

Functions for calculating the Nature Index

Note to participants in output 4 of PDP1 under BG03 - IBBIS

Bård Pedersen

Norwegian Institute for Nature Research, May 26, 2016

Contents

Introduction: Mathematical framework	2
Definition	2
Spatial units	2
Indicator observations	2
Scaling	3
Weights	3
Calculating the Nature Index	4
Functions	5
Input data	6
Indicator observations and reference values	6
Indicators	6
Basic spatial units and NI-areas	7
To relate indicator areas to indicators	7
To describe indicator areas in terms of basic spatial units	7
Test data set	8
Required libraries	10
Set of functions	10
elicitate.fct	10
sampleobs.fct	13
scaleobs.fct	15
areaweights.fct	17
munweights.fct	19
NIcalculation.fct	21
summary.NIresults	22
More examples of calculations based on the test data set	24
Calculate NI for basic spatial units for all four years.	24
Calculate NI with uncertain reference values	25
Estimate the effect of uncertainty in reference values	25
Calculate NI without uncertainties	26

This document explains how to calculate the Nature index (NI) using the functions found in the r-script file *Simplified functions for calculating the Nature Index.R*. The document begins with describing the calculation of the index before explaining the structure of objects containing the necessary input data. Thereafter each function is given a short description. Finally, the calculation of NI is described and illustrated with examples.

Introduction: Mathematical framework

Definition

The nature index is a weighted average of scaled indicator observations

$$NI_{jkt} = \sum_{i=1}^n w_{ijk} S_{ikt} \quad (1)$$

where S_{ikt} , $i = 1, \dots, n$ is the n scaled indicator states included in the calculation. The weights (w_{ijk}) must conform to $\sum_{i=1}^n w_{ijk} = 1$.

Spatial units

The Nature Index is calculated for a specified major habitat (j) in a defined area (k , here referred to as an NI-area) and for a particular year (t) from a set of indicator-observations.

Indicator observations are however collected from indicator-specific spatial units (here referred to as indicator-areas) that may have a different spatial extent than the NI-area. The spatial delineation of indicator areas may also vary among indicators.

That is, for each indicator there is a set of non-overlapping spatial units from which indicator observations are collected. However, each indicator-area and NI-area must consist of one or more basic spatial units. In the Norwegian implementation of the Nature index this basic spatial unit is the municipality.

Thus, in order to calculate the index, the delineation of each indicator-area and each NI-area must be provided. That is, each indicator- and NI-area must be specified in terms of the basic spatial units they consist of. Further, all indicator observations must be linked to the correct indicator area. See section Input data below.

Indicator observations

Indicator observations may be associated with errors and are therefore uncertain. Thus, in the context of the Nature Index, the indicator observations are specified as probability distributions where the dispersion of the distribution represents this uncertainty, and the central tendency of the distribution represents the size of the indicator value. Thus, each observation is regarded as a stochastic variable. This presupposes that all input values used to calculate the index are provided with an estimate of how uncertain the values are. This estimate is provided by each expert in terms of the interquartile distance in the distribution, while the central tendency is given as an expected value.

Elicitation is the term for the process that generates probability distributions for uncertain quantities based on experts' knowledge and beliefs about them. In the context of the Nature Index, this is a two-step process. For all indicator values, experts estimate the distribution's expected value and its lower and upper quartiles. The R-scripts for calculating NI presupposes that these three parameters are provided for all indicator observations included in the calculation of the nature index.

During the calculation of the index, a probability distribution is fitted from a number of model distributions for each indicator value. The fitting is based on a least squares criterion. All the indicators are supposed to be nonnegative variables with zero as the minimum value. Hence, the model distributions are non-negative, univariate distributions. These are truncated normal-, lognormal-, Weibull-, "zeroinflated" exponential -, and gamma distributions. The truncated normal distribution is left-truncated at zero. All model distributions thus have two parameters.

Scaling

Scaling the indicator values to a common scale is done using nonlinear scaling functions (Figure 5.1). The scaling functions contain only one parameter, the so-called reference value (Ref). Reference values are specified for each indicator area in which an indicator is observed. The scaling functions' values are over the interval $[0,1]$.

