8,4,4,4))
plot(outp, plottype = "polygons")
```

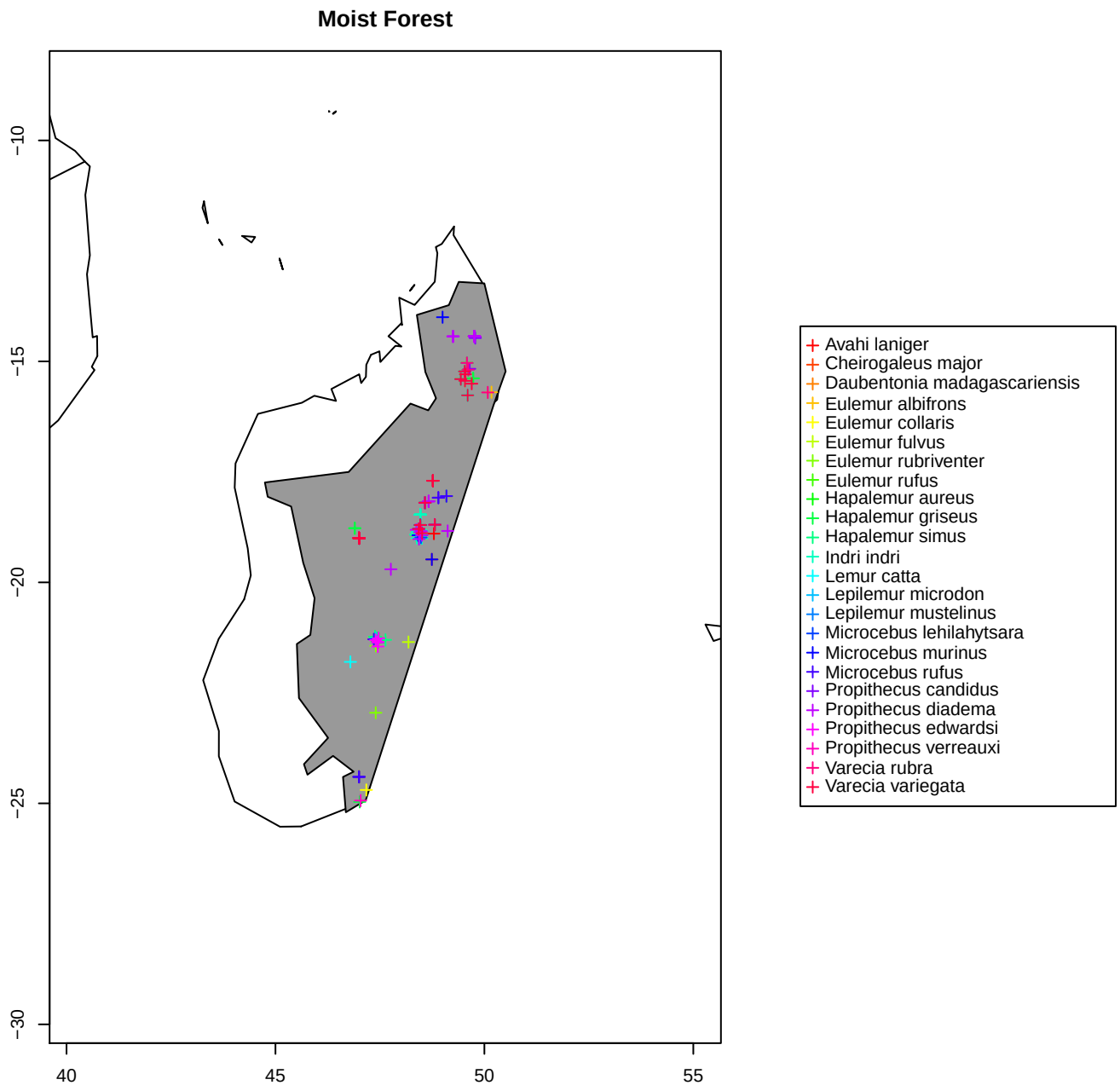


