

# Package ‘spotr’

May 9, 2026

**Title** Estimate Spatial Population Indices from Ecological Abundance Data

**Version** 0.1.0

**Description** Compute relative or absolute population trends across space and time using predictions from models fitted to ecological population abundance data, as described in Knape (2025) <[doi:10.1016/j.ecolind.2025.113435](https://doi.org/10.1016/j.ecolind.2025.113435)>. The package supports models fitted by 'mgcv' or 'brms', and draws from posterior predictive distributions.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** testthat (>= 3.0.0), brms, knitr, rmarkdown, sf, ggplot2, dplyr

**Config/testthat/edition** 3

**LinkingTo** Rcpp

**Imports** stats, mgcv, Rcpp

**Depends** R (>= 3.6.0)

**LazyData** true

**Config/Needs/website** rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Jonas Knape [aut, cre] (ORCID: <<https://orcid.org/0000-0002-8012-5131>>)

**Maintainer** Jonas Knape <jonas.knape@slu.se>

**Repository** CRAN

**Date/Publication** 2025-04-24 17:00:02 UTC

## Contents

cuckoo . . . . .	2
index . . . . .	2
swe_map . . . . .	4

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

---

cuckoo	<i>Counts of common cuckoo from the Swedish Bird Survey</i>
--------	---

---

**Description**

Counts per survey route and year between 2000 and 2021 of common cuckoos from the Swedish Bird Survey.

**Usage**

cuckoo

**Format**

cuckoo:

A data frame with 6 columns.

**count** Number of individuals counted.

**yr** Year

**route** Id of the survey route.

**county** County where route is located.

**lon** Longitude

**lat** Latitude

**Source**

[doi:10.15468/hd6w0r](https://doi.org/10.15468/hd6w0r)

---

index	<i>Compute scaled or absolute population indices from a fitted model object, or from posterior samples.</i>
-------	---

---

**Description**

Compute scaled or absolute population indices from a fitted model object, or from posterior samples.

**Usage**

```
index(
  object,
  newdata,
  timevar,
  ...,
  byvar = NULL,
```

```

type = "group",
weights = NULL,
bweights = NULL,
baseline = NULL,
alpha = c(0.8, 0.95),
nsamp = NULL
)

```

### Arguments

object	A matrix or an object of class gam or brmsfit. If a matrix, columns should correspond to (posterior) samples of abundance predictions and rows should match the rows in newdata.
newdata	A data frame containing a time variable, any grouping variable, and any variables needed for predicting from object. The data frame will be supplied to the predict method for gam and brms models. If object is of class gam, any variables needed for prediction that are not available in newdata will be treated as constant when computing index. For objects of class brmsfit, all variables needed for predicting need to be available in newdata. If object is a matrix, the rows of newdata should correspond to the rows of the matrix.
timevar	The name of the time variable in newdata over which an index should be computed.
...	Further arguments passed to predict functions.
byvar	Name of grouping variable in newdata. The default is NULL in which case a single index is computed. If not null an index is computed for each unique value of the grouping variable.
type	Type of index to compute, one of "group", "global", "delta", "raw". If "group", indices for each group are computed relative to the within group baseline. If "global", relative indices for each group are computed relative to the global baseline. If "delta", indices for each group are computed relative to the previous time point. If "raw", absolute (as opposed to relative) indices are computed.
weights	Weights for prediction points.
bweights	Weights for the baseline. If the argument is NULL, the baseline weights are set to be equal to the weights for the prediction points divided by the number of time points in the baseline (this assumes that newdata is balanced).
baseline	A set of time points that should be used as baseline for indices of type "group" or "global". The mean of the index over these time points will be one (see Knappe 2023). If missing, the first time point will be used as the baseline.
alpha	A vector of alpha levels for computing confidence intervals.
nsamp	Number of simulation samples to draw from gam objects. Defaults to NULL, in which case 1000 samples will be drawn.

### Details

The function computes spatio-temporal indices of relative population size using post-stratification of model based predictions, as detailed in Knappe (2025). This is done by summing (weighted) pre-

dicted abundances across the prediction grid defined by newdata for the numerator and denominator of the relative index.

**Warning:**

Large prediction tasks can require substantial memory. Use less rows in newdata, and/or fewer simulation samples to reduce memory footprint.

**Value**

A data frame containing indices and their uncertainties.

**References**

Knape, J. (2023). Effects of choice of baseline on the uncertainty of population and biodiversity indices. *Environmental and Ecological Statistics*, 30, 1–16. [doi:10.1007/s10651022005507](https://doi.org/10.1007/s10651022005507)

Knape, J. (2025). Spatially varying population indices. *Ecological Indicators*, 174, 113435. [doi:10.1016/j.ecolind.2025.113435](https://doi.org/10.1016/j.ecolind.2025.113435)

**Examples**

```
library(mgcv)
data(cuckoo)

# Simple model with abundance varying by year and latitude only.
gam_fit = gam(count ~ s(yr, lat), data = cuckoo, family = quasipoisson)

# Compute index relative to first year at three example latitudes
nd = expand.grid(yr = unique(cuckoo$yr), lat = c(55,60,65))
index(gam_fit, time = "yr", newdata = nd)
```

---

swe\_map

*Map of Swedish counties*

---

**Description**

Polygons of Swedish counties and their areas.

**Usage**

swe\_map

**Format**

**cuckoo:**  
An sf data frame with 3 columns.  
**county** Name of county.  
**area** Area of county in km<sup>2</sup>.  
**geometry** sf geometry column

**Source**

<https://www.scb.se/hitta-statistik/regional-statistik-och-kartor/regionala-indelningar/digitala-granser/>

# Index

\* **datasets**

cuckoo, [2](#)  
swe\_map, [4](#)

cuckoo, [2](#)

index, [2](#)

swe\_map, [4](#)