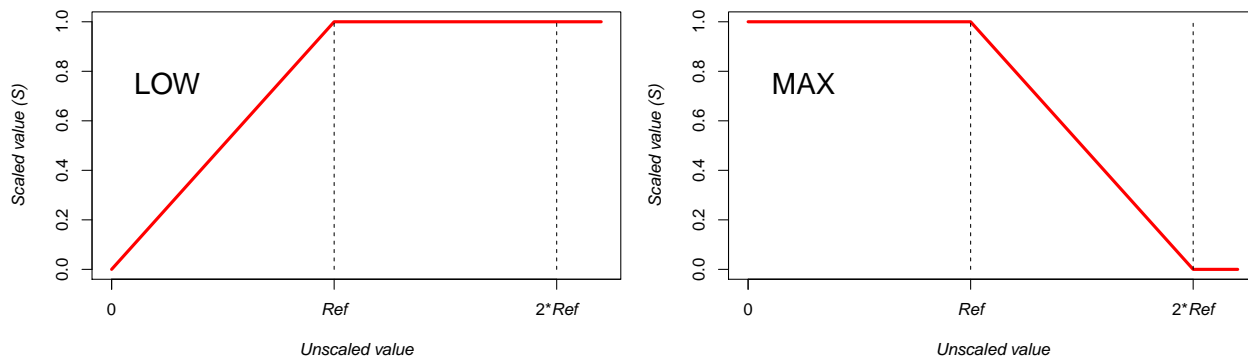


Figure 1: The scaling models *LOW* and *MAX*

There are two types of scaling functions, *LOW* and *MAX* (Figure 1). The choice of scaling function is determined by whether the indicator correlates positively or negatively with the aspect of biodiversity that the indicator represents. The *LOW* model is used when there is a positive correlation between the indicator and biodiversity. This holds true for most indicators. The *MAX* model is used when there is a negative correlation between the indicator and biodiversity. This applies only to certain indirect indicators that represent a negative effect that the measured indicator has on other components of biodiversity.

Weights

The nature index is a weighted average. In a weighted average, the state values used to calculate the average are weighted in relation to each other. Some indicator observations are given more weight than others. There are two motives for doing this in the context of the Nature Index. One motive is to correct for biases in the data from which the index is calculated. The second motive relates to the meaning of the index.

The current R-scripts calculate weights according to the Norwegian implementation of the Nature Index. However, the weighting system might be modified to better accord with the Bulgarian dataset and the purposes of the Bulgarian implementation. The Norwegian weighting system is described below.

The weights assigned to individual indicator values for the calculation of the Norwegian Nature Index for a basic spatial unit (i.e. a municipality) take into account the indicators' specificity / fidelity to the respective major ecosystem and the indicators' ecological function. The weights can be written as a product of two factors, a trophic weight ($w^{trophic}$) and a fidelity weight ($w^{fidelity}$).

The indicators' **fidelities** (φ_{ij}) to the various major habitats are determined by the individual expert based on the degree to which the underlying data for the indicator reflect the state of one or more major ecosystems. An indicator's total fidelity is 100%, i.e. $\sum_{j=1}^p \varphi_{ij} = 100\%$. Many indicators belong to only one of the major ecosystems; for example, mussels are only an indicator for the coast bottom major ecosystem and have a fidelity of 100% to the coast bottom, with a fidelity of 0% to the other major ecosystems. Conversely, marine fish species will often belong to two or more of the marine major ecosystems, ocean pelagic and ocean bottom, coast pelagic and coast bottom. In these situations (unless ecosystem-specific data are available), the fidelity is divided over several major ecosystems based on the extent to which the data for the indicators are affected by conditions in the different systems. The effect of this is that indicators that belong to several major ecosystems will be included in the calculation of the Nature Index for all of these systems. But their weight is then reduced in each of these calculations compared with indicators that have a 100% fidelity to one system.

In some cases, ecosystem-specific data is available for a species, and then separate indicators with 100% fidelity can be used. The **fidelity weight** of an indicator belonging to trophic group g is given by

$$w_{ijk}^{fidelity} = \frac{\varphi_{ij}}{\sum_{m \in g} \varphi_{mj}} \quad (2)$$

where the sum in the denominator is over all indicators within the trophic group observed in the geographical unit. The same holds for key indicators that are treated as a separate group in this context.

Trophic weight, ($w^{trophic}$), is equal to $\frac{1}{2}$ for the key indicators and equal to $\frac{1}{2r_{jk}}$ for other indicators, when observations of key indicators from municipality k is included in the calculation. When key indicators are lacking in k , the trophic weight for other indicators are $\frac{1}{r_{jk}}$. r_{jk} is the number of functional groups represented by indicator observations in the municipality k .

The product of an indicator observation's fidelity weight and trophic weight is the **total weight** given to the observation in a calculation of the nature index for a municipality k ,

$$w_{ijk}^p = w_{ijk}^{fidelity} * w_{ijk}^{trophic} \quad (3)$$

and the nature index for major habitat j in the municipality at time t is

$$NI_{jkt} = \sum_{i=1}^n w_{ijk}^p S_{ikt} \quad (4)$$

where the sum is over all indicators recorded in the municipality with $\varphi_{ij} > 0$ and $\sum_{i=1}^n w_{ijk}^p = 1$.