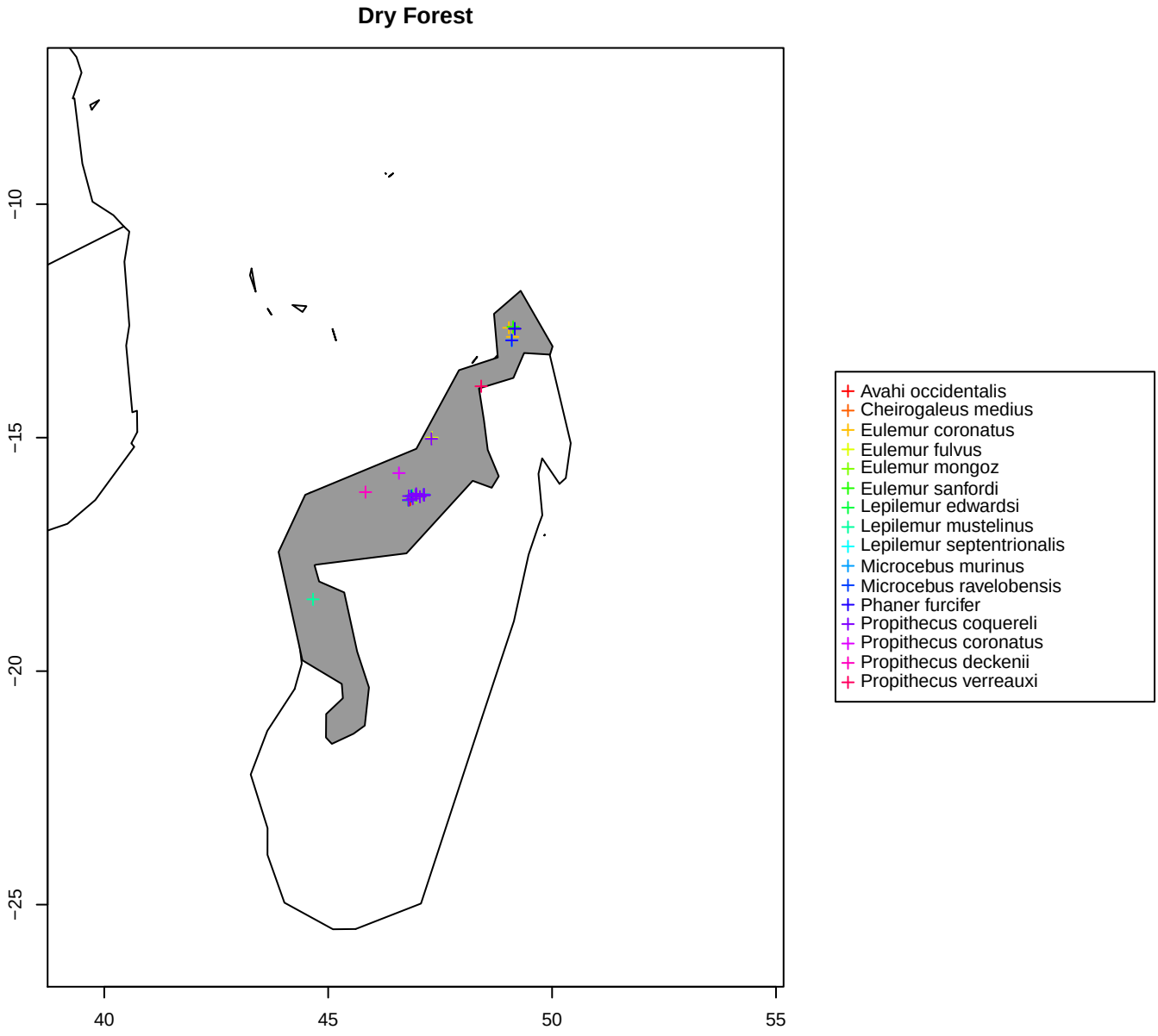


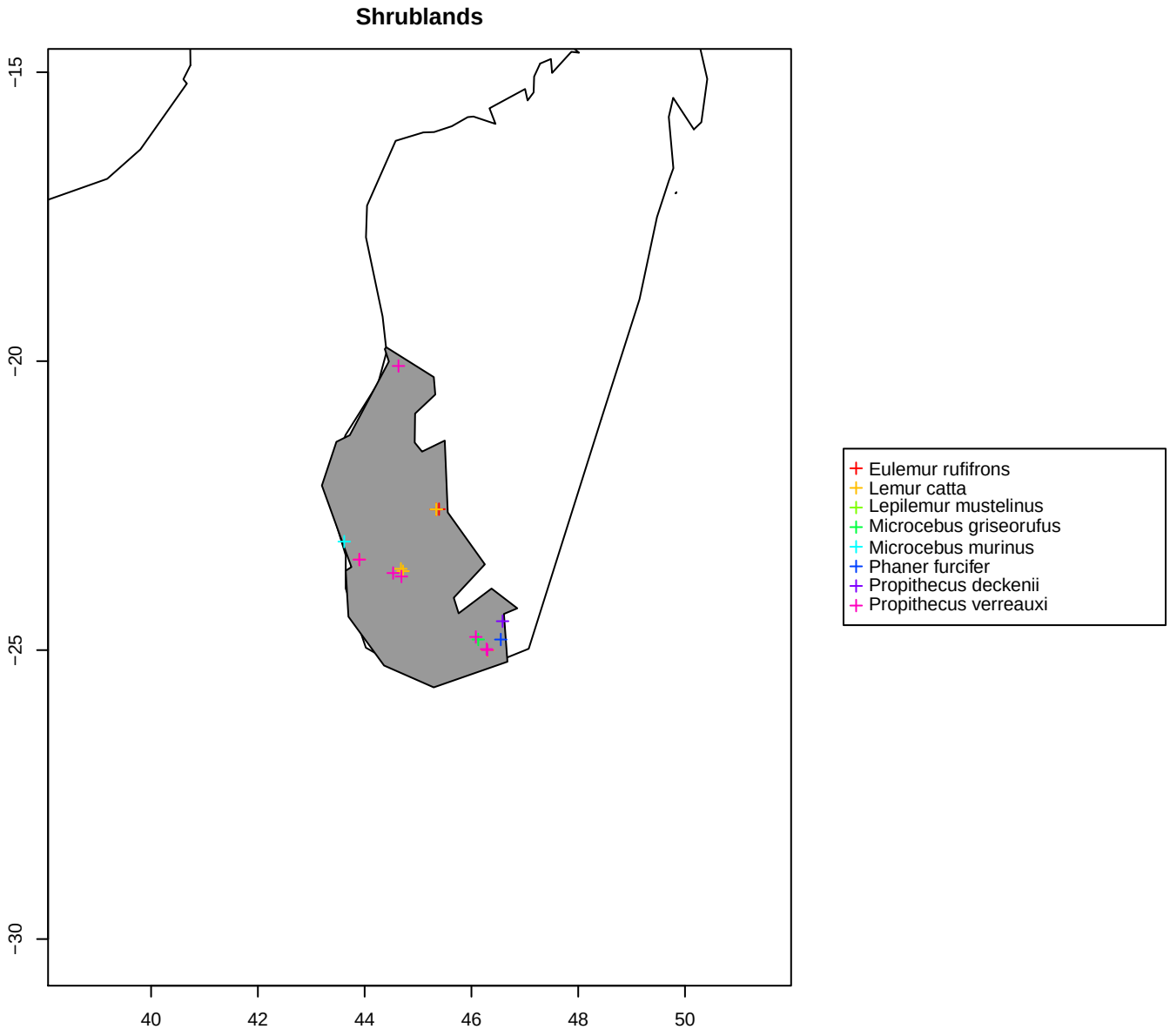


Or a map of occurrences per polygon:

```
par(mfrow = c(2,2), mar = c(8,4,4,4))
plot(outp, plottype = "mappolygons")
```




Dry Forest
−10
−15
−20
−25
40
45
50
55
Avahi occidentalis
Cheirogaleus medius
Eulemur coronatus
Eulemur fulvus
Eulemur mongoz
Eulemur sanfordi
Lepilemur edwardsi
Lepilemur mustelinus
Lepilemur septentrionalis
Microcebus murinus
Microcebus ravelobensis
Phaner furcifer
Propithecus coquereli
Propithecus coronatus
Propithecus deckenii
Propithecus verreauxi



SpeciesGeoCoder

The `SpeciesGeoCoder` function provides a wrapper to run an entire classification from text input to the standard output files in the working directory with one line of code.

```
SpeciesGeoCoder(lemurs, mdg_biomes, areanames = "name")
```

Output data

Speciesgeocoder can provide the species to area classification in various formats used in phylogenetic analyses. The `WriteOut` function provides a generic way to write results to the working directory. The type of output can be specified via the `writetype` argument. See `?WriteOut` for details. All results can be written out with `writetype = 'all'`.

Nexus file

The area classification of species can be written to a file in nexus trait format, which can be used by phylogenetic software (e.g. mesquite).

```
WriteOut(outp, writetype = "nexus")
```

BioGeoBEARS

The species area classification can be written out in the format of the geography input for a BioGeoBEARS analysis. Alternatively the Spgc2BioGeoBEARS function returns a R object in the BioGeoBEARS input format from an spgeoOUT object.

```
WriteOut(outp, writetype = "BioGeoBEARS")
bgb <- Spgc2BioGeoBEARS(outp)
head(bgb)
```

```
## $BioGeoBEARS_command
## [1] "38\t3\t(a b c)"
##
## $BioGeoBEARS_matrix
##                               Dry.Forest Moist.Forest Shrublands
## Avahi laniger                          0            1          0
## Avahi occidentalis                     1            0          0
## Cheirogaleus major                     0            1          0
## Cheirogaleus medius                    1            0          0
## Daubentonia madagascariensis           0            1          0
## Eulemur albifrons                      0            1          0
## Eulemur collaris                       0            1          0
## Eulemur coronatus                      1            0          0
## Eulemur fulvus                         1            1          0
## Eulemur mongoz                         1            0          0
## Eulemur rubriventer                    0            1          0
## Eulemur rufifrons                      0            0          1
## Eulemur rufus                          0            1          0
## Eulemur sanfordi                       1            0          0
## Hapalemur aureus                       0            1          0
## Hapalemur griseus                      0            1          0
## Hapalemur simus                        0            1          0
## Indri indri                            0            1          0
## Lemur catta                            0            1          1
## Lepilemur edwardsi                     1            0          0
## Lepilemur microdon                     0            1          0
## Lepilemur mustelinus                   1            1          1
## Lepilemur septentrionalis              1            0          0
```

```
## Microcebus griseorufus              0           0           1
## Microcebus lehilahytsara            0           1           0
## Microcebus murinus                  1           1           1
## Microcebus ravelobensis             1           0           0
## Microcebus rufus                    0           1           0
## Phaner furcifer                     1           0           1
## Propithecus candidus                0           1           0
## Propithecus coquereli               1           0           0
## Propithecus coronatus               1           0           0
## Propithecus deckenii                1           0           1
## Propithecus diadema                 0           1           0
## Propithecus edwardsi                0           1           0
## Propithecus verreauxi               1           1           1
## Varecia rubra                       0           1           0
## Varecia variegata                   0           1           0
```

Summary tables

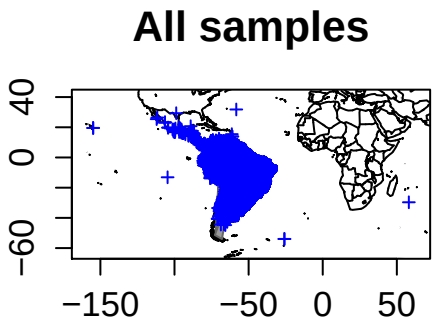
Summary tables of species occurrences in tab-delimited text format can be written out using `writetype = "statistics"`

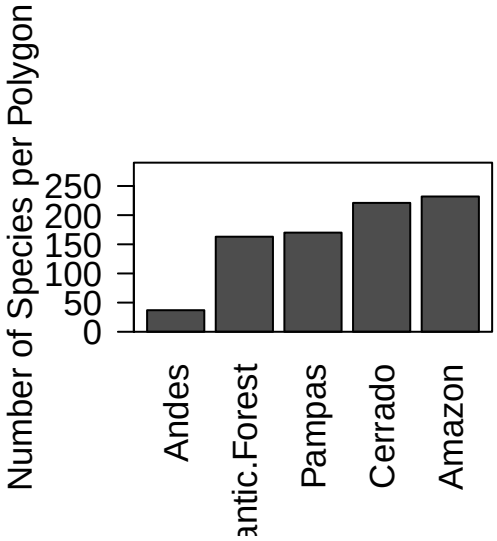
```
WriteOut(outp, writetype = "statistics")
```

Summary graphs and maps

Graphs visualizing the species occurrence per area (for all species and areas) and maps showing species occurrences can be written out as .pdf files using `writetype = "statistics"`.

