Package 'RArcInfo'

July 2, 2014

Version 0.4-12

Date 2011-11-06

Encoding latin1

Title Functions to import data from Arc/Info V7.x binary coverages

Author Virgilio Gómez-Rubio <virgilio.gomez@uv.es>

Maintainer Virgilio Gómez-Rubio <virgilio.gomez@uv.es>

Depends R (>= 2.3.0), RColorBrewer

Description This package uses the functions written by Daniel Morissette <danmo@videotron.ca> to read geographical information in Arc/Info V 7.x format and E00 files to import the coverages into R variables.

License GPL (>= 2)

URL http://sourceforge.net/projects/rarcinfo/,http://avce00.maptools.org/docs/v7_bin_cover.html

Repository CRAN

Date/Publication 2011-11-07 14:05:13

NeedsCompilation yes

R topics documented:

avctoe00 2
e00toavc 3
get.arcdata 3
get.bnddata 4
get.cntdata 5
get.labdata 5
get.namesofcoverages 6
get.nb 7

get.paldata 7
get.tabledata 8
get.tablefields 9
get.tablenames 9
get.toldata 10
get.txtdata 11
plotarc 11
plotpal 12
plotpoly 13
RArcInfo 15
read.coverage 17
thinlines 18

Index **19**

`avctoe00` *Converts a ARC/INFO binary coverage into an ESRI E00 file*

Description

This function makes a convertion to an ESRI E00 file from a binary coverage.

Usage

```
avctoe00(avcdir, e00file)
```

Arguments

`avcdir`	The path to the binary coverage we want to convert from.
`e00file`	The E00 file to be created.

Value

Returns 'NULL' on exit.

See Also

e00toavc

e00toavc — *Converts an ESRI E00 file into an Arc/Info V 7.x binary coverage*

Description

This function makes a convertion from an ESRI E00 file to a binary coverage. Usually two new directories are created. One with the name of the coverage and another called 'info', where some information about the tables are created. If this directory already exists (because there are already other binary coverages), then the new information is added and no file is replaced or deleted.

Usage

```
e00toavc(e00file, avcdir)
```

Arguments

e00file	The E00 file to be converted.
avcdir	The path to the binary coverage directory we want to create.

Value

Returns 'NULL' on exit.

See Also

avctoe00

get.arcdata — *Function for importing the contents of an ARC file into R*

Description

This function reads and imports into R the contents of an arcs definition file.

Usage

```
get.arcdata(datadir, coverage, filename="arc.adf")
```

Arguments

datadir	Directory under which all the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with.
filename	The name of the file in the coverage directory that stores the data. By default, it is called 'arc.dat'.

Value

This function returns a list with two elements. The first one is a data frame containing the next fields (by columns):

`ArcID`	A number that identifies this arc.
`ArcUserID`	Identifier defined by the user.
`FromNode`	The node where the arc begins.
`ToNode`	The node where the arc finishes.
`LeftPoly`	The number of the polygon that is to the left of the arc.
`RightPoly`	The number of the polygon that is to the right of the arc.
`NVertices`	The number of vertices the arc has.

The second element is a list that stores the vertices of the arc. So, each element in this list is also a list of two arrays: the first for the X coordinates and the secod for the Y coordinates.

`get.bnddata` *Function for importing the contents of a BND file into R*

Description

This function reads and imports into R the contents of a BND file. This kind of files store the bounds of the given coverage (the one we are working with). That is, the upper and lower bounds for the x and y coordinates.

Usage

```
get.bnddata(infodir, tablename)
```

Arguments

`infodir`	Directory where there is a file called arc.dat (usually, it is called 'info').
`tablename`	The name of the table in the coverage that stores the data (usually called 'COVERAGENAME.BND').

Value

A vector with the x min, y min, x max and y max values.

get.cntdata *Function for importing the contents of a CNT file into R*

Description

This function reads and imports into R the contents of a polygon centroid information file.

Usage

```
get.cntdata(datadir, coverage, filename="cnt.adf")
```

Arguments

datadir	Directory under which all the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with.
filename	The name of the file in the coverage directory that stores the data (usually called 'cnt.adf').

Value

This functions returns a list with two elements. The first one is a data frame with the next fields (columns):

PolygonID	The polygon itself.
CoordX	An array with the X coordinates of the centroid.
CoordY	An array with the Y coordinates of the centroid.
NLabels	The number of labels this polygon has.

