Package 'tissot'

May 8, 2026

Title Tissot Indicatrix for Map Projection Distortion

Version 0.2.0

Description Compute and visualize the 'Tissot Indicatrix' for map projections. The indicatrix characterizes projection distortion by computing scale factors, angular deformation, areal distortion, and convergence at arbitrary points. Based on the calculations shared by Bill Huber on <https://gis.stackexchange.com/a/5075/482>. Uses 'GDAL' for coordinate transformation. Developed using the method published in Snyder, JP (1987) <doi:10.3133/pp1395>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.6.0)

Imports tibble, gdalraster, graphics, grDevices

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spelling

Config/testthat/edition 3

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

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

Language en-US

NeedsCompilation no

Author Michael Sumner [aut, cre] (ORCID: <https://orcid.org/0000-0002-2471-7511>), Bill Huber [aut] (original algorithm and calculations)

Maintainer Michael Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2026-02-12 08:10:02 UTC

Contents

`indicatrix` *Compute Tissot indicatrix ellipse geometry*

Description

Generates indicatrix objects for plotting at each point. Returns an `indicatrix_list` containing individual indicatrix objects.

Usage

```
indicatrix(x, target = NULL, ..., source = "EPSG:4326")
```

Arguments

`x`	a `tissot_tbl`, or any xy-ish input (see `tissot()`)
`target`	target projection CRS (extracted from `tissot_tbl` attributes if `x` is one; required otherwise)
`...`	passed to `tissot()`
`source`	source CRS (default `"EPSG:4326"`)

Details

`indicatrix()` accepts either:

- A `tissot_tbl` object (from `tissot()`) — projection is extracted from attributes, `target` is optional
- Any xy-ish input with an explicit `target`

Value

An `indicatrix_list` object (a list of `indicatrix` objects with `source` and `target` stored as attributes)

See Also

tissot(), plot.indicatrix(), plot.indicatrix_list(), ti_ellipse()

Examples

```
## From a tissot_tbl
r <- tissot(cbind(seq(-150, 150, by = 30), 0), "+proj=robin")
ii <- indicatrix(r)
plot(ii, scale = 5e5)

## From raw coordinates
ii2 <- indicatrix(c(0, 45), "+proj=stere +lat_0=90")
plot(ii2)
```

---

plot.indicatrix *Plot an indicatrix*

Description

Draws a single Tissot indicatrix ellipse on the current plot. The ellipse shows the distortion of a unit circle under the map projection. Optional overlays include a reference unit circle, and lambda/phi direction axes.

Usage

```
## S3 method for class 'indicatrix'
plot(
  x,
  scale = 1e+05,
  n = 72,
  col = "#FF990055",
  border = "black",
  add = TRUE,
  show.axes = TRUE,
  show.circle = TRUE,
  ...
)
```

Arguments

x	an indicatrix object (from indicatrix())
scale	scaling factor for the ellipse size in projected units
n	number of points on the ellipse
col	fill colour for the ellipse
border	border colour
add	logical; add to existing plot?

show.axes	TRUE, FALSE, or a named list of graphical parameters for the direction lines. Defaults: list(col.lambda = "red", col.phi = "blue", lwd = 1.5).
show.circle	TRUE, FALSE, or a named list of graphical parameters for the reference circle. Defaults: list(col = adjustcolor("white", alpha.f = 0.6), border = "grey70", lwd = 2.5, lty = 2).
...	passed to graphics::polygon()

Details

show.circle and show.axes accept TRUE (use defaults), FALSE (hide), or a named list of graphical parameters to override defaults. For example, show.circle = list(border = "blue", lty = 3).

Value

The input x, invisibly.

See Also

indicatrix(), plot.indicatrix_list(), ti_ellipse()

---

plot.indicatrix_list *Plot a list of indicatrixes*

Description

Draws all indicatrixes in an indicatrix_list, optionally creating a new plot or adding to an existing one. Can colour-code the fill by a distortion metric.