The consequences of these allocations are first, that the sum of the weights w_{ijk}^p for key indicators is $\frac{1}{2}$. So key indicators weigh 50% in the Nature Index for a municipality where observations of key indicators have been included in the calculation. Second, the different trophic groups represented in the municipality are weighted equally, regardless of the number of indicators represented within each group.

Municipalities are the basic spatial units for terrestrial major habitats and freshwater in the Norwegian implementation the Nature Index. However, in practice, aggregated indices for larger NI-areas (K) are calculated as weighted averages of municipal index values, e.g. for country parts or the whole country:

$$NI_{jKt} = \sum_{k \in K} w_{jk}^{area} NI_{jkt} = \sum_{k \in K} \sum_{i=1}^n w_{jk}^{area} w_{ijk}^p S_{ikt} \quad (5)$$

where w_{jk}^{area} are the **area weights**, $w_{jk}^{area} = \frac{a_{jk}}{\sum_{q \in K} a_{jq}}$, a_{jk} is the area that major ecosystem j covers in

municipality k , and $\sum_{k \in K} w_{jk}^{area} = 1$.

Calculating the Nature Index

Since the Nature Index is calculated from indicator observations that are considered stochastic variables, the index itself is also a stochastic variable with an associated probability distribution. Parametric bootstrapping is used to simulate this distribution from the distributions of indicator values. An observation from the index's distribution is generated by randomly drawing one observation from each of the distributions of indicator observations used in the calculation. The Nature Index value is then calculated based on this. The

index's distribution is simulated by repeating this procedure 1000 times, for example. Normally, the median of the simulated distribution is given as a point estimate for the Nature Index, while the 95% confidence interval given by the 2.5% and 97.5% quantiles of the distribution is used as a measure of uncertainty for the index value.

There are different opinions on whether uncertainties in reference values should be accounted for in the calculation of the nature index. On one hand reference values may be considered as fixed parameters that have been decided on the basis of the ecological framework described in Pedersen *et al.* (2016)¹ rather than estimated from the framework. On the other hand, reference values could be viewed as uncertain estimates derived from or modelled on the basis of the framework. In the Norwegian implementation uncertainties in reference values are not accounted for in the calculations. This is because uncertainties in many reference values have not been estimated.

Functions

Simplified functions for calculating the Nature Index. *R* contains the following functions:

Function name	Operation
elicitate.fct	Fits probability distributions to indicator observations
sampleobs.fct	Samples from the indicator observation distributions
scaleobs.fct	Scales samples from indicator observations according to chosen scaling model
munweights.fct	Calculates the product: trophic weights * fidelity weights
areaweights.fct	Calculates area weights
NIcalculation.fct	Calculates a sample of the Nature Index from samples of scaled indicator observations and area-, trophic-, and fidelity weights
summary.NIresults	Provides point estimates of the Nature Index together with 95% confidence intervals

Several of these functions call other functions that are also included in the file.

¹Pedersen, B., Nybø, S., Sæther, S. A. (eds.) 2016. Nature Index for Norway 2015. Ecological framework, computational methods, database and information systems – NINA Report 1226. 84 pp.

Input data

In order to use the above functions to calculate NI for a set of $N.reg$ NI-areas, from a set of in total $N.obs$ indicator observations and reference values, over $N.years$ years, for $N.ind$ indicators from altogether $N.omr = N.obs / (N.years + 1)$ indicator areas involving $N.kom$ basic spatial units, the following input data should be provided:

1. A data frame with indicator observations **together** with reference values
2. A data frame with indicator data
3. A data frame with basic spatial unit data
4. A data frame relating indicator areas to indicators
5. A data frame describing indicator areas in terms of basic spatial units

Indicator observations and reference values

The data frame should contain at least the following variables (variable names may be different)

Variable	Type	Explanation
ValueID	integer	vector of indicator observation- and reference value IDs
Expected.value	double	vector of the indicator observations' and reference values' expected values
Lower	double	vector of the observations' 0.25 (lower) quartiles
Upper	double	vector of the observations' 0.75 (upper) quartiles
IndicatorID	integer	vector of corresponding indicator IDs
IndicatorareaID	integer	vector of corresponding indicator area IDs
ReferenceYearID	integer	vector of corresponding IDs for observation years / reference values

Thus each indicator observation consists of the expected value together with the upper and lower quartiles. Each observation should have a unique ID and be linked to the correct indicator and indicator area through their IDs.

