

# Package ‘graticule’

August 29, 2016

**Type** Package

**Title** Meridional and Parallel Lines for Maps

**Version** 0.1.2

**Description** Create graticule lines and labels for maps. Control the creation of lines by setting their placement (at particular meridians and parallels) and extent (along parallels and meridians). Labels are created independently of lines.

**License** GPL-3

**LazyData** TRUE

**Depends** sp

**Imports** raster

**Suggests** devtools, knitr, maptools, oce, rgdal, rworldmap, testthat

**VignetteBuilder** knitr

**BugReports** <https://github.com/mdsumner/graticule/issues>

**URL** <https://github.com/mdsumner/graticule>

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Michael D. Sumner [aut, cre]

**Maintainer** Michael D. Sumner <mdsumner@gmail.com>

**Repository** CRAN

**Date/Publication** 2016-02-02 12:16:29

## R topics documented:

graticule . . . . .	2
graticule_labels . . . . .	3
pathologicule . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

graticule

*graticule: graticule lines for maps*


---

## Description

graticule: graticule lines for maps

Specify the creation of lines along meridians by specifying their placement at particular lons (longitudes) and lats (latitudes) and their extents with xlim (extent of parallel line in longitude) and ylim (extent of meridional line in latitude).

## Usage

```
graticule(lons, lats, nverts = 60, xlim, ylim, proj = NULL, tiles = FALSE)
```

## Arguments

lons	longitudes for meridional lines
lats	latitudes for parallel lines
nverts	number of discrete vertices for each segment
xlim	maximum range of parallel lines
ylim	maximum range of meridional lines
proj	optional proj.4 string for output object
tiles	if TRUE return polygons as output

## Details

Provide a valid PROJ.4 string to return the graticule lines in this projection. If this is not specified the graticule lines are returned in their original longlat / WGS84.

The arguments xlim, ylim and nverts are ignored if tiles is TRUE.

## Examples

```
## Not run:
library(rgdal)
x <- as.matrix(expand.grid(x = seq(100, 240, by = 15), y = seq(-85, -30, by = 15)))
prj <- "+proj=laea +lon_0=180 +lat_0=-70 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs "
px <- project(x, prj)
g <- graticule(unique(x[,1]), unique(x[,2]))
pg <- spTransform(g, CRS(prj))
plot(px, type = "n")
plot(pg, add = TRUE)

g2 <- graticule(unique(x[,1]), unique(x[,2]), ylim = c(-90, 0), xlim = c(110, 250))
pg2 <- spTransform(g2, CRS(prj))
plot(px, type = "n")
plot(pg2, add = TRUE)
```

```

prj <- "+proj=laea +lon_0=0 +lat_0=-90 +ellps=WGS84"
xx <- c(-120, -100, -80, -60, -40); yy <- c(-65, -55, -45)
g3 <- graticule(xx, yy, ylim = c(-70, -30), proj = prj)
g3labs <- graticule_labels(xx, c(-65, -45), xline = -85, yline = -30, proj = prj)
plot(g3)
text(g3labs, lab = parse(text = g3labs$lab))

## polygonal graticule on Orthographic projection
xx <- seq(-90, 90, length = 10) + 147
yy <- seq(-90, 90, length = 5)
portho <- "+proj=ortho +lon_0=147 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs"
g <- graticule(xx, yy, proj = portho, tiles = TRUE)
plot(g, col = c("black", "grey"))

library(maptools)
data(wrld_simpl)
w <- spTransform(subset(wrld_simpl, NAME == "Australia"), CRS(projection(g)))
plot(w, add = TRUE, border = "dodgerblue")

## End(Not run)

```

---

graticule\_labels      *Create graticule labels.*

---

## Description

Returns a set of points with labels, for plotting in conjunction with [graticule](#).

## Usage

```
graticule_labels(lons, lats, xline, yline, proj = NULL)
```

## Arguments

lons	longitudes for meridional labels
lats	latitudes for parallel labels
xline	meridian/s for placement of parallel labels
yline	parallel/s for placement of meridian labels
proj	optional proj.4 string for output object

## Details

SpatialPoints are returned in the projection of proj if given, or longlat / WGS84.

**Examples**

```
xx <- c(100, 120, 160, 180)
yy <- c(-80,-70,-60, -50,-45, -30)
prj <- "+proj=lcc +lon_0=150 +lat_0=-80 +lat_1=-85 +lat_2=-75 +ellps=WGS84"
plot(graticule(lons = xx, lats = yy, proj = prj))
labs <- graticule_labels(lons = xx, lats = yy, xline = 100, yline = -80, proj = prj)
op <- par(xpd = NA)
text(labs, lab = parse(text = labs$lab), pos = c(2, 1)[labs$islon + 1], adj = 1.2)
par(op)
```

---

pathologicule

---

*Create a mesh of evenly spaced lines in another projection.*


---

**Description**

Create a mesh of evenly spaced lines in another projection.

**Usage**

```
pathologicule(x, proj)
```

**Arguments**

x	object to build line mesh for
proj	the other projection

**Value**

spatial object

**Examples**

```
## Not run:
library(maptools)
data(wrld_simpl)
library(raster)
w <- subset(wrld_simpl, NAME == "Australia")
plot(w)
laea <- pathologicule(w, "+proj=laea +lon_0=147 +lat_0=-42 +ellps=WGS84")
stere <- pathologicule(w, "+proj=stere +lon_0=147 +lat_0=-42 +ellps=WGS84")
plot(laea, add = TRUE, col = "dodgerblue")
plot(stere, add = TRUE, col = "firebrick")
pst90 <- "+proj=stere +lat_0=-90 +ellps=WGS84"
p <- spTransform(subset(wrld_simpl, coordinates(wrld_simpl)[,2] < -20), pst90)
plot(extent(p) + 1e6, asp = 1, type = "n"); plot(p, add = TRUE)
laea <- pathologicule(p, "+proj=laea +lon_0=147 +lat_0=-72 +ellps=WGS84")
stere <- pathologicule(p, "+proj=stere +lon_0=147 +lat_0=-42 +ellps=WGS84")
plot(laea, add = TRUE, col = "dodgerblue")
plot(stere, add = TRUE, col = "firebrick")
```

## End(Not run)

# Index

graticule, [2](#), [3](#)  
graticule-package (graticule), [2](#)  
graticule\_labels, [3](#)  
pathologicule, [4](#)