Objects in NIcalc

Bård Pedersen

11 4 2021

This vignette describes the structure and format of `S3` classes defined within `NIcalc`. These are lists containing input data to calculations of the Nature Index and objects containing the results from the calculation.

Data input to index calculations

Introduction

The Nature Index is calculated as a weighted average of scaled indicator observations, most often for a specified major ecosystem, in a set of defined spatial unit (NIunits), and for a series of years.

Indicator observations are collected from indicator-specific spatial units (ICunits) that may have a different spatial extent than the NIunit. The spatial delineation of indicator areas may also vary among indicators. Each ICunit and NIunit consist of one or more basic spatial units (BSunit). The Norwegian implementation of the Nature Index uses municipalities as of 01.01.2010 as BSunits.

Scaling of observations to a common scale is done using nonlinear scaling functions with only one parameter, the socalled reference value. There are two types of scaling functions, LOW and MAX. Indicator observations and reference values are specified as probability distributions. Such distributions are represented in the input data either as parameter values of two-parameter model distributions or as distribution objects generated by the `distr` package.

Weights assigned to indicator observations depend on indicators' specificity / fidelity to the respective major ecosystem, a grouping of the indicators into trophic groups and key indicators, and the area covered by the major ecosystem in question in BSunits within the intersections between each ICunit and the NIunit.

In order to calculate the Nature Index, characteristics of indicators, indicator observations, reference values, as well as BS-, IC- and NIunits must be provided. These data are stored in the the Nature Index database, which is an SQL relational database. The database consists of a set of main tables containing data on the most important object types. Further, it consists of several lookup tables that mostly contain information and names of the features that characterize the objects in the main tables. The relevant data are found distributed over several tables in the Nature Index database, and the information is not easily combined into simple data structures or objects.

NIcalc therefore contains a set of utility functions especially adapted to the structure of the Nature Index database and to the requirements set by the framework for calculating the index. `importDatasetApi` harvest datasets from the database via the The Nature Index Application Programming Interface. `checkInputData` checks whether a candidate dataset contains all the necessary data objects and variables for calculating the Nature Index, and whether the data contain consistent information. `assembleNiObject` calls `checkInputData` before it assembles and structures data into a complete and consistent dataset for calculating the Nature Index. `importDatasetApi` and `assembleNiObject` return lists of class `niDataImport` and `niInput` respectively.

Class `niDataImport` lists

Lists of class `niDataImport` are `S3` objects returned by the function `importDatasetApi`. They represent candidate data sets for calculating the Nature Index or a thematic index imported from the Nature Index database.

`niDataImport` lists contain the following elements:

Element	Name	Class	Explanation / Specification
[[1]]	indicators	data.frame	$id - integer $name - character $active - logical - active or passive indicator $keyElement - logical - key indicator or not $functionalGroupId - integer $functionalGroup - character $scalingModelId - integer $scalingModel - character in addition one or more variables with fidelities to major ecosystems, depending on parameters in `importDatasetApi`
[[2]]	referenceValues	list of 2	
	$referenceValues	data.frame	$indId - integer $indName - character $ICunitId - integer $ICunitName - character $yearId - integer - id for reference values $yearName - character - i.e. “Referenseverdi” $expectedValue - numeric $lowerQuantile - numeric $upperQuantile - numeric $customDistributionUUID - character - name of corresponding distribution object $distributionFamilyId - integer $distributionFamilyName - character $distParameter1 - numeric $distParameter2 - numeric
	$customDistributions	list	list of named distribution objects
[[3]]	indicatorObservations	list of 2	
	$indicatorValues	data.frame	$indId - integer $indName - character $ICunitId - integer $ICunitName - character $yearId - integer $yearName - character $expectedValue - numeric $lowerQuantile - numeric $upperQuantile - numeric $customDistributionUUID - character - name of corresponding distribution object $distributionFamilyId - integer $distributionFamilyName - character $distParameter1 - numeric

Element	Name	Class	Explanation / Specification
			$distParameter2 - numeric
	$customDistributions	list	list of named distribution objects
[[4]]	ICunits	data.frame	 $id - integer $name - character $BSunitId - integer $indId - integer
[[5]]	BSunits	data.frame	 $id - integer $name - character in addition variables describing BSunits, depending on parameters in `importDatasetApi` and the content of the `Kommune` table in the Nature Index database
[[6]]	ecosystems	data.frame	 $id - integer $name - character

Class `niInput` lists

Lists of class `niInput` are `S3` objects returned by the function `assembleNiObject`. They represent data sets that are controlled for consistency and for including all the necessary data objects for calculating the Nature Index or a thematic index. `calculateIndex` requires that datasets for calculating indices are entered as `niInput` objects.