Usage

```
## S3 method for class 'indicatrix_list'
plot(
  x,
  scale = 1e+05,
  n = 72,
  col = "#FF990055",
  border = "black",
  add = FALSE,
  show.axes = TRUE,
  show.circle = TRUE,
  fill.by = NULL,
  palette = NULL,
  ncolors = 64L,
  ...
)
```

Arguments

x	an indicatrix_list (from indicatrix())
scale	scaling factor for ellipse size in projected units
n	number of points per ellipse
col	fill colour. If a single colour, used for all ellipses. If NULL and fill.by is set, colours are generated automatically.
border	border colour
add	logical; add to existing plot? If FALSE, creates a new plot sized to contain all ellipses.
show.axes	TRUE, FALSE, or a named list of graphical parameters. See plot.indicatrix() for defaults.
show.circle	TRUE, FALSE, or a named list of graphical parameters. Default TRUE for the list method (the circle-vs-ellipse comparison makes distortion visible at map scale). See plot.indicatrix() for defaults.
fill.by	character; name of a distortion metric to colour-code the fill. One of "scale_area", "angle_deformation", "scale_h", "scale_k", "scale_a", "scale_b". Default NULL (uniform fill).
palette	colour palette function (default grDevices::hcl.colors())
ncolors	number of colours in the palette (default 64)
...	passed to plot.indicatrix()

Value

The input x, invisibly.

See Also

indicatrix(), plot.indicatrix(), tissot_map()

Examples

```
xy <- expand.grid(seq(-150, 150, by = 30), seq(-60, 60, by = 30))
r <- tissot(xy, "+proj=robin")
ii <- indicatrix(r)

## Uniform fill
plot(ii, scale = 6e5, add = FALSE)
tissot_map()

## Colour by areal distortion
plot(ii, scale = 6e5, add = FALSE, fill.by = "scale_area")
tissot_map()
```

`tissot` *Compute Tissot indicatrix properties*

Description

Compute the Tissot indicatrix at given longitude/latitude locations for a map projection. Returns scale factors, angular deformation, convergence, and related distortion properties.

Usage

```
tissot(
  x,
  target,
  ...,
  source = "EPSG:4326",
  A = 6378137,
  f.inv = 298.257223563,
  dx = 1e-04
)
```

Arguments

`x`	input coordinates — any xy-ish object: a two-column matrix, data.frame, tibble, list with `x`/`y` or `lon`/`lat` components, or a length-2 numeric vector for a single point
`target`	target projection CRS string (required)
`...`	ignored
`source`	source CRS (default `"EPSG:4326"`)
`A`	semi-major axis of the ellipsoid (default WGS84)
`f.inv`	inverse flattening (default WGS84)
`dx`	finite difference step in degrees (default 1e-4)

Details

The Jacobian of the projection is computed via finite differences, projecting all base and offset points in a single batched call to `gdalraster::transform_xy()`. All subsequent calculations (SVD, distortion metrics) are fully vectorized.

Input 'x' is assumed to be longitude,latitude values, with default 'EPSG:4326'. Set 'source' for a different 'geographic' coordinate reference system.

Value

A `tissot_tbl` tibble with columns: x (lon), y (lat), dx_dlam, dy_dlam, dx_dphi, dy_dphi, scale_h, scale_k, scale_omega, scale_a, scale_b, scale_area, angle_deformation, convergence. The `source` and `target` CRS strings are stored as attributes.

See Also

indicatrix(), tissot_map(), tissot_unproject(), gdalraster::transform_xy()

Examples

```
tissot(c(0, 45), "+proj=robin")
tissot(cbind(seq(-180, 180, by = 30), 0), "+proj=robin")
```

---

tissot_get_proj *Get or set the current plot projection*

Description

Deprecated. Prefer passing `target` explicitly.