The second element in the list is an array with the label identifiers related to this polygon.

get.labdata *Funtion for importing the contents of a LAB file into R*

Description

This function reads and imports into R the contents of a polygon labels definition file.

Usage

```
get.labdata(datadir, coverage, filename="lab.adf")
```

Arguments

datadir	Directory under which iall the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with
filename	The name of the file in the coverage directory that stores the data. By default, called 'lab.adf'.

Value

This function returns a data frame with the next fields (by columns):

LabelUserID	The label of the polygon.
PolygonID	The polygon related to the label.
Coord1X	The X component of the first coordinate.
Coord2X	The X component of the second coordinate.
Coord3X	The X component of the third coordinate.
Coord1Y	The Y component of the first coordinate.
Coord2Y	The Y component of the second coordinate.
Coord3Y	The Y component of the third coordinate.

---

get.namesofcoverages *Function for getting the names of the coverages*

Description

This function returns the name of all the coverages under the given directory.

Usage

```
get.namesofcoverages(directory)
```

Arguments

directory	A character string with the name of the directory which contains the coverages.

Value

A list of character strings with the names of the files and directories under the given one.

get.nb — *Function for calculating neighbouting polygons.*

Description

This function allows the user to calculate, for every given polygon (in the array 'index'), the list of its neighbouring polygons.

Usage

```
get.nb(arc,pal, index=NULL)
```

Arguments

arc	The list of arc definitions, as returned by 'get.arcdata'.
pal	The list of polygon definitions, as returned by 'get.paldata'.
index	An array with the polygons we want to use to calculate their neighbours. It must be an array. If 'index' is not set, then all the polygons are used.

Value

A list in which the first element is a vector of the neighbouring polygons of the fist element in 'index', and so on.

get.paldata — *Function for importing the contents of a PAL file into R*

Description

This function reads and imports into R the contents of a polygon definitions file.

Usage

```
get.paldata(datadir, coverage, filename="pal.adf")
```

Arguments

datadir	Directory under which all the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with
filename	The name of the file in the coverage directory that stores the data. By default, it is called 'pal.adf'

Value

This function returns a list with two elements. The first one is a data frame with the next fields (columns):

PolygonID	A number that identifies this polygon.
MinX	Minimum value for all the X component of the coordinates.
MinY	Minimum value for all the Y component of the coordinates.
MaxX	Maximum value for all the X component of the coordinates.
MaxY	Maximum value for all the Y component of the coordinates.
NArcs	?

The second element in the list is also a list in which each element is composed by three arrays with information about the polygons that are in the polygon boundary: 'Arc ID', 'From Node' and 'Adjacent Polygon'.

---

get.tabledata *Function for importing the contents of a table file into R*

Description

This function reads and imports into R the contents of a table file.

Usage

```
get.tabledata(infodir, tablename)
```

Arguments

infodir	Info directory where there is a file called arc.dat
tablename	The name of the table from which we want to import the data

Value

This function returns a data frame in which each column stores the data of a field from the table. The columns are not assigned the names of the fields, but this can be got with the get.tablefields function.

get.tablefields *Function for reading names of the table fields in the coverages*

Description

This function returns the names of the fields (and its type) in the table whose names are provided by the user.

Usage

```
get.tablefields(infodir,tablename)
```

Arguments

infodir	Info directory where there is a file called arc.dat
tablename	The name of the table from which we want to get the fields

Value

This function returns a data frame with the next fields (columns):

FieldName	The name of the field.
FieldType	This is an integer from 1 to 6 that explain the kind of data: • 1Date • 2Character String • 3Integer (stored as a character string) • 4Numeric (stored as a character string) • 5Binary integer • 6Binary float

get.tablenames *Function for reading the names of the tables in the coverages*

Description

This function reads the arc.dat file in the info directory and it returns a list with some data about the tables. Each element of this list is another list with data from a single table.

Usage

```
get.tablenames(infodir)
```

Arguments

infodir	info dir where there is a file called arc.dat

Value

A list with some information about the tables stored in all the coverages:

- Table Name
- Info File
- Number of fields
- Record Size
- Number of records
- Internal/External Table (FALSE/TRUE)

`get.toldata` *Function for importing the contents of a TOL file into R*

Description

This function reads and imports into R the contents of a tolerance definition file.

Usage

```
get.toldata(datadir, coverage, filename="tol.adf")
```

Arguments

`datadir`	Directory under which all the coverages and a directory called 'info' are.
`coverage`	The name of the coverage we want to work with
`filename`	The name of the file in the coverage directory that stores the data. By default it is called 'tol.adf'. In some cases, when the values are stored in double precission, the tolerances are in a file whose name is 'par.adf'.

