Package ‘Grid2Polygons’

February 21, 2015

Version 0.1-5
Date 2015-02-20
Title Convert Spatial Grids to Polygons
Author Jason C. Fisher
Maintainer Jason C. Fisher <jfisher@usgs.gov>
Depends R (>= 2.15.0), sp, rgeos
Suggests rgdal
Description Converts a spatial object from class SpatialGridDataFrame to SpatialPolygonsDataFrame.
License GPL (>= 2)

URL https://github.com/jfisher-usgs/Grid2Polygons

BugReports https://github.com/jfisher-usgs/Grid2Polygons/issues
ByteCompile yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-02-21 00:04:22

R topics documented:

<table>
<thead>
<tr>
<th>Grid2Polygons</th>
<th>2</th>
</tr>
</thead>
</table>

Index 5
Grid2Polygons

Convert Spatial Grids to Polygons

Description

Converts sp spatial objects from class SpatialGridDataFrame to SpatialPolygonsDataFrame. Spatial polygons can then be transformed to a different projection or datum with spTransform in package rgdal. Image files created with spatial polygons are reduced in size and result in a much "cleaner" version of your image.

Usage

Grid2Polygons(grd, zcol = 1, level = FALSE, at, cuts = 20, pretty = FALSE, xlim = NULL, ylim = NULL, ply = NULL)

Arguments

grd SpatialGridDataFrame; spatial grid data frame.
zcol character or integer; attribute name or column number in attribute table.
level logical; if TRUE a set of levels is used to partition the range of z, its default is FALSE.
at numeric; a vector giving breakpoints along the range of z.
cuts integer; number of levels the range of z would be divided into.
pretty logical; whether to use pretty cut locations.
xlim numeric; vector of length 2 giving left and right limits of the spatial grid, data outside these limits is excluded.
ylim numeric; vector of length 2 giving lower and upper limits of the spatial grid, data outside these limits is excluded.
ply gpc.poly, SpatialPolygons, or SpatialGridDataFrame; cropping polygon.

Value

Returns an object of class SpatialPolygonsDataFrame. The objects data slot is a data frame, number of rows equal to the number of Polygons objects and a single column containing values of z. If level is TRUE, z values are set equal to the midpoint between breakpoints. The status of the polygon as a hole or an island is taken from the ring direction, with clockwise meaning island, and counter-clockwise meaning hole.

Note

The traditional R graphics model does not draw polygon holes correctly, holes overpaint their containing Polygon object using a user defined background color (white by default). Polygon holes are now rendered correctly using the plot method for spatial polygons (SpatialPolygons-class), see polypath for more details. The Trellis graphics model appears to rely on the traditional method so use caution when plotting with spplot.
Grid2Polygons

Author(s)
J.C. Fisher

References
A general explanation of the algorithm provided here; inspiration provided here.

See Also
SpatialPolygons

Examples

# Example 1

```r
z <- c(1.1, 1.5, 4.2, 4.1, 4.3, 4.7,
      1.2, 1.4, 4.8, 4.8, NA, 4.1,
      1.7, 4.2, 1.4, 4.8, 4.0, 4.4,
      1.1, 1.3, 1.2, 4.8, 1.6, NA,
      3.3, 2.9, NA, 4.1, 1.0, 4.0)

m <- 5
n <- 6
x <- rep(0:n, m + 1)
y <- rep(0:m, each = n + 1)
xc <- c(rep(seq(0:n, n - 0.5, by = 1), m))
yc <- rep(rev(seq(0:n, m - 0.5, by = 1)), each = n)
grd <- data.frame(z, xc, yc)
coordinates(grd) <- ~ xc + yc
gridded(grd) <- TRUE
grd <- as(grd, "SpatialGridDataFrame")
image(grd, col = gray.colors(SP), axes = TRUE)
grid(col = "black", lty = 1)
points(x, y, pch = 16)
text(cbind(x, y), labels = z)
text(cbind(x = x + 0.1, y = rev(y + 0.1)), labels = 1:((m + 1) * (n + 1)),
     cex = 0.6)

at <- 1:ceiling(max(z, na.rm = TRUE))
plys <- Grid2Polygons(grd, level = TRUE, at = at)
cols <- rainbow(length(plys), alpha = 0.3)
plot(plys, add = TRUE, col = cols)
zz <- plys[[1]]
legend("top", legend = zz, fill = cols, bty = "n", xpd = TRUE,
       inset = c(0, -0.1), ncol = length(plys))

p1 <- Polygon(rbind(c(1.2, 0.5), c(5.8, 1.7), c(2.5, 5.1), c(1.2, 0.5)),
              hole = FALSE)
p2 <- Polygon(rbind(c(2.5, 2.5), c(3.4, 1.8), c(3.7, 3.1), c(2.5, 2.5)),
              hole = TRUE)
p3 <- Polygon(rbind(c(-0.3, 3.3), c(1.7, 5.1), c(-1.0, 7.0), c(-0.3, 3.3)),
              hole = FALSE)
p <- SpatialPolygons(list(Polygons(list(p1, p2, p3), 1)))
```
plys <- Grid2Polygons(grd, level = TRUE, at = at, ply = p)
plot(plys, col = rainbow(length(zz), alpha = 0.6)[zz in plys[[1]]], add = TRUE)

# Example 2

data(meuse.grid)
coordinates(meuse.grid) <- ~ x + y
gridded(meuse.grid) <- TRUE
meuse.grid <- as(meuse.grid, "SpatialGridDataFrame")
meuse.plys <- Grid2Polygons(meuse.grid, "dist", level = FALSE)
op <- par(mfrow = c(1, 2), oma = rep(0, 4), mar = rep(0, 4))
plot(meuse.plys, col = heat.colors(length(meuse.plys)))
title("level = TRUE", line = -7)

meuse.plys.lev <- Grid2Polygons(meuse.grid, "dist", level = TRUE)
plot(meuse.plys.lev, col = heat.colors(length(meuse.plys.lev)))
title("level = TRUE", line = -7)
par(op)
Index

*Topic manip
  Grid2Polygons, 2

Grid2Polygons, 2

polypath, 2

SpatialGridDataFrame, 2
SpatialPolygons, 3
SpatialPolygonsDataFrame, 2
spplot, 2