# Package 'adehabitatMA'

January 13, 2020

**Version** 0.3.14

**Date** 2020-01-11

**Depends** R (>= 2.10.0), sp, methods

**Suggests** maptools, tkrplot, MASS

**Imports** utils, graphics, grDevices, stats, filehash

**Title** Tools to Deal with Raster Maps

**Author** Clement Calenge, contributions from Mathieu Basille

**Maintainer** Clement Calenge <clement.calenge@ofb.gouv.fr>

**Description** A collection of tools to deal with raster maps.

**License** GPL (>= 2)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-01-13 19:20:08 UTC

## R topics documented:

---

`adeoptions` *Setting options for the adehabitat* package*

---

## Description

The function `adeoptions` defines the options for the package `adehabitatMA`, and more generally, for all brother packages (`adehabitatHR`, `adehabitatHS` and `adehabitatLT`).

## Usage

```
adeoptions(...)
```

## Arguments

| | |
|---|---|
| `...` | a list of named elements giving the value of options for the package `adehabitatMA` |

## Details

This functions sets the value of components of the internal list `.adeoptions`, which contains the values of the options. Two options are currently implemented:

`epsilon` the smallest significant distance between two points above which we should consider to deal with two separate locations

`shortprint` logical. Defines whether special `show` methods should be used when printing objects belonging to classes of the package `sp`

## Value

a list of options for the package `adehabitatMA`

## Author(s)

Clement Calenge <`clement.calenge@ofb.gouv.fr`>

## Examples

```
## load the data
data(lynxjura)

## short printing method
oldopt <- adeoptions(shortprint=TRUE)

lynxjura$map
```

```
## 'sp' print method for Spatial objects
adeoptions(shortprint=FALSE)

lynxjura$map

## original options
adeoptions(oldopt)

lynxjura$map
```

---

ascgen *Making Raster Maps From SpatialPoints Objects*

---

## Description

ascgen creates an object of class SpatialPixelsDataFrame using a set of points contained in an object of class SpatialPointsDataFrame.

## Usage

```
ascgen(xy, cellsize = NULL, nrcol = NULL, count = TRUE)
```

## Arguments

| | |
|---|---|
| xy | an object inheriting the class SpatialPoints |
| cellsize | the cell size (resolution) of the object of class SpatialPixelsDataFrame to be built |
| nrcol | the size of the square raster map to be created (number of rows/columns) |
| count | logical. If TRUE, the resulting object contains the number of points in each cell. If FALSE, all the cells are set to zero |

## Value

Returns an object of class SpatialPixelsDataFrame.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additional information on objects of class SpatialPixelsDataFrame and SpatialPoints-class for additional information on objects of class SpatialPoints.

## Examples

```
## generates a random sample of points
xy <- matrix(runif(1000), ncol=2)

## coerce them to SpatialPoints
xy <- SpatialPoints(xy)
plot(xy)

## generate a SpatialPixelsDataFrame
## (and count the number of points)
spd <- ascgen(xy, cellsize=0.1)
image(spd)
```

---

buffer *Compute Buffers Regions*

---

## Description

buffer computes buffers regions from spatial objects belonging to the following classes:

SpatialPoints (all pixels located within a specified distance of a point take the value one)

SpatialPointsDataFrame with one column (this column is considered to be a factor, and the buffer is computed for each level of this factor)

SpatialLines to compute buffers from lines.

SpatialPolygons to compute buffers from polygons.

## Usage

```
buffer(xy, x, dist)
```

## Arguments

| | |
|---|---|
| xy | an object of class SpatialPoints, SpatialPointsDataFrame with one column (a factor considered to be the identity of the points), SpatialLines, or SpatialPolygons. |
| x | an object inheriting the class SpatialPixels with the same attributes (resolution, size) as those desired for the output. |
| dist | a value of distance |

## Value

An object of class SpatialPixelsDataFrame.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(lynxjura)

# locs is the SpatialPointsDataFrame containing the
# locations of lynx indices in the Jura mountains
locs <- lynxjura$locs
head(locs)

## just for the sake of illustration: sample 100 points
suppressWarnings(RNGversion("3.5.0"))
set.seed(233)
locs <- locs[sample(1:nrow(locs), 100),]