The last variable should signal whether the entry is a reference value or indicator observation. If it is an indicator observation the year or a yearID should be entered. If it is a reference value, an integer code, e.g. 0, should be entered.

Indicators

The data frame should contain at least the following variables (variable names may be changed)

Variable	Type	Explanation
IndicatorID	integer	vector of indicator IDs
Indicator_name	character	vector of the names of indicators
Scalingmodel	integer	vector of scaling model IDs, 1 = LOW and 2 = MAX
TrophicgroupID	integer	vector of trophic group IDs that the corresponding indicator belongs to
Key.indicators	logical	vector indicating whether corresponding indicator is a key indicator or not
Fidelity	double	vector of indicator fidelities to the major habitat in question

The data frame should contain complete data for all indicators included in the calculation. The code for scaling models should be 1 for LOW and 2 for MAX. Key indicators should be signalled with TRUE, non-key indicators with FALSE. Fidelity should be given as a number between 0 and 1.

Basic spatial units and NI-areas

The data frame should contain at least the following variables (variable names may be changed)

Variable	Type	Explanation
Basicsunit	character	vector of names of basic spatial units
NIarea	character	vector of NI area names that the corresponding basic spatial units are part of
Area	double	vector containing the areas of basic spatial units

An NI-area consists of those basic spatial units that share the same name in the variable NIarea. Basic spatial units may be grouped into NI-areas in several different ways by adding new “NIarea”-variables to the data frame

To relate indicator areas to indicators

The data frame should contain the following variables (variable names may be changed)

Variable	Type	Explanation
IndicatorareaID	integer	vector of unique indicator area IDs
Indicator_name	character	vector of indicator names

All indicator areas that are defined for the same indicator should have identical entries in Indicator_name.

To describe indicator areas in terms of basic spatial units

The data frame should contain at least the following variables (variable names may be changed)

Variable	Type	Explanation
IndicatorareaID	integer	vector of indicator area IDs
Basicsunit	character	vector of names of basic spatial units

The ID of an indicator area consisting of n basic spatial units should be entered n times in IndicatorareaID, while the names of the n basic spatial units should be entered in the corresponding entries of Basicsunit.

Test data set

An example data set is found in the package Nicalc:

```
require(Nicalc)

data(BSunits, Indic, Indicator.area, Indicator.area.ind, Observations)
ls()

## [1] "allowedIndicatorValues"      "Alt_DistID"
## [3] "Alt_DistIDyear2"            "areaweights1"
## [5] "areaweights2"                "bootmatall"
## [7] "bootmatyear2"                "BSunits"
## [9] "D"                            "discretePointEstimate"
## [11] "E"                            "EmpiricalPointEstimate"
## [13] "i"                            "Indic"
## [15] "Indicator.area"              "Indicator.area.ind"
## [17] "indicatorData"               "indicatorPosterior"
## [19] "indicatorValueProbabilities" "iresult"
## [21] "munweights"                  "munweightsforall"
## [23] "myIndicators"                "myProbs"
## [25] "N"                            "newValues"
## [27] "NIi"                          "NImethod"
## [29] "NIresult"                    "NIresult1"
## [31] "niToken"                     "nMean"
## [33] "nSamples"                    "nSd"
## [35] "Observations"                "Observations2"
## [37] "Observationsyear2"           "Observationsyear2only"
## [39] "P"                            "password"
## [41] "sampleMethod"                "scaled.bootmatall"
## [43] "scaled.bootmatyear2"          "scaled.bootmatyear2only"
## [45] "scaled.bootmatyeari"          "updatedIndicatorData"
## [47] "username"                     "year"
## [49] "years"
```

The test data consist of 236 observations for $N.ind = 33$ indicators in $N.omr = 59$ indicator areas over $N.years = 4$ years, together with reference values for each of the indicator areas. So $N.obs = 295$. Years are coded 2, 3, 4, 8, and reference values 0. The following table contains the first and last six entries of the data.frame Observations:

```
print(xtable(rbind(head(Observations), tail(Observations))))
```

	ValueID	IndicatorID	Expected.value	Lower	Upper	ReferenceYearID	IndicatorareaID
1	96	101	20.00	13.00	27.00	2	1523
2	97	101	20.00	10.00	30.00	2	1524
3	99	101	10.00	3.00	16.00	3	1523
4	101	101	10.00	3.00	16.00	4	1523
5	103	101	10.00	5.00	15.00	3	1524
6	104	101	10.00	5.00	15.00	4	1524
290	39416	379	4.00	4.00	4.00	0	5973
291	39402	364	4.00	3.50	4.50	2	6872
292	39403	364	4.00	3.50	4.50	3	6872
293	39404	364	4.00	3.50	4.50	4	6872
294	54616	364	4.00	3.50	4.50	8	6872
295	39400	364	4.00	4.00	4.00	0	6872

