# Package 'latbias'

October 25, 2025

**Type** Package

**Title** Calculate the Latitudinal Bias Index

**Version** 1.0.0

**Description** Studies that report shifts in species distributions may be biased
by the shape of the study area. The main functionality of this package is to
calculate the Latitudinal Bias Index (LBI) for any given shape. The LBI is
bounded between +1 (100% probability to exclusively record latitudinal
shifts, i.e., range shifts data sampled along a perfectly South-North
oriented straight line) and -1 (100% probability to exclusively record
longitudinal shifts, i.e., range shifts data sampled along a perfectly
East-West oriented straight line).

**Depends** R (>= 3.5.3)

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** dplyr, geosphere, ggplot2, psych, RColorBrewer, reshape2, sf,
sp, terra, tidyr, units

**RoxygenNote** 7.3.3

**Suggests** elevatr, knitr, progress, rmarkdown, rnaturalearth,
rnaturalearthdata, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** https://github.com/pierredenelle/latbias,

**BugReports** https://github.com/pierredenelle/latbias/issues

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Pierre Denelle [aut, cre] (ORCID:
<https://orcid.org/0000-0001-5037-2281>),
Pieter Sanczuk [aut] (ORCID: <https://orcid.org/0000-0003-1107-4905>)

**Maintainer** Pierre Denelle <pierre.denelle@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-10-25 12:40:02 UTC

# Contents

---

LBI                              *LBI*

---

**Description**

Computes the Latitudinal Bias Index (LBI) for a given shapefile, by calculating the distance between two random locations within the shape multiple times (see details).

**Usage**

```
LBI(
  study_area_id,
  study_area_polygon,
  nobs = 250,
  nboot = 1000,
  fact_location = 10,
  elevation = NULL,
  raw_output = FALSE
)
```

**Arguments**

study_area_id    Character string. Name ID of the study case area or country name.

study_area_polygon

Polygon shapefile. It should be a sfc object, of class POLYGON or MULTI-POLYGON.

nobs             Numeric. Number of random observations in each sample. 250 by default.

nboot            Numeric. Determines how many times the random shifts are calculated. 1,000 by default.

fact_location    Numeric. fact_location x nobs determine all the possible coordinates that can be sampled within the provided polygon for all bootstraps (faster than to generate a set of random location at each bootstrap)

elevation        Elevation raster. elevation in wgs84; if not provided, NA will be returned for null-model elevational shifts.

raw_output       Logical. FALSE by default. If TRUE, all bootstraps are returned as a data.frame. => say that the raw outputs are accessible

## Details

The main output contains the following columns:

- study_area_id: ID or name of the study case region

- distance_km: average expected geographic distance shift between t1 and t2

- null_mod_SN_shift: average expected South-North shift between t1 and t2, in absolute values

- null_mod_EW_shift: average expected East-West shift between t1 and t2, in absolute values

- null_mod_elevation_shift: average expected elevation shift between t1 and t2, in absolute values

- LBI: the Latitudinal Bias Index value

LBI formula is

$$LBI = 2 \times \left( \frac{mean(|\frac{A_n lat}{A_n lon}|)}{1 + mean(|\frac{A_n lat}{A_n lon}|)} - 0.5 \right)$$

with $A_n lat$ and $A_n lon$ denoting the geographic displacement of the centroid positions of both sets of observation in the nth iteration by means in the latitudinal and the longitudinal dimension.

## Value

A data.frame or a list of two data.frames if raw_output is set to TRUE.

## References

Sanczuk et al. submitted.

## Examples

```
study_area <- rnaturalearth::ne_countries(scale = 110, continent = "Europe",
country = "Sweden", type = "map_units", returnclass = "sf")
study_area <- sf::st_union(study_area)

LBI(study_area_id = "Sweden", study_area_polygon = study_area,
nobs = 10, nboot = 10, fact_location = 5, elevation = NULL)

# With elevation
elevation_df <- elevatr::get_elev_raster(
locations = sf::st_as_sf(study_area), z = 5)
LBI(study_area_id = "Sweden", study_area_polygon = study_area,
nobs = 10, nboot = 10, fact_location = 5, elevation = elevation_df)
```

---

| windrose | *windrose* |
|---|---|

---

## Description

Plot the directional shifts for a given bootstrap.

## Usage

```
windrose(
  data,
  spd,
  dir,
  spdres = 30,
  dirres = 30,
  spdmin = 0,
  spdmax = 150,
  spdseq = NULL,
  palette = "Spectral",
  countmax = NA,
  debug = 0
)
```

## Arguments

| | |
|---|---|
| data | ddata frame with two columns including the bearing and distance or speed of shifts |
| spd | distance or speed column |
| dir | bearing column |
| spdres | 30 by default. Numeric; resolution for plotting distance of speed categories; adapt here the legend and breaks. |
| dirres | 30 by default. Numeric, resolution for plotting bearings on windrose; adapt here the number of bars plotted on the windrose. |
| spdmin | 0 by default. Numeric, minimal distance/speed for plotting. |
| spdmax | 150 by default. Numeric, maximal distance/speed for plotting. |
| spdseq | NULL |
| palette | Character string defining the color palette to be used. Available options are taken from rownames(RColorBrewer::brewer.pal.info). |
| countmax | NA by default. Numeric, optional, adjust the y-axis limit (maximum) |
| debug | 0 by default. Numeric, if >0, run debug to find optimal bins of spd and dir. |

## Details

This is intended to plot raw simulation output (that is, bearing and distance of random shifts) on a windrose.

## Value

Windrose plot, ggplot object.

## References

Sanczuk et al. submitted.

## Examples

```
study_area <- rnaturalearth::ne_countries(scale = 110, continent = "Europe",
country = "Sweden", type = "map_units", returnclass = "sf")
study_area <- sf::st_union(study_area)
po <- LBI(study_area_id = "Sweden", study_area_polygon = study_area,
nobs = 10, nboot = 10, fact_location = 5, elevation = NULL,
raw_output = TRUE)
test <- as.data.frame(po$all[, c("study_area_id", "distance_km",
"bearing", "rep")])
test$distance_km <- as.numeric(test$distance_km)
pop <- windrose(data = test, spd = "distance_km", dir = "bearing")
```

# Index