`niInput` lists are structured into the following elements:

Element	Name	Class	Explanation / Specification
[[1]]	indicators	data.frame	 $id - integer $name - character $keyElement - logical - key indicator or not $functionalGroup - character $functionalGroupId - integer $scalingModel - character $scalingModelId - integer in addition one or more variables with fidelities to major ecosystems
[[2]]	ICunits	integer matrix	BSunitArea x indicators matrix of ICunits. Each element contains the ID of an indicator's ICunit in a BSunit.
[[3]]	BSunits	data.frame	 $id - integer $name - character $area - numeric - area data used in the calculation of weights additional variables describing BSunits, not required.
[[4]]	referenceValues	data.frame	 $ICunitId - integer $ICunitName - character $indId - integer

Element	Name	Class	Explanation / Specification
			$indName - character $yearId - integer - id for reference values $yearName - character - i.e. “Referenseverdi” $expectedValue - numeric $lowerQuantile - numeric $upperQuantile - numeric $customDistributionUUID - character - name of corresponding distribution object $distributionFamilyId - integer $distributionFamilyName - character $distParameter1 - numeric $distParameter2 - numeric $customDistribution
[[5]]	indicatorValues	list	Each element is a data.frame with the same variables as [[referenceValues]] and in addition scalingModel and scalingModelId. Each element corresponds to one year in a time series.
[[6]]	NIunits	integer matrix	0/1 BSunit x NIunit matrix delineating NIunits in terms of Bsunits. 1 means that the NIunit includes the BSunit.

Imputations

Imputations for missing indicator observations are relevant when e.g. calculating a time series of indices. `ImputeData` calculates multiple imputations which fit the requirements of the Nature Index framework. Imputations are stored as class `niImputations` lists.

Class `niImputations` lists

Lists of class `niImputations` are `S3` objects.

Each object of class `niImputations` is associated with a list of class `niInput`. The `niInput` object contains data for calculating the Nature Index or a thematic index, while the `niImputations` list contains imputed indicator observations for those missing in the dataset. Both objects must be given as arguments to `calculateIndex` whenever imputations are intended to complement the dataset in the calculations of the index.

`niImputations` lists are structured into the following elements:

Element	Name	Class	Explanation / Specification
[[1]]	identifiers	data.frame	Variables relating the imputed indicator observation to a missing observation in the dataset. $ICunitId - integer $indName - character $year - $refss - character $stringsAsFactors - integer
[[2]]	imputations	numeric matrix	Each row represents a missing indicator observation in the corresponding data set and contains single draws from each of nsim imputed distributions.

Results

`calculateIndex` calculates indices from `niInput` and `niImputations` lists and produces an extensive output for each index value to facilitate further analyses of the results.

Class `niOutput` lists

Lists of class `niOutput` are `S3` objects. Each element is a list of class `niSeries`.

The function `calculateIndex` returns an object of class `niOutput`. It contains the output from the calculation of the Nature Index or a thematic index for a set of NIunits.

Class `niSeries` lists

Lists of class `niSeries` are `S3` objects. Each element is a list of class `niValue`.

`niSeries` lists are elements in `niOutput` objects. Each contain the results from the calculation of a (time) series of index values for a single NIunit.