# sa is the SpatialPixelsDataFrame object containing
# maps of the study area
sa <- lynxjura$map

# Buffer of 2000 m from all points
bu <- buffer(locs, sa, 2000)
image(bu)

# displays all the pixels of the study area within 2000 m
# of a point, for each type of indices (see ?lynxjura)
buani <- buffer(locs[,2], sa, 2000)
buani
par(mar=c(0,0,2,0))
opar<-par(mfrow=c(3,4))
lapply(1:11, function(i) {
   image(buani[,i])
   title(main = names(slot(buani, "data"))[i])
})
par(opar)

## buffer from a polygon
sa2 <- sa[,4]
sa2[[1]][sa2[[1]]<5000] <- NA
image(sa2)

## gets the contour line
gc <- getcontour(sa2)
plot(gc, add=TRUE)

## a buffer of 2000 metres
```

```
image(buffer(gc, sa, 2000))
plot(gc, add=TRUE)
```

---

calcperimeter *Compute Perimeters of Objects of Class "SpatialPolygonsDataFrame" and "PolyLinesDataFrame"*

---

## Description

calcperimeter computes the perimeters of polygons in objects of class SpatialPolygonsDataFrame and PolyLinesDataFrame.

## Usage

```
calcperimeter(x)
```

## Arguments

| | |
|---|---|
| x | object of class SpatialPolygonsDataFrame or PolyLinesDataFrame |

## Value

an object of class SpatialPolygonsDataFrame or PolyLinesDataFrame with an additionnal column containing the perimeter of the polygons/polylines.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPolygonsDataFrame-class for additionnal information on objects of class SpatialPolygonsDataFrame.

## Examples

```
data(meuse.grid)
a <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)


## the contour of the map
gc <- getcontour(a[,1])
plot(gc)

## Transforms the SpatialPolygons into SpatialPolygonsDataFrame
gc <- SpatialPolygonsDataFrame(gc, data.frame(x=1))
```

```
## The perimeter of this map (in units of the data):
ii <- calcperimeter(gc)
as.data.frame(ii)
```

---

count.points *Number of Points in Each Pixel of a Raster Map*

---

## Description

count.points counts the number of points in each pixel of a raster map inheriting the class SpatialPixels.

## Usage

```
count.points(xy, w)
```

## Arguments

| | |
|---|---|
| xy | an object of class SpatialPoints, or SpatialPointsDataFrame with one column. In the latter case, the column is considered as a factor giving, for each point, the membership of the point to a set. |
| w | an object inheriting the class SpatialPixels. |

## Value

an object of class SpatialPixelsDataFrame containing the number of points in each cell of the raster map. If xy is a SpatialPointsDataFrame with one column (a factor), the resulting object contains one column per level of this factor.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(lynxjura)

# locs is the SpatialPointsDataFrame containing the
# locations of presence indices of Lynx in the Jura mountains (France)
locs <- lynxjura$locs
```

```
head(locs)

# sa is the SpatialPixelsDataFrame object containing
# maps of the study area
sa <- lynxjura$map

# Count all points
cp <- count.points(locs, sa)
cp
image(cp)

# count the points per type of lynx presence indices:
cp <- count.points(locs[,2], sa)
cp
```

---

`distfacmap` *Compute distances to the different levels of a factor map*

---

## Description

This function computes maps of distances to patches belonging to the different levels of a factor variable (mapped in an object of class `SpatialPixelsDataFrame`).

## Usage

```
distfacmap(x, lev = NULL)
```

## Arguments

| | |
|---|---|
| `x` | an object of class `SpatialPixelsDataFrame` with one column (considered as a factor by the function) |
| `lev` | a vector of character strings giving the labels of the levels of the factor. |

## Value

An object of class `SpatialPixelsDataFrame`.

## Author(s)

Clement Calenge `<clement.calenge@ofb.gouv.fr>`

## See Also

`SpatialPixelsDataFrame-class` for additionnal information on objects of class `SpatialPixelsDataFrame`.

## Examples

```
## Load meuse.grid data set and converts it to
## SpatialPixelsDataFrame
data(meuse.grid)
m <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)

## look at the soil type
image(m[,6])

## compute the distance to each soil type
sor <- distfacmap(m[,6], lev = c("type1","type2", "type3"))

## The results
sor
mimage(sor)
```

---

`explore` *Interactive Exploration of Maps of Class 'SpatialPixelsDataFrame' (requires the package tkrplot)*

---

## Description

This interface allows to explore distances, values, etc. on a map of class `SpatialPixelsDataFrame`.

## Usage

```
explore(ka, coltxt="black",
        hscale = 1, vscale = 1,
        panel.last = NULL, ...)
```

## Arguments

| | |
|---|---|
| `ka` | An object of class `kasc` |
| `coltxt` | character. the color of the text to be printed |
| `hscale` | passed to tkrplot |
| `vscale` | passed to tkrplot |
| `panel.last` | an expression to be evaluated after plotting has taken place |
| `...` | additional parameters to be passed to the function `image` |

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.