```
WriteOut(outp, writetype = "graphs")
WriteOut(outp, writetype = "maps")
```





Data exploration features

Minimum occurrence threshold

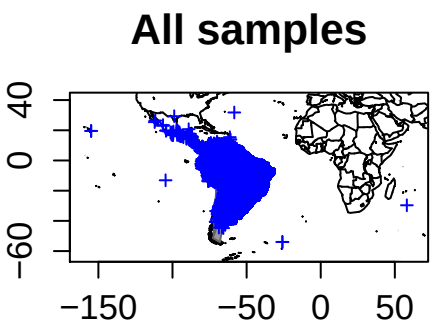
For widespread species, or if the data quality is questionable, it can be of interest to set a minimum number of occurrences that a species needs to be counted as present in a given area. The `occ.thresh` argument allows you to set a minimum percentage of occurrences. For example, we can code lemur species as present in a biome only if at least 10 percent of the occurrences of this species occur in the given biome:

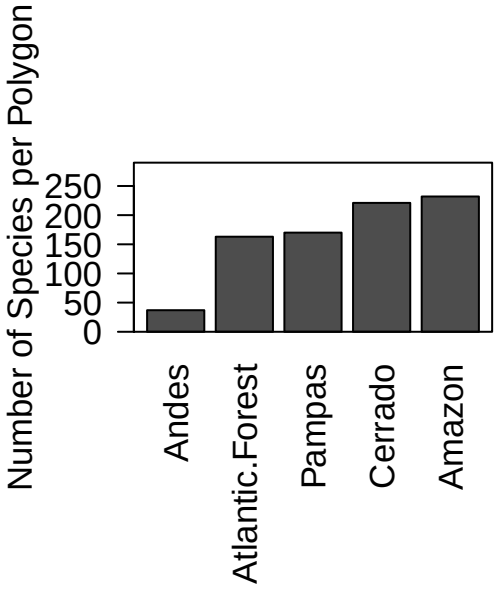
```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name", occ.thresh = 10)
```

Including Elevation in the area classification

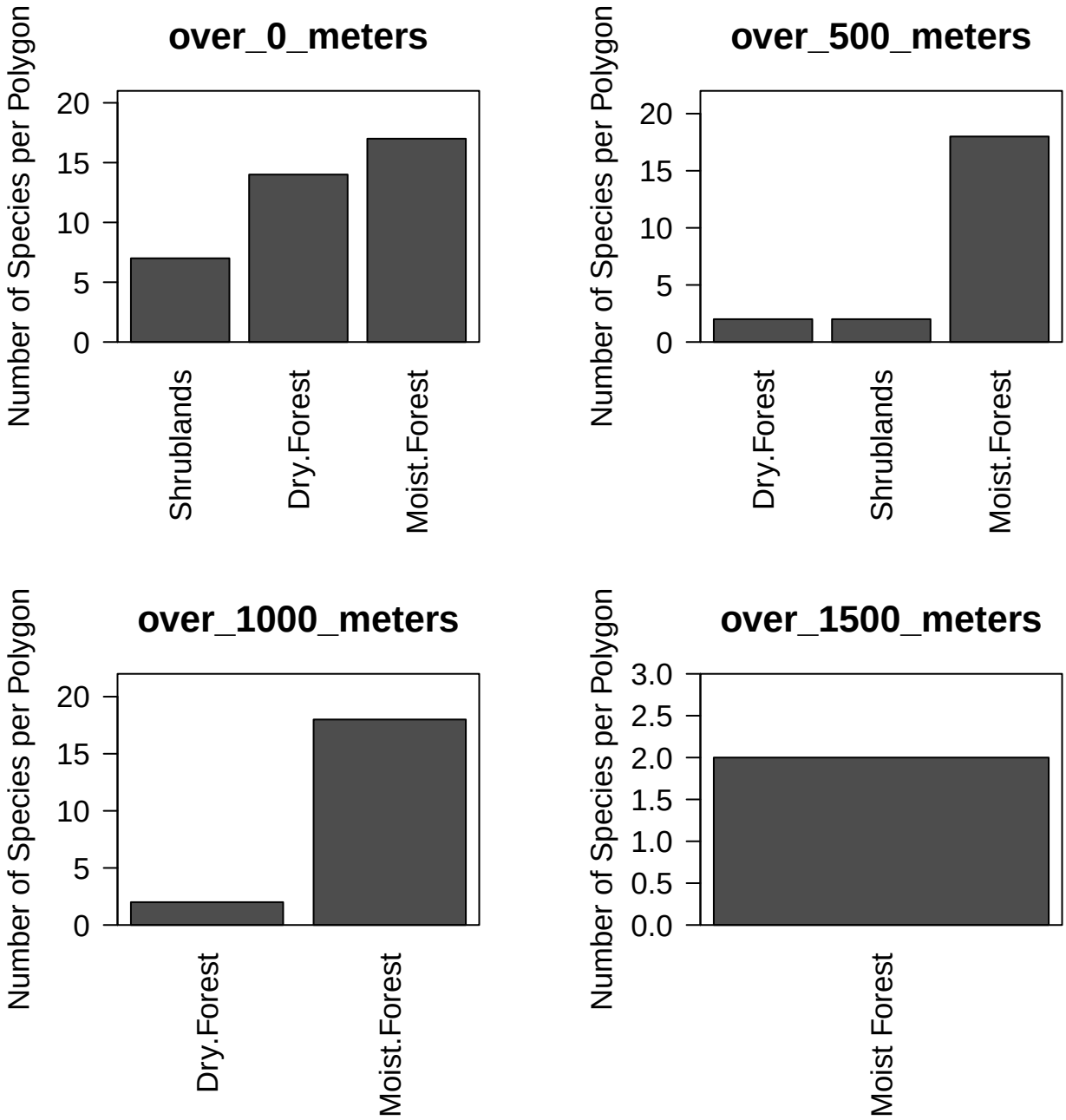
It is often desirable to include elevation as a proxy for environmental conditions into the area classification (i.e. consider highland and lowland species differently). You can do this by setting the `elevation` argument of `SpeciesGeoCoder` or `SpGeoCod` to TRUE and providing one or several elevation thresholds with the `threshold` argument. The output will then be a list of spgeoOUT objects according to the elevation thresholds. You can write the results to the working directory directly from the list using `WriteOut` or explore the objects further in R. If you are for example interested in highland and lowland rain forest species you can include a 500m, 1000m and 2000m threshold on the example dataset (strictly speaking it will divide the occurrences, but the result will be the same). **This will download large elevation data files from (http://srtm.csi.cgiar.org) to the working directory and might take long time for large datasets.**

