Package: tissot (via r-universe)

September 6, 2024

Type Package

Title The Tissot Indicatrix

Version 0.0.1.9006

Description Create and plot the Tissot Indicatrix.

License GPL-3

Depends R (>= 4.0.0)

Imports tibble, reproj (>= 0.6.0)

RoxygenNote 7.2.3

LazyData true

LazyDataCompression xz

Encoding UTF-8

Additional_repositories `https://hypertidy.r-universe.dev`

URL `https://github.com/hypertidy/tissot`,
`https://hypertidy.github.io/tissot/`

BugReports `https://github.com/hypertidy/tissot/issues`

Repository https://hypertidy.r-universe.dev

RemoteUrl https://github.com/hypertidy/tissot

RemoteRef HEAD

RemoteSha d7b2685782262e4fa8580d412b0dfdcf5ed5468f

Contents

`indicatrix` *Indicatrix*

Description

Indicatrix

plot indicatrix

Usage

```
indicatrix(x, scale = 1, ...)

## S3 method for class 'indicatrixes'
plot(
  x,
  asp = 1,
  xlab = "x",
  ylab = "y",
  add = FALSE,
  ...,
  col.base = rgb(0, 0, 0, 0.1),
  col.lambda = grey(0.75),
  col.phi = "#1b9e77",
  col.major = "#7570b3",
  col.minor = "#d95f02",
  col.outline = "black"
)

indicatrix0(x, scale = 1, ...)

## S3 method for class 'indicatrix0'
plot(
  x,
  asp = 1,
  xlab = "Easting",
  ylab = "Northing",
  add = FALSE,
  ...,
  col.base = rgb(0, 0, 0, 0.1),
  col.lambda = grey(0.75),
  col.phi = "#45A271",
  col.major = "#A782C3",
  col.minor = "#C87A8A",
  col.outline = "black"
)
```

Arguments

`x`	object from `tissot`
`scale`	scaling
`...`	arguments `n`, `from` and `to` passed to `ti_ellipse` function
`asp`	aspect ratio
`xlab`	x-axis labels
`ylab`	y-axis labels
`add`	add to existing plot
`col.base`	colour of base
`col.lambda`	colour of lambda
`col.phi`	colour of phi
`col.major`	major axis colour
`col.minor`	minor axis colour
`col.outline`	outline colour

Details

Reprocesses the output of `tissot` into convenient geometrical data.

`tissot` *Tissot*

Description

Create the Tissot Indicatrix.

Usage

```
tissot(
  lambda,
  phi = NULL,
  degrees = TRUE,
  A = 6378137,
  f.inv = 298.257223563,
  ...,
  proj.in,
  proj.out
)
```

Arguments

`lambda`	longitude
`phi`	latitude
`degrees`	logical, work in degrees or radians
`A`	ellipsoidal semi-major axis (meters)
`f.inv`	the inverse flattening
`...`	passed to internal function
`proj.in`	projection of input
`proj.out`	projection of context

Details

Compute properties of scale distortion and Tissot's indicatrix at location x = c(lambda, phi) using prj as the projection. A is the ellipsoidal semi-major axis (in meters) and f.inv is the inverse flattening. The projection must return a vector (x, y) when given a vector (lambda, phi). (Not vectorized.) Optional arguments ... are passed to prj. Source: Snyder pp 20-26 (WGS 84 defaults for the ellipsoidal parameters). All input and output angles are in degrees.

Value

list with stuff as per below

Examples

```
x <- seq(-175, 175, by = 15)
y <- seq(-82.5, 82.5, by = 15)
xy <- expand.grid(x, y)
r <- tissot(xy,
            proj.in= "OGC:CRS84",
            proj.out= "+proj=robin")

j <- which.min(abs(135 - r$lon) + abs(54 - r$lat))
idx0 <- indicatrix0(r[j, ], scale=10^4, n=71)
op <- par(mfrow =   c(2, 1))
plot(idx0, add = FALSE)
idx <- indicatrix(r, scale=3e5, n=71)
plot(idx, add = FALSE)
tissot_abline(r$lon[j], r$lat[j])
par(op)
```

tissot_map *Get last plot projection*

Description

'tissot_map()' will add the [world] coastline to any map.

Usage

```
tissot_map(..., add = TRUE)

tissot_abline(lambda, phi = NULL, ..., proj.in = NULL)

tissot_get_proj()
```

Arguments

`...`	graphical parameters for [lines()] if 'add = TRUE', or for [plot()] if 'add = FALSE'
`add`	logical, default 'TRUE' add to existing plot or create new
`lambda`	longitude at which to draw a vertical line
`phi`	latitude at which to draw a horizontal line
`proj.in`	projection for expert use

Details

'tissot_get_proj()' When the indicatrix is plotted it registers its projection. This string can be obtained with this getter function.

'tissot_abline()' will draw a vertical and horizontal line at a give longitude latitude (where they intersect is the actual lon,lat location)

Value

'tissot_map()' returns the internal world map data (projected if one is current) as a matrix

'tissot_abline()' called for its side effect of drawing on the plot

'tissot_get_proj()' returns the value of the current projection, or NULL

`ti_ellipse` *Ellipse*

Description

Ellipse

Usage

```
ti_ellipse(center, axes, scale = 1, n = 36, from = 0, to = 2 * pi)
```

Arguments

`center`	center
`axes`	axes
`scale`	scale
`n`	n
`from`	from
`to`	to

Value

matrix

`world` *world coastline*

Description

A modified matrix version of data from the maps package.

Usage

```
world
```

Format

An object of class `matrix` (inherits from `array`) with 82403 rows and 2 columns.

Details

Basically longitudes have been smooshed to 'abs(lon) < 180'

Index