All indicators are scaled with the LOW model. They are all indicators for the same major habitat (which is unspecified), but with varying fidelity. Non-key indicators are grouped in 5 functional groups coded 1, 2, 5, 8, 9.

```
print(xtable(rbind(head(Indic), tail(Indic))))
```

	IndicatorID	Indicator_name	TrophicgroupID	Scalingmodel	Key.indicators	Fidelity
1	16	Ind 1	8	1	FALSE	0.25
2	17	Ind 2	9	1	FALSE	0.80
3	27	Ind 3	9	1	FALSE	0.70
4	83	Ind 4	1	1	FALSE	1.00
5	84	Ind 5	8	1	FALSE	1.00
6	92	Ind 6	9	1	FALSE	1.00
28	361	Ind 28	8	1	TRUE	1.00
29	364	Ind 29	5	1	TRUE	1.00
30	365	Ind 30	5	1	FALSE	0.25
31	371	Ind 31	5	1	FALSE	0.80
32	378	Ind 32	8	1	TRUE	1.00
33	379	Ind 33	1	1	FALSE	1.00

There are only $N.kom = 4$ basic spatial units, “Mun1” to “Mun4”, and most, but not all indicator areas consist of a single basic spatial unit. There are 2 different predefined sets of NI-areas with $N.reg = 1, 2$ respectively, in addition to the set of basic spatial units.

```
print(xtable(BSunits))
```

	BasicunitID	Basicunit	Area	NIarea1	NIarea2
1	2301	Mun 1	249618.00	All area	Part 1
2	2302	Mun 2	488272.00	All area	Part 2
3	2303	Mun 3	109780.00	All area	Part 1
4	2304	Mun 4	21819.00	All area	Part 2

```
print(xtable(rbind(head(Indicator.area.ind), tail(Indicator.area.ind))))
```

	IndicatorareaID	IndicatorID	Indicator_name
1	1456	16	Ind 1
2	1457	16	Ind 1
3	1463	92	Ind 6
4	1470	114	Ind 9
5	1473	147	Ind 11
6	1481	203	Ind 15
54	5969	365	Ind 30
55	5971	371	Ind 31
56	5973	379	Ind 33
57	6683	160	Ind 13
58	6723	83	Ind 4
59	6872	364	Ind 29

```
print(xtable(rbind(head(Indicator.area), tail(Indicator.area))))
```

	IndicatorareaID	BasicsunitID	Basicunit
1	1456	2302	Mun 2
2	1457	2301	Mun 1
3	1463	2302	Mun 2
4	1470	2303	Mun 3
5	1473	2301	Mun 1
6	1481	2304	Mun 4
62	5971	2301	Mun 1
63	5973	2301	Mun 1
64	6683	2303	Mun 3
65	6723	2303	Mun 3
66	6723	2304	Mun 4
67	6872	2301	Mun 1

Required libraries

The functions `elicitate.fct()` and `sampleobs.fct()` requires the R-libraries

```
library(MASS)
library(gamlss.dist)
library(truncnorm)
require(msm)
```

Set of functions

The set of functions are incorporated in the Nlcalc package.

elicitate.fct

`elicitate.fct` fits probability distributions to a set of indicator observations by selecting for each observation the distribution among a predetermined set of model distribution families that best fits the observation. The function presumes that indicator observations are each given as three parameters of a continuous probability distribution: expected value and the lower- and upper quartiles. The predetermined set of continuous distributions is {Truncated Normal, Lognormal, Weibull, Gamma, zero-inflated exponential}

Input arguments:

Name	Object type	Mode	Dim	Explanation
Expected.value	vector	double	$N.obs$	indicator observations' expected values
Lower	vector	double	$N.obs$	indicator observations' 0.25-quantiles (lower quartiles)
Upper	vector	double	$N.obs$	indicator observations' 0.75-quantiles (upper quartiles)

Expected.value, Lower and Upper should be of equal length.

For indicator observations and reference values with no uncertainty, Expected.value, Lower, and Upper should be equal.

Expected.value, Lower and Upper should be nonnegative. Negative values in an element of any of the these vectors are interpreted as missing value and the corresponding output elements are set to NA. Also, if one of Expected.value, Lower and Upper is NA, all corresponding output elements are set to NA.