```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name",
                 elevation = T, threshold = c(500, 1000, 1500))

#write to working directory
WriteOut(outp)

#plot species numbers at all elevation intervals
par(mar = c(8,4,4,4), mfrow = c(2,2))
```






```
for(i in 1:length(outp)){
  plot(outp[[i]], plottype = "speciesrichness", main = names(outp)[[i]])
  title(names(outp)[[i]])
}
```



WWF ecoregions

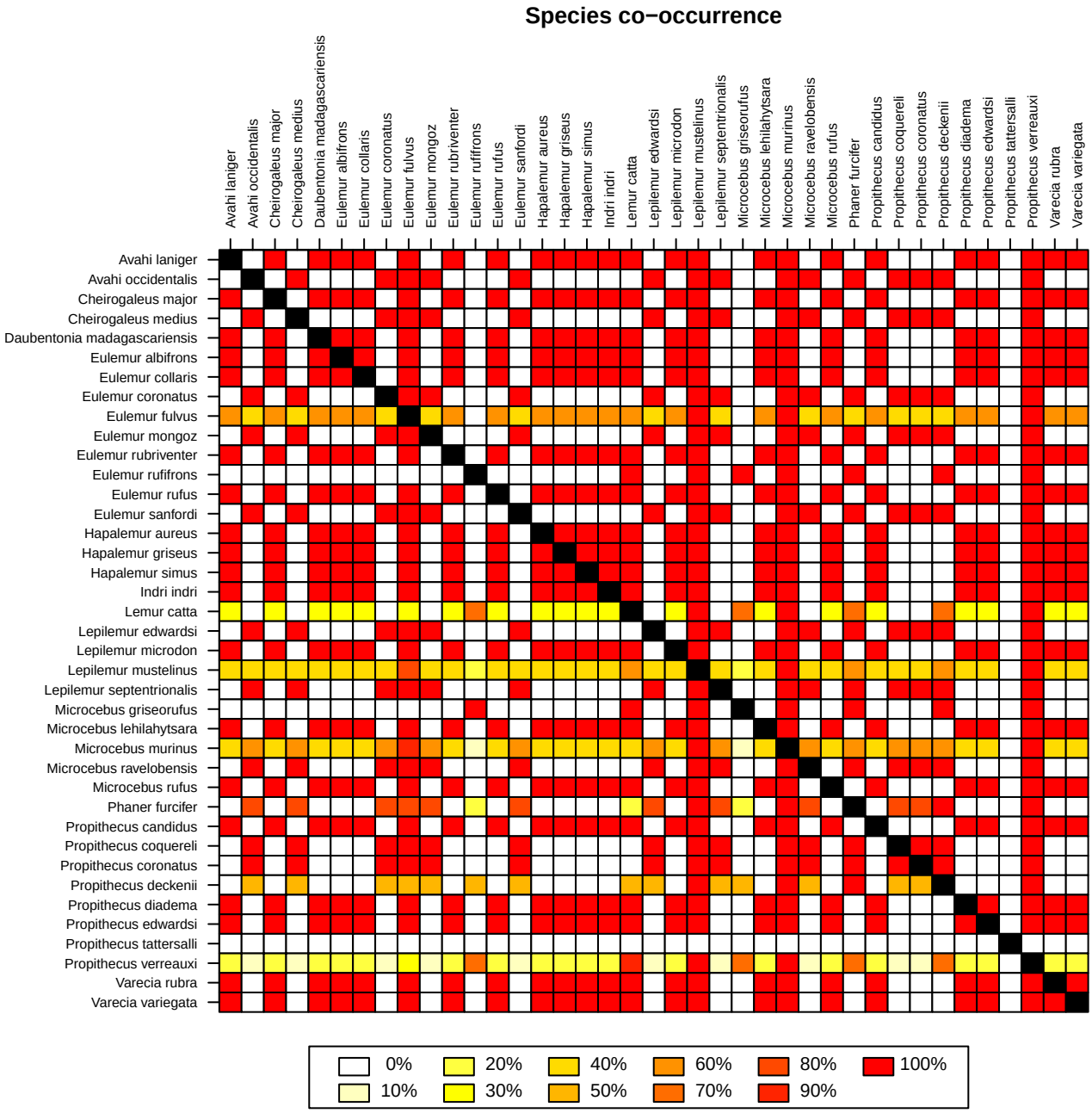
Biomes, as potentially biologically meaningful units, are of special interest in biogeography. You can directly use the WWF ecoregions (`areanames = "ECO_NAME"`) and biomes (`areanames = "BIOMES"`) in a speciesgeocodeR analyses with `WwfLoad` (Olson et al. 2001, WWF (2014)).