## Examples

```
## Not run:

data(meuse.grid)
m <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)
explore(m[,3:5])

data(lynxjura)
explore(lynxjura$map, panel.last=function() points(lynxjura$locs))



## End(Not run)
```

---

# getcontour *Computes the Contour Polygon of a Raster Object*

---

## Description

getcontour computes the contour polygon of a raster object of class SpatialPixelsDataFrame. When the object is made of several parts, the function returns one polygon per part.

## Usage

```
getcontour(sp)
```

## Arguments

| | |
|---|---|
| sp | an object of class SpatialPixelsDataFrame |

## Value

Returns an object of class SpatialPolygons.

## Warning

Holes in the polygons are not taken into account by the function.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(meuse.grid)
a <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)


## the contour of the map
gc <- getcontour(a[,1])
plot(gc)
```

---

# hist.SpatialPixelsDataFrame

*Histograms of Mapped Variables*

---

## Description

hist.SpatialPixelsDataFrame performs histograms of the variables mapped in objects of class SpatialPixelsDataFrame.

## Usage

```
## S3 method for class 'SpatialPixelsDataFrame'
hist(x, type = c("h", "l", "b"),
                                      adjust = 1, col, border,
                                      lwd = 1, ...)
```

## Arguments

| | |
|---|---|
| x | a raster map of class SpatialPixelsDataFrame |
| type | what type of plot should be drawn. Possible types are:<br>* "h" for histograms,<br>* "l" for kernel density estimates (see ?density).<br>* "b" for both histograms and kernel density estimates (see ?density).<br>By default, type = "h" is used. If type = "l" is used, the position of the mean of each distribution is indicated by dotted lines |
| adjust | if type = "l", a parameter used to control the bandwidth of the density estimate (see ?density) |
| col | color for the histogram |
| border | color for the border of the histogram |

| | |
|---|---|
| lwd | if type = "l", line width for the density estimate |
| ... | further arguments passed to or from other methods |

## Author(s)

Mathieu Basille <basille@ase-research.org>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(lynxjura)
hist(lynxjura$map, type = "h")

hist(lynxjura$map, type = "l")
```

---

# join *Finds the Value of Mapped Variables at some Specified Locations (Spatial Join)*

---

## Description

join finds the value of a mapped variable at some specified locations.

## Usage

```
join(xy, x)
```

## Arguments

| | |
|---|---|
| x | an object of class SpatialPixelsDataFrame |
| xy | an object of class SpatialPointsDataFrame |

## Value

If only one variable is mapped in x, a vector with length equals to the number of points in xy.

If only several variables are mapped in x, a data.frame with a number of columns equal to the number of variables in the object of class SpatialPixelsDataFrame, and with each row corresponding to the rows of xy.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for additionnal information on objects of class SpatialPixelsDataFrame.
SpatialPoints-class for additionnal information on objects of class SpatialPoints.

## Examples

```
data(lynxjura)

## show the data
image(lynxjura$map)
points(lynxjura$locs)

## join the data to the maps:
res <- join(lynxjura$locs, lynxjura$map)
head(res)
```

---

kasc2spixdf *Conversion of old classes (adehabitat) to new classes (sp,adehabitatMA)*

---

## Description

These functions convert maps of classes available in adehabitat toward classes available in the package sp and conversely.

kasc2spixdf converts an object of class kasc into an object of class SpatialPixelsDataFrame.

asc2spixdf converts an object of class asc into an object of class SpatialGridDataFrame.

area2spol converts an object of class area into an object of class SpatialPolygons.

spol2area converts an object of class SpatialPolygons or SpatialPolygonsDataFrame into an object of class area.

attpol2area gets the data attribute of an object of class SpatialPolygonsDataFrame and stores it into a data frame.

## Usage

```
kasc2spixdf(ka)
asc2spixdf(a)
area2spol(ar)
spol2area(sr)
attpol2area(srdf)
```

### Arguments

| | |
|---|---|
| ka | an object of class kasc. |
| a | an object of class asc. |
| ar | an object of class area. |
| sr | an object of class SpatialPolygons or SpatialPolygonsDataFrame. |
| srdf | an object of class SpatialPolygonsDataFrame. |

### Details

We describe here more in detail the functions spol2area and attpol2area. Objects of class area do not deal with holes in the polygons, whereas the objects of class SpatialPolygons do. Therefore, when holes are present in the SpatialPolygons object passed as argument, the function spol2area ignore them and returns only the external contour of the polygon (though a warning is returned).

### Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

---

## labcon *Labelling Connected Features*

---

### Description

This function attributes unique labels to pixels belonging to connected features on a map of class SpatialPixelsDataFrame.

### Usage

```
labcon(x)
```

### Arguments

| | |
|---|---|
| x | an object of class SpatialPixelsDataFrame with one column |

### Value

Returns a matrix of class asc, of type "factor", with a number of levels equals to the number of connected components

### Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

### See Also

SpatialPixelsDataFrame-class for further information on the class SpatialPixelsDataFrame

## Examples

```
data(lynxjura)
sa <- lynxjura$map[,1]

## build an image with separate components
sa[[1]][sa[[1]]<50] <- NA

## show the connected components
image(sa)
image(labcon(sa))
```

---

lowres *Reducing the Resolution of a Map*

---

## Description

lowres is used to reduce the resolution of maps of class SpatialPixelsDataFrame.

## Usage

```
lowres(x, np = 2, which.fac=NULL, ...)
```

## Arguments

| | |
|---|---|
| x | an object of class SpatialPixelsDataFrame |
| np | a number giving the number of pixels to merge together (see below) |
| which.fac | a vector containing the indices of the columns of x, which should be considered as a factor |
| ... | further arguments passed to or from other methods |

## Details

The function merges together squares of np * np pixels. For variables of type "numeric", the function averages the value of the variable. For maps of type "factor", the function gives the most frequent level in the square of np * np pixels. When several levels are equally represented in the square of np * np pixels, the function randomly samples one of these levels.

## Value

Returns an object of class SpatialPixelsDataFrame.

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for further information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(meuse.grid)
m <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)
m

m <- m[,3:6]

## The initial image
image(m,3)

## The transformed image
m2 <- lowres(m, np = 4)
image(m2, 3)
```

---

# lynxjura *Monitoring of Lynx*

---

## Description

This data set stores the results of the monitoring of lynx in the French Jura between 1980 and 1999. These data have been collected by the Lynx Network of the french wildlife management office (Office national de la chasse et de la faune sauvage).

## Usage

```
data(lynxjura)
```

## Format

The list lynxjura has two components: map is an object of class SpatialPixelsDataFrame (see help(SpatialPixelsDataFrame)) that describes several variables on the study area: forets is the density of forests, hydro is the density of rivers, routes is the density of roads and artif is the distance from urbanized areas.

locs is an object of class SpatialPointsDataFrame containing the locations of presence indices of the lynx. X and Y are the x and y coordinates, Date is the date of the collection of the indice and Type represents the type of data (C: alive lynx captured, D: attacks on livestock, E: prints or tracks, F: feces, J: hairs, L: corpse of lynx, O: sightings and P: attacks on wild prey).

## Source

Vandel, J.M. (2001) *Repartition du Lynx (Lynx lynx) en France (Massif Alpin, Jurassien et Vosgien). Methodologie d'etude et statut actuel*. Ecole Pratique des Haute Etudes de Montpellier II: Dissertation.

---

`mimage` *Displaying Multi-layer Raster Maps*

---

## Description

This function allows to display the whole content of an object of class `SpatialPixelsDataFrame`

## Usage

```
mimage(x, var = names(slot(x, "data")), col = gray((240:1)/256), mfrow = NULL)
```

## Arguments

| | |
|---|---|
| `x` | an object of class `SpatialPixelsDataFrame` |
| `var` | The names or index of the variables to be plotted |
| `col` | a vector of colors to be used for plotting |
| `mfrow` | The parameter `mfrow` for the resulting graph (see `help(par)` |

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

`par` for information about `mfrow`, and `SpatialPixelsDataFrame-class` for additionnal information on objects of class `SpatialPixelsDataFrame`.

## Examples

```
data(lynxjura)

lynxjura$map

mimage(lynxjura$map)

mimage(lynxjura$map, c("forets", "routes"),
       col=grey(seq(0,1, length=100)))
```

morphology *Morphology: Erosion or Dilatation of Features on a Raster Map*

## Description

morphology performs morphological operations on images of class SpatialPixelsDataFrame.

## Usage