Output:

Name	Object type	Mode	Dim	Explanation
elicitation.results	data.frame		$N.obs \times 4$	consisting of the following vectors
[[1]] \$FK_DistID	vector	character	$N.obs$	names of fitted model distributions
[[2]] \$mu	vector	double	$N.obs$	first parameter of fitted model distributions
[[3]] \$sig	vector	double	$N.obs$	second parameter of fitted model distributions
[[4]] \$ssq	vector	double	$N.obs$	sum of squared deviations between observed parameters and those of the fitted model distributions.

Functions called:

estim.fct() local function in present collection

estimlight.fct() local function in present collection

elicitation.fct first try to fit a model distribution with a call to the function estim.fct. If this results in an error, it calls estimlight.fct instead. These functions again each calls a series of other local functions included in *Simplified functions for calculating the Nature Index.R*, which again make use of routines in the MASS, gamlss.dist and truncnorm packages. Descriptions of all local functions are found in the file.

Examples:

```
elicitate(expected.value = Observations$Expected.value[1:10], lower = Observations$Lower[1:10],
          upper = Observations$Upper[1:10])
```

```
##      FK_DistID      mu      sig      ssq
## 1  TruncNormal 19.290161 1.079512e+01 5.646243e-02
## 2  TruncNormal 12.904372 1.880387e+01 8.556086e-01
## 3  TruncNormal -48.054273 2.695735e+01 1.225590e+00
## 4  TruncNormal -48.054273 2.695735e+01 1.225590e+00
## 5  TruncNormal  6.452186 9.401937e+00 2.139022e-01
## 6  TruncNormal  6.452186 9.401937e+00 2.139022e-01
## 7      Weibull 100.000000 1.366650e+09 6.669306e-15
## 8  TruncNormal 99.719655 3.721850e+01 2.563655e-02
## 9   LogNormal  2.924586 3.172087e-01 1.906000e-02
## 10  LogNormal  2.924917 3.172494e-01 2.382124e-02
```

```
Observations2 <- data.frame(Observations, elicitate(expected.value = Observations$Expected.value,
            lower = Observations$Lower, upper = Observations$Upper)[, 1:3])
head(Observations2)
```

```
##   ValueID IndicatorID Expected.value Lower Upper ReferenceYearID
## 1     96         101           20    13    27             2
## 2     97         101           20    10    30             2
## 3     99         101           10     3    16             3
## 4    101         101           10     3    16             4
## 5    103         101           10     5    15             3
```

## 6	104	101	10	5	15	4
##	IndicatorareaID	FK_DistID		mu	sig	
## 1	1523	TruncNormal	19.290161	10.795121		
## 2	1524	TruncNormal	12.904372	18.803874		
## 3	1523	TruncNormal	-48.054273	26.957346		
## 4	1523	TruncNormal	-48.054273	26.957346		
## 5	1524	TruncNormal	6.452186	9.401937		
## 6	1524	TruncNormal	6.452186	9.401937		

sampleobs.fct

sampleobs.fct returns *nsim* random draws from a set of distributions given by the vector *DistID* and the parameter vectors *mu* and *sig*. The function presumes that the set only includes the truncated normal-, lognormal-, weibull-, gamma-, zero-inflated exponential, poisson-, negative binomial-, and zero-inflated poisson distributions. The lower bound in the truncated normal distribution is always zero, while the upper bound is infinity. With these restrictions all theoretical distributions have two parameters, except the poisson with only one parameter. Alternatively, if *distrib* == "NoBoot", the function returns *nsim* copies of the observation's expected value. The function does not handle missing values (NAs).

Input arguments:

Name	Object type	Mode	Dim	Explanation
ValueID	vector	integer	<i>N.obs</i>	indicator observation IDs
DistID	vector	character	<i>N.obs</i>	distribution families names or "NoBoot" (when copy obs)
mu	vector	double	<i>N.obs</i>	vector of parameter 1 of model distributions
sig	vector	double	<i>N.obs</i>	vector of parameter 2 of model distributions
Value	vector	double	<i>N.obs</i>	vector of observed expected value to be copied if <i>distrib</i> == "NoBoot".
nsim	vector	integer	1	number of draws (simulations)

Allowed names of distribution families are "TruncNormal", "LogNormal", "Weibull", "ZIExponential", "Gamma", "Poisson", "NegBinom", "ZIP", "NoBoot"

Output:

Name	Object type	Mode	Dim	Explanation
bootmat	matrix	double	<i>N.obs x nsim</i>	random draws from specified (by <i>DistID</i> , <i>mu</i> and <i>sig</i>) models or <i>nsim</i> copies of <i>Value</i>

Functions called:

rtnorm - random generation function in the truncnorm package.

rLOGNO, rWEI, rGA, rPO, rNBII, rZIP - random generation functions in the gamlss.dist package.

runif and rexp - random generation functions from the stats package.