```
wd <- getwd()
wwf <- WwfLoad(wd)
outp <- SpGeoCod(lemurs, wwf, areanames = "ECO_NAME")
```

Coexistence matrix

You can use `CoExClass` to Calculate a coexistence matrix of species given the areas of interest. The rows indicate which percentage of a species occurrences fall in the same area as the species named in the column.

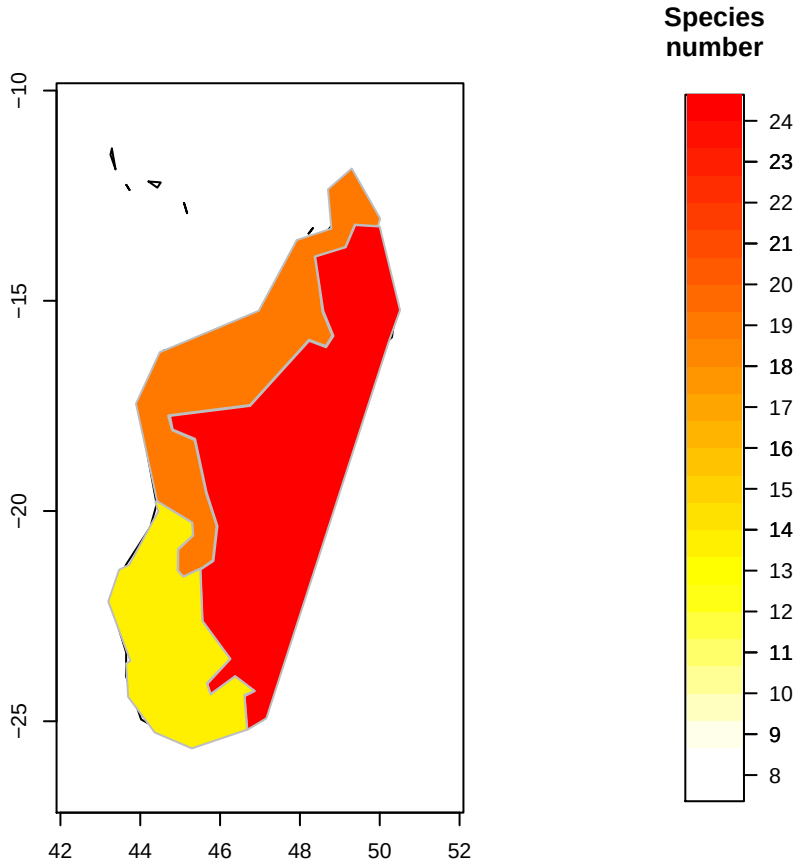
```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name")

coex <- CoExClass(outp)
plot(coex, plottype = "coexistence")
```



Mapping species richness in Polygons

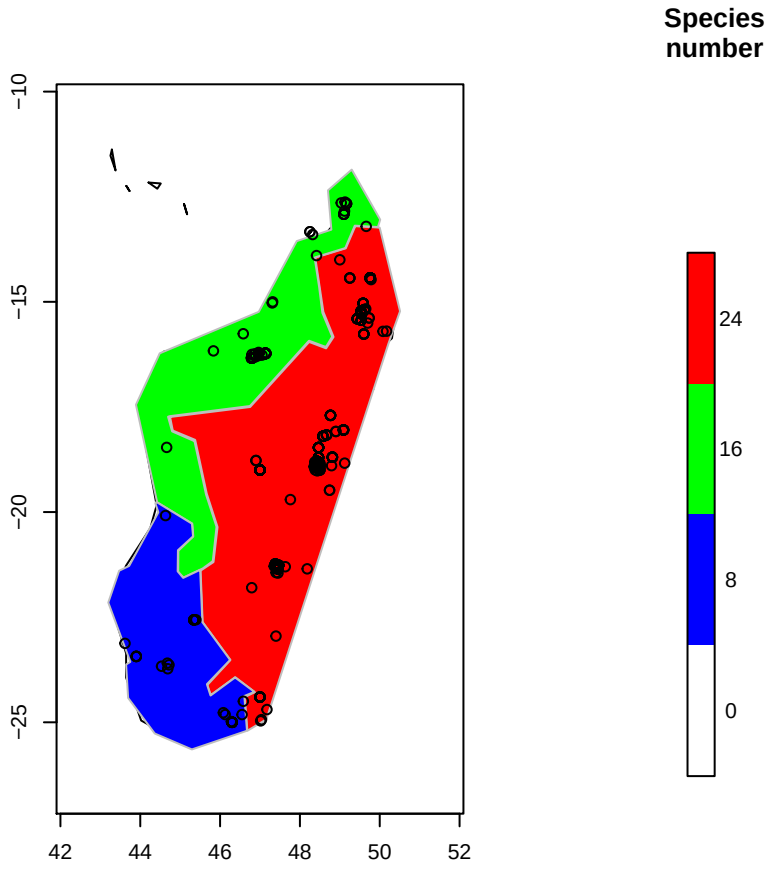
Use `MapRichness` to map species numbers in each polygon, using a continuous or discrete colour scheme.

```
outp <- SpGeoCod(lemurs, mdg_biomes, areanames = "name")

MapRichness(outp, leg = "continuous", xlim = c(42,52), ylim = c(-27, -10))
```




```
## class       : SpatialPolygonsDataFrame
## features    : 3
## extent      : 43.19906, 50.50903, -25.64502, -11.85583  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## variables   : 8
## names       : id,       name,       name, id, ident.add, sp.count,      code, ord
## min values  :  1, Dry Forest, Dry Forest,  1,         1,        8, #FF0000FF,   1
## max values  :  3, Shrublands, Shrublands,  3,         3,       24, #FFF000FF,   3
```