Value

This function returns a data frame with the next fields (columns):

`Type`	A number from 1 to 10 showing the tolerance type.
`Status`	This field indicates whether the tolerance is active or not.
`Value`	Tolerance value.

get.txtdata *Function for importing the contents of an TXT file into R*

Description

This function reads and imports into R the contents of a file of annotations (TXT).

Usage

```
get.txtdata(datadir, coverage, filename="txt.adf")
```

Arguments

datadir	Directory under which all the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with.
filename	The name of the file in the coverage directory that stores the data. By default, it is called 'txt.dat'.

Value

This function returns a list with two elements. The first one is a dataframe with the next columns:

TxtID	This field identifies the anotation.
UserID	Identifier defined by the user.
Level	I don't know what this exactly means. Please, help me to fix this.
NVerticesLine	Number of vertices pairs that are valid.
NVerticesArrow	I don't know what this exactly means. Please, help me to fix this.
Text	Some text related to the annotation.

The second element in the list is another list containing the vertices related the annotation. For each annotation there are two vectors, for the X and Y coordinates.

plotarc *Plots the data imported from an ARC file*

Description

Taking as argument the list returned by the get.arcdata function, this function plots all the arcs. With the argument new the user can decide whether to plot on a new window/device or on the last window/device.

If the user provides the arguments 'xlim' and 'ylim' they will be used when calling the 'plot' function. Other way, the real boundary of the plotted arcs will be used.

Usage

```
plotarc(arc, new=TRUE, index=NULL, xlim, ylim, ...)
```

Arguments

arc	The data returned by a call to get.arcdata
new	Do you want to plot on the last window/device or on a new one?
index	A vector containing the indexes of the arcs to be plotted. If it nos supplied all the arcs will be plotted.
xlim	x limits
ylim	y limits
...	Options to be passed to a call to the function plot when creating the display (i. e., window or file).

Value

This function returns nothing, but plots a nice map :-D.

---

plotpal *Plots the data imported from an ARC file according to the contents of a PAL file*

Description

This function works like plotarc, but we can also decide what polygons to plot. The arc definitions are stored in the arc variable, and the polygon definitions are in the pal variable.

Usage

```
plotpal(arc, pal, new=TRUE, index, ...)
```

Arguments

arc	The data returned by a call to get.arcdata
pal	The data returned by a call to get.paldata
new	Do you want to plot on the last window/device or on a new one?
index	The indices of the polygons to be plotted.
...	Options to be passed to a call to the function plot when creating the display (i. e., window or file).

Value

This function returns nothing, but plots a nice map :-D.

plotpoly *Plots polygons defined by the coverages.*

Description

This function is capable of plotting polygons, referenced by its id number. These can be filled with colors acording to a given variate.

Usage

```
plotpoly(arc,bnd,pal,index=NULL,col, border=NULL,xratio=1, yratio=1,...)
```

Arguments

arc	The data returned by a call to get.arcdata
bnd	The data returned by a call to get.bnddata
pal	The data returned by a call to get.paldata
index	IDs of the polygon to be plotted. If it is 'NULL' then all the polygons are plotted.
col	Colors to be used when filling the polygons
border	Colors used for the lines of the polygons. If it's not set, it is set to the value of 'col'.
xratio	Controls x-axis ratio. It can take any value from 0 to 1.
yratio	Controls x-axis ratio. It can take any value from 0 to 1.
...	Options to be passed to a call to the function plot when creating the display (i. e., window or file).

