

# Package ‘graticule’

September 25, 2023

**Type** Package

**Title** Meridional and Parallel Lines for Maps

**Version** 0.4.0

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

**License** GPL-3

**Depends** sp

**Imports** raster, utils, geosphere, stats, reproj (>= 0.4.3)

**Suggests** devtools, knitr, spex, testthat (>= 2.1.0), rmarkdown, covr

**VignetteBuilder** knitr

**Encoding** UTF-8

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

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

**RoxygenNote** 7.2.3

**NeedsCompilation** no

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

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

**Repository** CRAN

**Date/Publication** 2023-09-25 10:00:02 UTC

## R topics documented:

graticule . . . . .	2
graticule_labels . . . . .	3
lonlat . . . . .	4

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

---

`graticule`*Create graticule lines.*

---

### Description

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 = NULL, xlim, ylim, proj = NULL, tiles = FALSE)
```

### Arguments

<code>lons</code>	longitudes for meridional lines
<code>lats</code>	latitudes for parallel lines
<code>nverts</code>	number of discrete vertices for each segment
<code>xlim</code>	maximum range of parallel lines
<code>ylim</code>	maximum range of meridional lines
<code>proj</code>	optional proj.4 string for output object
<code>tiles</code>	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. All segments are discretized as `_rhumb_lines_` at `'getOption("graticule.mindist")'` metres, which defaults to `'5e4'`. The arguments `xlim`, `ylim` and `nverts` are ignored if `tiles` is TRUE.

### Value

SpatialLines or SpatialPolygons object

### Examples

```
graticule()
```

---

graticule_labels	<i>Create graticule labels.</i>
------------------	---------------------------------

---

### 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.

### Value

SpatialPoints object with labels for downstream use

### 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)
```

---

lonlat                      *Add longitude latitude lines to a plot*

---

### Description

Use the coordinates of the input raster to generate coordinate rasters, these are then used in a contour plot.

### Usage

```
lonlat(  
  x,  
  na.rm = FALSE,  
  lon = FALSE,  
  lat = FALSE,  
  ...,  
  plot = TRUE,  
  add = TRUE  
)
```

### Arguments

x	input raster
na.rm	logical, remove missing values from generated coordinates
lon	if TRUE, only longitude plotted
lat	if TRUE (and 'lon = FALSE') only latitude plotted
...	passed to [graphics::contour()]
plot	logical, plot the result
add	logical, add to current plot or instantiate one

### Details

Plot is added to an existing plot by default.

### Value

RasterBrick of the longitude and latitude values, two layers

(invisibly) the raster (RasterBrick) object with longitude and latitude values of the input as two layers, otherwise this function used for side-effect (drawing on a plot)

**Examples**

```
plot(c(-180, 180), c(-90, 90))  
lonlat(raster::raster())
```

```
p <- raster::projectExtent(raster::raster(), "+proj=igh")  
lonlat(p, add = FALSE)  
lonlat(p, levels = seq(-180, 180, by = 15), add = FALSE)
```

```
lonlat(p, levels = seq(-180, 180, by = 5), add = FALSE, lon = TRUE)  
lonlat(p, levels = seq(-180, 180, by = 15), add = TRUE, lat = TRUE)
```

# Index

graticule, [2](#), [3](#)  
graticule\_labels, [3](#)  
lonlat, [4](#)