```
MapRichness(outp, leg = "discrete", show.occ = T, xlim = c(42,52), ylim = c(-27, -10))
```



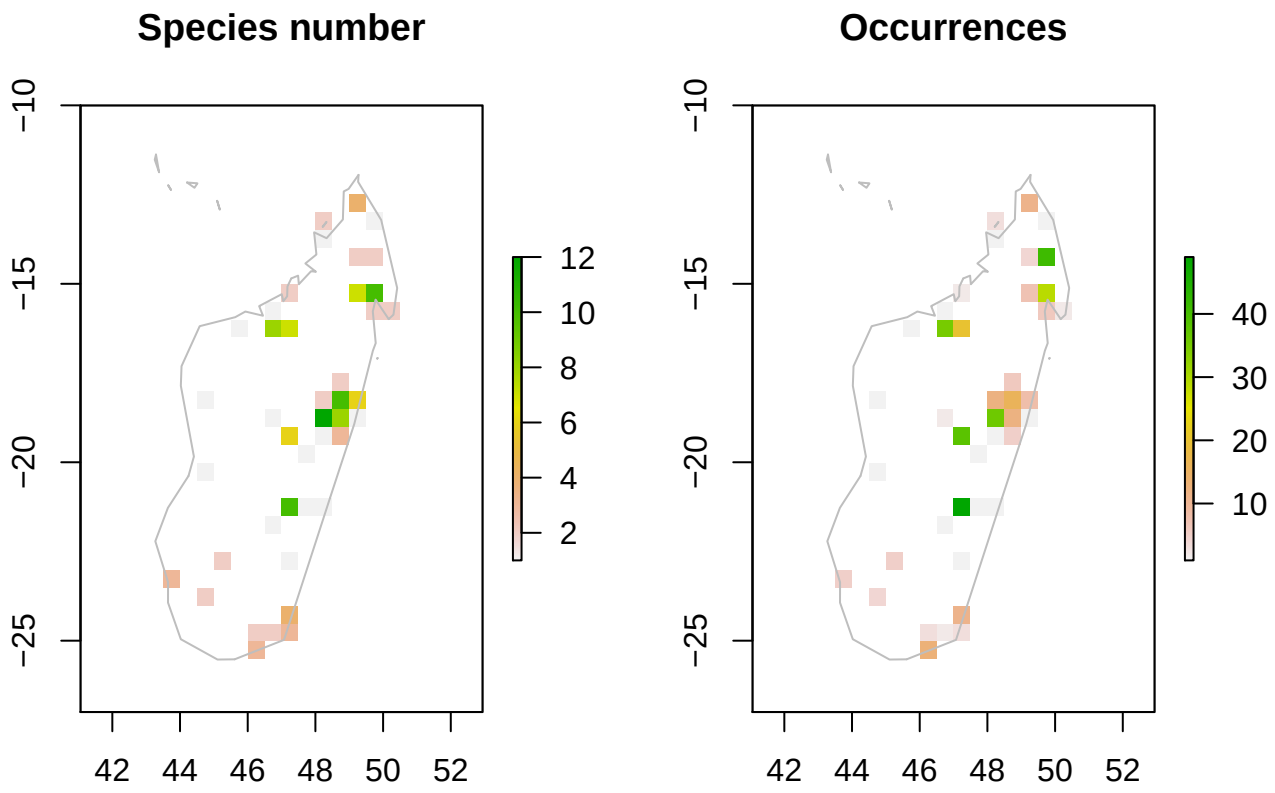

```
## class       : SpatialPolygonsDataFrame
## features    : 3
## extent      : 43.19906, 50.50903, -25.64502, -11.85583  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## variables   : 8
## names       : id,       name,       name, id, ident.add, sp.count,      code, ord
## min values  :  1, Dry Forest, Dry Forest,  1,         1,        8, #0000FFFF,   1
## max values  :  3, Shrublands, Shrublands,  3,         3,       24, #FF0000FF,   3
```

Richness grids

Use `RichnessGrid` to produce a raster with species number per grid cell or the number of occurrences per grid cell, and `MapGrid` to visualize the grid in the geographic context.

```
data(lemurs)

limits <- c(42, 52, -27, -10)
lemurs.spnum <- RichnessGrid(lemurs, limits, reso = 30, type = "spnum") #species number

lemurs.abu <- RichnessGrid(lemurs, limits,reso = 30, type = "abu") #occurrence number
par(mfrow = c(1,2))
MapGrid(lemurs.spnum)
```

```
title("Species number")

MapGrid(lemurs.abu)
title("Occurrences")
```



Range size/Extent of occurrence

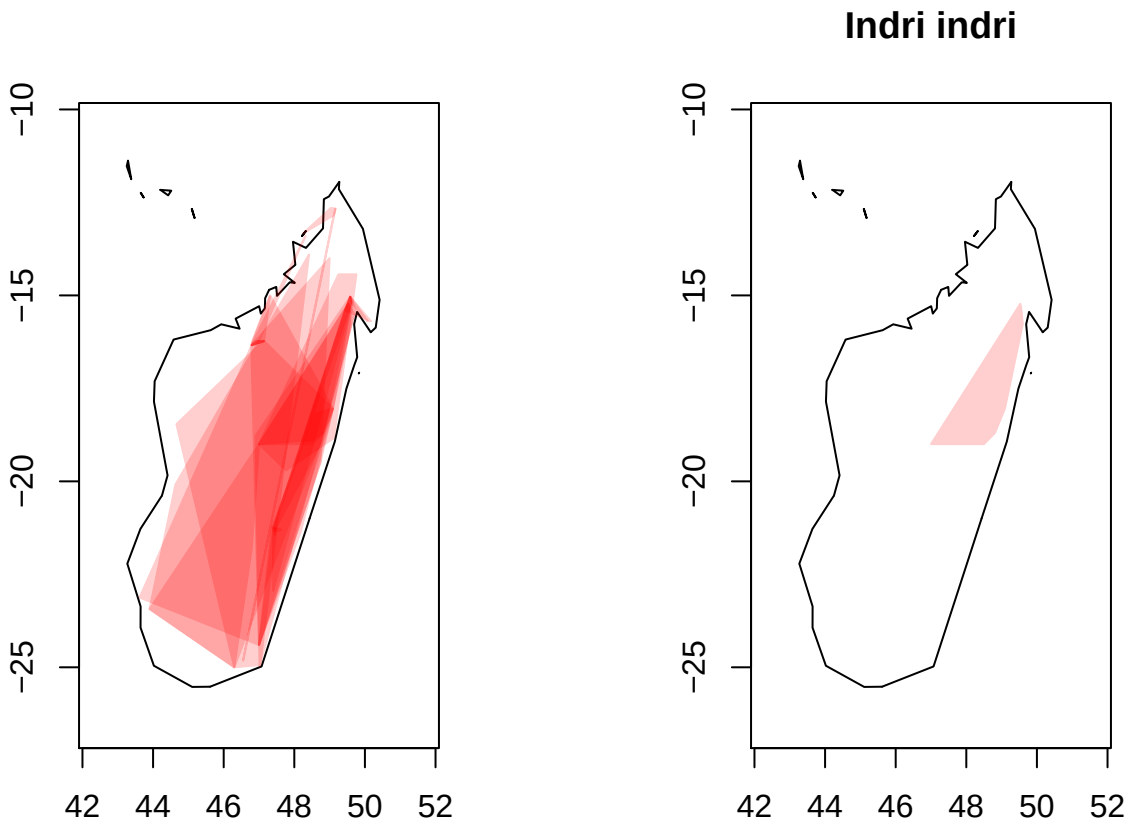
You can calculate species ranges and range size using the `CalcRange` function. This uses the occurrence points to calculate a convex hull polygon representing an estimate of the species ranges given the data. The `value` argument controls the output value, which can either be a data.frame of ranges sizes in square kilometres or a set of polygons showing the ranges.