Class `niValue` lists

Lists of class `niValue` are `S3` objects containing the results for one index value calculated for a particular year and NIunit. `niValue` lists contain the following elements:

Element	Name	Class	Explanation
[[1]]	indexArea	character string	name of NIunit
[[2]]	call	object of mode "call"	unevaluated function call to `calculateIndex`
[[3]]	calculationParameters	list	options chosen for the calculation.
[[4]]	metadata	named numeric vector	`names(metadata) = c("nIndicators", "nBSunits", "nICunits", "nImputations")`, metadata describing the input dataset
[[5]]	year	integer scalar	
[[6]]	indicators	character vector	Indicator names
[[7]]	indicatorData	data.frame	Input indicator data
[[8]]	ICunits	integer vector	ICunit IDs
[[9]]	ICunitMatrix	named integer matrix	BSunit x indicator matrix giving the delineation of each ICunit in terms of BSunits.
[[10]]	imputations	data.frame	lists indicators and ICunits with imputed values, or `NULL`
[[11]]	BSunits	character vector	BSunit names
[[12]]	BSunitData	data.frame	Input BSunit data
[[13]]	BSunitWeights	named numeric matrix	BSunit x indicator matrix of BSunit weights
[[14]]	NIunitWeights	named numeric vector	NIunit weights for each BSunit
[[15]]	BSunitIndices	named numeric matrix	BSunit x nsim matrix of nsim index values per BSunit
[[16]]	BSunitbbb	named numeric vector	bbb statistic for each BSunit. Used to calculate location displacement (cf. Pedersen & Skarpaas 2012)
[[17]]	indexWeights	named numeric matrix	BSunit x indicator matrix of (NI)weights
[[18]]	index	numeric vector	nsim index values
[[19]]	bbb	numeric scalar	bbb statistic

The list of calculation options contains the following elements:

Element	Name	Mode	Explanation
[[1]]	fids	logical	Are weights based on indicator fidelities?
[[2]]	tgroups	logical	Are weights based on a grouping of indicators into trophic and key indicator groups?
[[3]]	keys	character	One of `c("none","asGroup", "ignore","specialWeight")`. `keys="none"` when `tgroups = FALSE`. `keys="asGroup"`: Key indicators are treated as an ordinary trophic group, no special weighting of key indicators. `keys="ignore"`: All indicators are treated as non-key indicators and according to their trophicGroup. `keys="specialWeight"`: special weighting of key indicators.
[[4]]	www	numerical	trophic weight given to key indicators when `keys="specialWeight"`.
[[5]]	awbs	logical	Are weights based on a BSunit variable (`awbs = TRUE`), or are all BSunits within the NIunit given equal weight (`awbs = FALSE`)?
[[6]]	stochastic	character	One of `c("both","observations", "reference","none")`. Are uncertainty in indicator observations and/or reference values accounted for in the calculations?
[[7]]	truncAtRef	logical	Are scaled indicator observations truncated at the reference value or not?
[[8]]	imputations	logical	Whether imputations were present in the input dataset or not.
[[9]]	weights	character	Do weights given to indicator observations from the same ICunit vary among years in the time series of index values (`weights = "Recalculated each year"`, i.e. there as some missing values in the data set), or is the weighting of indicator observations identical among years (`weights = "Identical all years"`)?
[[10]]	nsim	numerical	number of bootstrap simulation (draws for the index' distribution)

Updating indicator observations

Class `indicatorData` lists

The functions `getIndicatorValues`, `setIndicatorValues`, and `writeIndicatorValues` may be used in scripts that update the NI database with new or revised indicator observations for the indicators they are responsible for. `getIndicatorValues` retrieves the current observations for a given indicator from the NI database as an S3 object of class `indicatorData`, `setIndicatorValues` updates `indicatorData` objects with new indicator observations, and `writeIndicatorValues` posts objects with updated values to the 'Verdier' table in the database via the Nature Index API.

`indicatorData` lists are structured into the following elements:

Element	Name	Class	Explanation / Specification
[[1]]	$indicatorValues	data.frame	$indicatorId – integer $indicatorName – character $areaId – integer $areaName – character $yearId - integer $yearName - character $verdi - numeric $nedre_Kvartil - numeric $ovre_Kvartil - numeric $datatypeId – integer $datatypeName – character $UnitOfMeasurement – character $customDistributionUUID - character - name of corresponding distribution object $distributionName - character $distributionId - integer $distParam1 - numeric $distParam2 - numeric
[[2]]	$customDistributions	list	list of named distribution objects

Methods

`NIcalc` contains the following methods

Method	Class
plot	niSeries
	niValue
summary	niSeries
	niOutput