Value

Thins function returns nothing, but plots a nice map. :-D

Examples

```
#This example is the same as the one provided under the test directory

#1.- Create temporary directory (if needed)
#1.- Extract the E00 file form a ZIP file
#2.- Create an Arc/Info binary coverage
#3.- Create the map

#get current working directory
cwd<-getwd()
#Create tmp directory.
tmpdir<-tempdir()


datadir<-system.file("exampleData",package="RArcInfo")
```

```
setwd(datadir)
file.copy(c("valencia.zip", "data_valencia.csv"), tmpdir, overwrite = TRUE)

setwd(tmpdir)

#Convert E00 file to a binary covertage to be imported into R

#Comment this line if the file valencia.e00 already exists
unzip(zipfile="valencia.zip", file="valencia.e00")

#Comments this lines if the binary coverage already exists
library(RArcInfo)
e00toavc("valencia.e00", "valencia")


#Read map data
arcsmuni<-get.arcdata(".", "valencia")
palmuni<-get.paldata(".", "valencia")
bnd.muni<-get.bnddata("info/", "VALENCIA.BND")
patmuni<-get.tabledata("./info", "VALENCIA.PAT")

#Number of polygons
nmuni<-length(palmuni[[1]][[1]])

municipios<-data.frame(1:nmuni, patmuni$"VALENCIA-ID")
names(municipios)<-c("INDEX", "CODMUNICI")


#Datafiles to be used

unemp<-read.table(file="data_valencia.csv", sep=";",
dec = ",",skip=1)

unemp<-unemp[,c(1,3)]
names(unemp)<-c("CODMUNICI", "UNEMP")

breaks<-quantile(unemp[,2], c(0,  .025,.2, .8, .975, 1))

unemp<-cbind(unemp, CAT=as.ordered(cut(unemp[,2], breaks, include.lowest = TRUE) ))


library(RColorBrewer)

#Colors to be used in maps
#colors<-brewer.pal(5, "Oranges")
colors<-brewer.pal(5, "Greens")

ldata<-merge(unemp, municipios, by.x="CODMUNICI", by.y="CODMUNICI")


#Valencia
idx<-(ldata$"CODMUNICI">=46000)
bnd.muni<-c(626679.9, 4250000, 760000, 4460000)
```

```
p<-par()
pin<-1.5*p$pin
main<-"Rate of unemployment"

plotpoly(arc=arcsmuni, pal=palmuni, bnd=bnd.muni,
index=ldata$INDEX[idx], col=colors[ldata$CAT][idx],
xlab="", ylab="", main=main,
xaxt="n", yaxt="n", bty="n")

#Set legend
l<-levels(unemp$CAT)
l[1]<-"[0.00,1.26]"
legend(700000, 4460000, fill=colors,
legend=l, bty="n", cex=1)

setwd(cwd)
```

RArcInfo *RArcInfo*

Description

This package allows the user to import into R binary coverages in format Arc/Info V 7.x. These coverages represent geographical data in several forms: points, lines, polygons, point labels, etc.

RArcInfo uses the library AVCE00, written by Daniel Morissette, to whom I would thank fo his marvelous work. But RArcInfo is much more than a wrapper of this library, because it provides functions to plot the data and draw maps.

Since the geographical data are separated into several files, RArcInfo provides a different function to read each file. These functions are called get.XXXdata, where XXX is the file name to open (usually the extension is 'adf'):

get.arcdata ARC files contain arcs definition and their vertices.

get.bnddata BND files contain coordinates for the boundary of the data.

get.cntdata CNT files contain polygon centroid information.

get.labdata LAB files contain label point records.

get.paldata PAL files contain the polygon definitions.

get.toldata TOL files contain the tolerance values that were used when processing the polygon coverage.

get.txtdata TXT files contain annotations (or labels) about the data.

Besides these files, binary coverage store several tables containing additional information (like name of the city, population, etc.). To get this data, RArcInfo provides the next functions:

get.tablenames Gets all the table names and the coverage each table belongs to.

get.tablefields Gets the names of the fields for a given table.

get.tabledata Gets the data stored in the table.

In order to plot the data, RArcInfo has several functions:

plotarc Plots all the arcs.

plotpal Plots all the polygons.

plotpoly Like plotpal, but it allows to select the polygons we want to plot, colour and other stuff. This is useful to plot maps according the value of some covariate.

To get all the names of the coverages, the user can call 'get.namesofcoverages'. Other two interesting functions are:

thinlines Useful to reduce the number of points in an arc according to a given tolerance.

get.nb Calculates the neighbouring polygons of a given set of polygons.

References

V. Gómez-Rubio and A. López-Quílez (2005). RArcInfo: Using GIS data with R. Computers & Geosciences 3(8), 1000-1006.

More information about this kind of data can be found at http://avce00.maptools.org/docs/v7_bin_cover.html.