```
data(lemurs)
data(mdg_poly)

inp <- ReadPoints(lemurs, mdg_poly)

are <- CalcRange(data.frame(inp$identifier, inp$species_coordinates),
                 value = "area")
```

```
head(are)
```

```
##                        EOO
## Avahi laniger        20777
## Avahi occidentalis     124
## Cheirogaleus major   48435
## Cheirogaleus medius    121
## Eulemur albifrons     1868
## Eulemur coronatus     1405
```

```
shp <- CalcRange(data.frame(inp$identifier,
                      inp$species_coordinates), value = "shape")
par(mfrow = c(1,2))
PlotHull(shp, select = "all", xlim = c(42, 52), ylim = c(-27, -10))
PlotHull(shp, select = "Indri indri", xlim = c(42, 52), ylim = c(-27, -10))
```

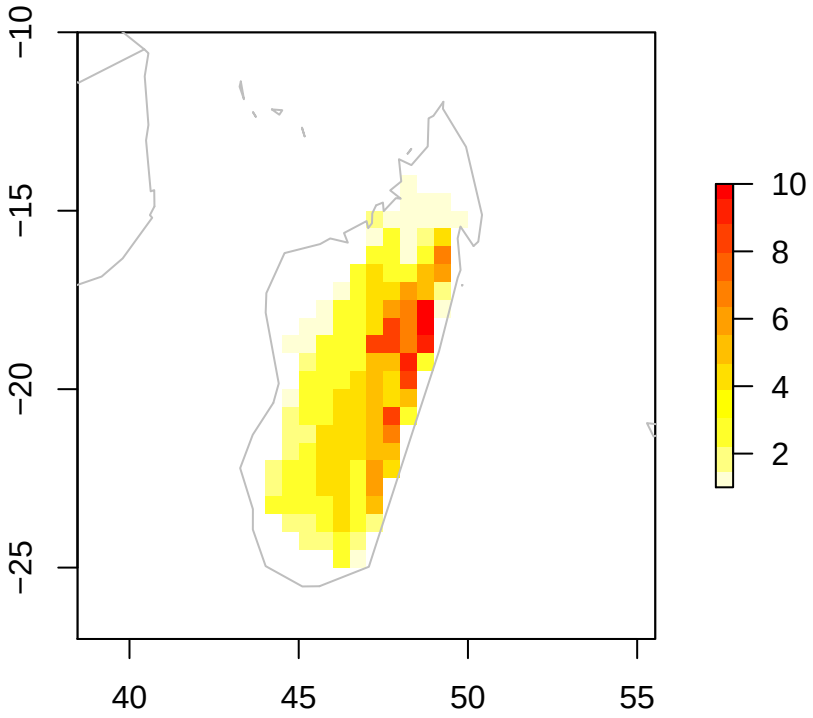


Species richness from ranges

Richness estimates based on range polygons instead of point occurrences can give more adequate estimate of species richness, if occurrence data are scarce. The RangeRichness function can be used to calculate species

richness based on range polygons. Here, we will use the polygons calculated from the point occurrences of the previous example.

```
bb <- RangeRichness(shp, limits = c(42, 52, -27, -10), reso = 30)
MapGrid(bb, col = rev(heat.colors(12)))
```



Becker, Richard A, Allan R Wilks, Ray Brownrigg, and Thomas P Minka. 2013. *maps: Draw Geographical Maps*. http://cran.r-project.org/package=maps.

Bivand, Roger S, Edzer Pebesma, and Virgilio Gomez-Rubio. 2013. *Applied spatial data analysis with R, Second edition*. Springer.

Bivand, Roger, and Nicholas Lewin-Koh. 2013. *maptools: Tools for reading and handling spatial objects*. http://cran.r-project.org/package=maptools.

Bivand, Roger, and Colin Rundel. 2014. *rgeos: Interface to Geometry Engine - Open Source (GEOS)*. http://cran.r-project.org/package=rgeos.

Crawley, M J. 2012. *The R book*. John Wiley; Sons.

Hijmans, Robert J. 2014. *Raster: Geographic data analysis and modeling*. http://cran.r-project.org/package=raster.

Olson, D M, E Dinerstein, E D Wikramanayake, N.l D Burgess, G V N Powell, E C Underwood, J A D'amico, et al. 2001. "Terrestrial Ecoregions of the World: A New Map of Life on Earth." *BioScience* 51 (11): 933–38.

Pebesma, E J, and R S Bivand. 2005. *Classes and methods for spatial data in R.* http://cran.r-project.org/doc/Rnews/.

Töpel, M, M F Calio, A Zizka, R Scharn, D Silvestro, and A Antonelli. 2014. "SpeciesGeoCoder: Fast categorisation of species occurrences for analyses of biodiversity, biogeography, ecology and evolution." *BioRxiv*.

WWF. 2014. "WWF terrestrial ecoregions:" worldwildlife.org/publications/terrestrial-ecoregions-of-the-world.