Examples:

```
samplebootmat(ValueID = Observations2$ValueID, Value = Observations2$Expected.value,  
  RefobsID = Observations2$ReferenceYearID, DistID = Observations2$FK_DistID,  
  mu = Observations2$mu, sig = Observations2$sig, nsim = 5)[1:10, ]
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]  
## 96    17.298852   6.102929  27.2030560  29.163525  17.644701  
## 97    17.662833  10.524388  15.0873929  15.202633   5.628781
```

```
## 99      2.803823    1.658370  21.7653359    2.565324    1.043895
## 101     6.024229   13.018797   0.8521935    7.528379   25.770230
## 103    10.917745    3.143546   4.9941095    9.677082    7.883795
## 104    12.970503    9.845415    7.9643388    7.025224   10.403211
## 107   100.000000  100.000000  100.000000  100.000000  100.000000
## 111   100.000000  100.000000  100.000000  100.000000  100.000000
## 1343   24.267983   13.799615   28.1465715   26.641405   19.330092
## 1344   13.447322   31.471882   29.6276569   23.778309   17.478713
```

In order to base the calculation of NI on fixed reference values (i.e. to ignore uncertainties in them), the corresponding DistIDs should be set to “NoBoot”

```
Alt_DistID <- Observations2$FK_DistID
Alt_DistID[Observations2$ReferenceYearID == 0] <- "NoBoot"

samplebootmat(ValueID = Observations2$ValueID, Value = Observations2$Expected.value,
  RefobsID = Observations2$ReferenceYearID, DistID = Alt_DistID, mu = Observations2$mu,
  sig = Observations2$sig, nsim = 5)[1:10, ]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]
## 96    28.0315020  4.658286  28.896433  24.165389  35.0286816
## 97    31.6545815 14.166191  53.965598  18.015107  2.1132568
## 99     0.7056494  8.179786   4.796470   9.618021  0.4844678
## 101    0.8685109  5.441320   3.814947   6.287686  1.5062100
## 103    28.7237949 13.487508  12.350086   4.262983  15.5786931
## 104    11.2339953 11.002447  19.271724   1.707695   8.9253355
## 107   100.0000000 100.000000  100.000000  100.000000  100.0000000
## 111   100.0000000 100.000000  100.000000  100.000000  100.0000000
## 1343   18.0703489 20.626596  21.671846  29.843686  26.4648653
## 1344   32.0730425 14.682591  18.553632  26.141351  23.4498695
```

NI is calculated for one year at a time. To include observations from year 2 only:

```
Observationsyear2 <- Observations2[Observations2$ReferenceYearID %in% c(0, 2),
  ]
Alt_DistIDyear2 <- Alt_DistID[Observations2$ReferenceYearID %in% c(0, 2)]

bootmatyear2 <- samplebootmat(ValueID = Observationsyear2$ValueID, Value = Observationsyear2$Expected.v,
  RefobsID = Observationsyear2$ReferenceYearID, DistID = Alt_DistIDyear2,
  mu = Observationsyear2$mu, sig = Observationsyear2$sig, nsim = 1000)
bootmatyear2[1:10, 1:5]
```

```
##          [,1]      [,2]      [,3]      [,4]      [,5]
## 96    22.18963  28.64623  44.13059   0.6149252  21.929336
## 97    18.63570  25.21604  20.04155   3.7658315   9.964985
## 107   100.00000 100.00000  100.00000  100.0000000  100.0000000
## 111   100.00000 100.00000  100.00000  100.0000000  100.0000000
## 1345   40.00000  40.00000  40.00000  40.0000000  40.0000000
## 1346   40.00000  40.00000  40.00000  40.0000000  40.0000000
## 1349   60.19665  43.70504  49.01369  34.9064081  58.022831
## 1350   54.36872  54.78913  55.97720  49.5573858  39.854217
## 1357   85.00000  85.00000  85.00000  85.0000000  85.0000000
## 1358   85.00000  85.00000  85.00000  85.0000000  85.0000000
```

scaleobs.fct

scaleobs.fct scales a set of indicator observations according to chosen scaling model (*LOW* or *MAX*) and associated reference values. Each indicator observation should be entered either as *nsim* random draws from distributions fitted to the original observations (that were provided as expected values and upper and lower quartiles), or as *nsim* repetitions of the original expected values provided by experts. In the same way, reference values should be entered either as *nsim* random draws from distributions fitted to the original reference values, or as *nsim* repetitions of the original expected values.