```
morphology(x, operation = c("erode", "dilate"), nt = 5)
```

## Arguments

| | |
|---|---|
| x | a matrix of class SpatialPixelsDataFrame with one column |
| operation | a character string indicating the operation to be processed: either "erode" or "dilate" |
| nt | the number of times that the operation should be processed |

## Value

Returns an object of class SpatialPixelsDataFrame with one column, containing 1 when the pixel belong to one feature of the image and NA otherwise (see examples).

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>

## See Also

SpatialPixelsDataFrame-class for further information on objects of class SpatialPixelsDataFrame.

## Examples

```
data(meuse.grid)
a <- SpatialPixelsDataFrame(points = meuse.grid[c("x", "y")],
                            data = meuse.grid)


## dilatation
toto1 <- morphology(a, operation = "dilate", nt = 1)
toto2 <- morphology(a, operation = "dilate", nt = 2)
toto3 <- morphology(a, operation = "dilate", nt = 3)
toto5 <- morphology(a, operation = "dilate", nt = 5)
colo <- grey((1:5)/6)
image(toto5, col = colo[1])
image(toto3, add = TRUE, col = colo[2])
image(toto2, add = TRUE, col = colo[3])
```

```
image(toto1, add = TRUE, col = colo[4])
image(a, add = TRUE)

## erosion
colo <- grey((1:12)/13)
image(a, col = "black")
for (i in 1:12) {
  toto <- morphology(a, operation = "erode", nt = i)
  image(toto, add = TRUE, col = colo[i])
}
```

---

puechabonsp *Radio-Tracking Data of Wild Boar*

---

## Description

This data set stores the results of the monitoring of 4 wild boars at Puechabon (Mediterranean habitat, South of France). These data have been collected by Daniel Maillard (Office national de la chasse et de la faune sauvage).

## Usage

```
data(puechabonsp)
```

## Details

The list puechabonsp has two components:

puechabonsp$map is an object of class SpatialPixelsDataFrame that describes several variables on the study area.

puechabonsp$relocs is an object of class SpatialPointsDataFrame containing the relocations of the wild boar resting sites in summer. Information on wild boars is provided by factors Name, Sex, Age.

## References

Maillard, D. (1996). *Occupation et utilisation de la garrigue et du vignoble mediterraneens par le Sanglier*. Universite d'Aix-Marseille III: PhD thesis.

## show-methods *Print Functions for Objects of the Package 'sp' Used in 'adehabitatMA'*

### Description

These are functions allowing a shorter print of the content of objects of class SpatialPolygonsDataFrame , SpatialPixelsDataFrame, SpatialPixels, and SpatialGridDataFrame. Original printing methods can be recovered by setting adeoptions(shortprint=FALSE).

### Methods

**object = "SpatialGridDataFrame"** show function for the class SpatialGridDataFrame.

**object = "SpatialPixels"** show function for the class SpatialPixels.

**object = "SpatialPixelsDataFrame"** show function for the class SpatialPixelsDataFrame.

**object = "SpatialPolygonsDataFrame"** show function for the class SpatialPolygonsDataFrame.

## subsetmap *Storing a Part of a Map*

### Description

subsetmap is used to store a part of any given map of class SpatialPixelsDataFrame into an other object.

### Usage

```
subsetmap(x, xlim = NULL, ylim = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | an object of class SpatialPixelsDataFrame |
| xlim | numerical vector of length 2. The x limits of the rectangle including the new map |
| ylim | numerical vector of length 2. The y limits of the rectangle including the new map |
| ... | further arguments passed to or from other methods |

### Details

If xlim or ylim are not provided, the function asks the user to click on the map to delimit the lower left corner and the higher right corner of the new map (see Examples).

## Value

Returns an object of class SpatialPixelsDataFrame

## Author(s)

Clement Calenge <clement.calenge@ofb.gouv.fr>, improvements by Jon Olav Vik

## Examples

```
data(lynxjura)
map <- lynxjura$map

## limits of the new map:
xl <- c(839938.7, 858990.8)
yl <- c(2149019, 2168761)

## computation of the new map:
su <- subsetmap(map, xlim = xl, ylim = yl)
su

## Display
opar <- par(mar = c(0,0,0,0))
layout(matrix(c(1,1,1,1,1,1,1,1,2), byrow = TRUE, ncol = 3))
image(map, axes = FALSE)
polygon(c(xl[1], xl[2], xl[2], xl[1]),
        c(yl[1], yl[1], yl[2], yl[2]))
image(su, axes = FALSE)
box()

par(opar)
par(mfrow = c(1,1))


## Not run:
 ## Interactive example
 su <- subsetmap(map)

 image(su)

## End(Not run)
```

# Index