These functions access a global option. Prefer passing `target` explicitly to `tissot_map()` and `tissot_abline()`, or let `plot.indicatrix_list()` set the projection automatically.

Usage

```
tissot_get_proj()

tissot_set_proj(target)
```

Arguments

`target`	projection CRS string

Value

`tissot_get_proj()` returns the current projection string or `NULL`

---

tissot_map *Plot world coastline on a projected map*

Description

`tissot_map()` draws the bundled world coastline, projected if a projection is current. The projection is determined in this order:

1. An explicit `target` argument
2. The projection stored by the last `plot.indicatrix_list()` call

Usage

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

tissot_abline(x, y = NULL, ..., source = "EPSG:4326", target = NULL)
```

Arguments

`...`	graphical parameters passed to `graphics::lines()` (if adding) or `graphics::plot()` (if creating new)
`target`	target CRS. If `NULL`, uses the last plot projection (from `plot.indicatrix_list()`) or draws in lon/lat.
`add`	logical; add to existing plot (default `TRUE`) or create new
`x`	longitude values (or any xy-ish input; see `tissot()`)
`y`	latitude values (ignored if `x` is a matrix)
`source`	source CRS for the coordinates (default `"EPSG:4326"`)

Details

`tissot_abline()` draws vertical and horizontal reference lines at a given longitude/latitude in projected coordinates.

Value

`tissot_map()` invisibly returns the (projected) world coastline matrix

`tissot_abline()` is called for its side effect

See Also

`plot.indicatrix_list()`, `tissot()`

Examples

```
r <- tissot(cbind(seq(-150, 150, by = 30), 0), "+proj=robin")
ii <- indicatrix(r)
plot(ii, scale = 6e5, add = FALSE)
tissot_map()
```

`tissot_unproject` *Unproject coordinates to geographic (lon/lat) CRS*

Description

A convenience wrapper around `gdalraster::transform_xy()` for converting projected coordinates to geographic. Useful for generating regular grids in a projected CRS to feed to `tissot()`, which requires lon/lat input.

Usage

```
tissot_unproject(x, target = "EPSG:4326", ..., source = NULL)
```

Arguments

x	input coordinates — any xy-ish object: a two-column matrix, data.frame, tibble, list with x/y or lon/lat components, or a length-2 numeric vector for a single point
target	target CRS string (default "EPSG:4326"). Must be geographic.
...	ignored
source	source CRS string (required). Must be a projected CRS.

Value

A two-column matrix of longitude and latitude values.

See Also

tissot(), gdalraster::transform_xy()

Examples

```
## regular grid in UTM zone 55S, unprojected to lon/lat for tissot()
xy <- expand.grid(x = seq(4e5, 6e5, length.out = 5),
                  y = seq(5200000, 5400000, length.out = 4))
ll <- tissot_unproject(xy, source = "EPSG:32755")
tissot(ll, "+proj=utm +zone=55 +south")
```

---

ti_ellipse *Generate ellipse coordinates for an indicatrix*

Description

Generate ellipse coordinates for an indicatrix

Usage

```
ti_ellipse(x, scale = 1e+05, n = 72, ...)
```

Arguments

x	an indicatrix object
scale	scaling factor for the ellipse size in projected units
n	number of points on the ellipse
...	ignored

Value

A two-column matrix of projected coordinates tracing the ellipse

See Also

indicatrix(), plot.indicatrix()

---

world *World coastline*

Description

A matrix of longitude/latitude coordinates representing a simplified world coastline, derived from the maps package. Continent boundaries are separated by NA rows. Longitudes are constrained to abs(lon) <= 180.

Usage

```
world
```

Format

A two-column numeric matrix (longitude, latitude)

[.indicatrix_list *Subset an indicatrix_list*

Description

Subset an indicatrix_list

Usage

```
## S3 method for class 'indicatrix_list'
x[i]
```

Arguments

x	an indicatrix_list
i	indices

Value

An indicatrix_list with the selected elements

Index