See Also

get.arcdata, get.bnddata, get.cntdata, get.labdata, get.paldata, get.toldata, get.txtdata, get.tablenames, get.tablefields, get.tabledata, get.namesofcoverages, read.coverage, thinlines, get.nb

Examples

```
#See example(plotpoly) for another example

library(RArcInfo)

datadir<-system.file("exampleData",package="RArcInfo")
infodir<-system.file("exampleData","info",package="RArcInfo")
coveragedir<-system.file("exampleData","wetlands",package="RArcInfo")

#get.bnddata needs the last slash...
infodir<-paste(c(infodir,"/"), collapse="")

#List all the tables

covnames<-get.namesofcoverages(datadir)
tablenames<-get.tablenames(infodir)


#Display the name of the table and its fields
for(i in 1:length(tablenames[[1]]))
```

```
{
print(c("Table: ",tablenames$TableName[i]))

fields<-get.tablefields(infodir,tablenames$TableName[i])
print("Fields")
for(j in 1:length(fields))
print(fields[[j]][1])

#Get the data
if(i==1)
tabledata<-get.tabledata(infodir,tablenames$TableName[i])
else
tabledata<-c(tabledata, get.tabledata(infodir,tablenames$TableName[i]) )
}

#Import data from some tables
arc<-get.arcdata(datadir,"wetlands")
pal<-get.paldata(datadir,"wetlands")
lab<-get.labdata(datadir,"wetlands")
cnt<-get.cntdata(datadir,"wetlands")

bnd<-get.bnddata(infodir,"WETLANDS.BND")

print("Plotting all the arcs")
plotarc(arc)

print("Plotting the first ten polygons (in red) on the previous plot")
par(col="red")
plotpal(arc,pal,new=FALSE, index=1:10)
```

`read.coverage` *Function for retrieving basic data from a given coverage*

Description

This function allows the user to retrieve the main data stored in the chosen coverage. This doesn't mean that all the data are retrieved, but only the most important.

Tables are not imported, but a dataframe containig the information provided by a call to get.tablenames of all the tables related to the coverage.

For a single coverage there can be more than one pal file, so just their names are imported (into 'palfiles'). Then, the user can choose which one to import by calling get.paldata. But, if there is just one pal file, it is imported into 'pal'.

The names of the different files are supposed to be the default names when calling the proper get.XXXdata. If some of these names change then the function will not import that feature properly. But this fact should never happen.

Usage

```
read.coverage(datadir, coverage)
```

Arguments

datadir	Directory under which all the coverages and a directory called 'info' are.
coverage	The name of the coverage we want to work with

Value

This function returns a list of all the data directory, coverage name and all the data imported: i

datadir	The directory where all the coverages are stored, the one used when this function was called.
coverage	The name of the coverage used when calling this function.
arc	The arc definitions, as returned by get.arcdata.
bnd	The bounday definition, as returned by get.bnddata.
cnt	The polygon centroids, as returned by get.cntdata.
lab	The label point records of the coverage.
pal	If there is just one file for the polygon definitions inside the coverage, it is imported here using get.paldata. Otherwise, it is filled with a 'NULL'.
palfiles	The names of all the polygon definitions files.
tblnames	The description of all the tables related to the coverage, as returned by get.tablenames.
tol	The tolerance values.

References

More information about this kind of data can be found at http://pages.infinit.net/danmo/e00/docs/v7_bin_cover.html.

---

thinlines *Function for deleting points in an arc definition object*

Description

Usually the plotting device resolution allows us to plot only a few points per arc with no difference.

Usage

```
thinlines(arc, tol)
```

Arguments

arc	The original arc definition object, as retuend by get.arcdata.
tol	The theshold we used to define which polygons are 'too close'.

Value

A new arc definition object, usually with less points per arc than the original. Notice that no modification is made to the topology but the number of vertices is updated in the table related to the arc definition.

Index

∗Topic **file**
avctoe00, 2
e00toavc, 3
get.arcdata, 3
get.bnddata, 4
get.cntdata, 5
get.labdata, 5
get.namesofcoverages, 6
get.nb, 7
get.paldata, 7
get.tabledata, 8
get.tablefields, 9
get.tablenames, 9
get.toldata, 10
get.txtdata, 11
RArcInfo, 15
read.coverage, 17
thinlines, 18
∗Topic **hplot**
plotarc, 11
plotpal, 12
plotpoly, 13

avctoe00, 2

e00toavc, 3

get.arcdata, 3
get.bnddata, 4
get.cntdata, 5
get.labdata, 5
get.namesofcoverages, 6
get.nb, 7
get.paldata, 7
get.tabledata, 8
get.tablefields, 9
get.tablenames, 9
get.toldata, 10
get.txtdata, 11

plotarc, 11
plotpal, 12
plotpoly, 13

RArcInfo, 15
read.coverage, 17

thinl *(*thinlines*)*, 18
thinlines, 18