Reference values and indicator observations should be entered together in the same matrix *bootmat*. Further a vector with codes identifying reference values from indicator observations and of length equal to the number of indicator observations and reference values should be entered. Vectors of the same length containing the observations/reference values area IDs and indicator IDs should also be entered.

Scaling models are specific for each indicator. They should be entered as a vector of scaling model codes where 1 = *LOW* and 2 = *MAX*, together with a vector of associated indicator IDs of the same length.

Input arguments:

Name	Object type	Mode	Dim	Explanation
ValueID	vector	integer	$N.obs$	indicator observation IDs
FK_OmraadeID	vector	integer	$N.obs$	indicator area IDs
FK_IndicatorID	vector	integer	$N.obs$	indicator IDs
FK_RefAarID	vector	integer	$N.obs$	IDs for observation years / reference values
nsim	vector	integer	1	number of draws (simulations) in the estimation of NI
bootmat	matrix	double	$N.obs \times nsim$	random draws from fitted distributions and/or copies of expected values
ref.value.code	vector	integer	1	code for reference values used in FK_RefAarID, default = 0
IndicatorID	vector	integer	$N.ind$	unique indicator IDs
FK_Scalingmodel	vector	integer	$N.ind$	scaling model IDs, 1 = <i>LOW</i> and 2 = <i>MAX</i>

Output:

Name	Object type	Mode	Dim	Explanation
scaled.bootmat	matrix	double	$N.obs \times nsim$	scaled draws from fitted distributions and/or scaled copies of expected values

Note that scaleobs.fct scaled.bootmat is of the same dimension as bootmat. This means that scaled.bootmat contains *nsim* draws of scaled reference values which all are equal to 1. These must be removed from scaled.bootmat before using Nicalculation.fct to calculate random draws of the nature index.

Further, if the input bootmat matrix contains indicator observations for several years, so will the scaled.bootmat. However, the index is calculated for one year, so the output matrix scaled.bootmat must be split accordingly into a set of matrices, where each matrix contains draws of scaled indicator observations for one single year only.

Example:

```
scaled.bootmatyear2 <- scaleobs(ValueID = Observationsyear2$ValueID, FK_OmraadeID = Observationsyear2$FK_OmraadeID,
  FK_IndicatorID = Observationsyear2$IndicatorID, FK_RefAarID = Observationsyear2$ReferenceYearID,
  nsim = 1000, bootmat = bootmatyear2, ref.value.code = 0, IndicatorID = Indic$IndicatorID,
  FK_Scalingmodel = Indic$Scalingmodel)
scaled.bootmatyear2[1:9, 1:5]
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## 96  0.2218963 0.2864623 0.4413059 0.006149252 0.21929336
## 97  0.1863570 0.2521604 0.2004155 0.037658315 0.09964985
## 107 1.0000000 1.0000000 1.0000000 1.000000000 1.00000000
## 111 1.0000000 1.0000000 1.0000000 1.000000000 1.00000000
## 1345 1.0000000 1.0000000 1.0000000 1.000000000 1.00000000
## 1346 1.0000000 1.0000000 1.0000000 1.000000000 1.00000000
## 1349 1.0000000 1.0000000 1.0000000 0.872660203 1.00000000
## 1350 1.0000000 1.0000000 1.0000000 1.000000000 0.99635542
## 1357 1.0000000 1.0000000 1.0000000 1.000000000 1.00000000
```

To remove reference values from the output matrix scaled.bootmatyear2

```
scaled.bootmatyear2only <- scaled.bootmatyear2[Observationsyear2$ReferenceYearID !=
  0, ]
scaled.bootmatyear2only[1:9, 1:5]
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## 96  0.2218963 0.2864623 0.4413059 0.006149252 0.21929336
## 97  0.1863570 0.2521604 0.2004155 0.037658315 0.09964985
## 1349 1.0000000 1.0000000 1.0000000 0.872660203 1.00000000
## 1350 1.0000000 1.0000000 1.0000000 1.000000000 0.99635542
## 1863 0.2744396 0.0966586 0.8026740 0.608327230 0.96949896
## 10246 0.6610960 0.6678266 0.6086991 0.737374892 0.74097995
## 10264 0.6544057 0.5230710 0.6461385 0.538771101 0.70660885
## 10265 0.5175575 0.5701991 0.6309908 0.675296657 0.67063929
## 11349 1.0000000 0.9015318 1.0000000 1.000000000 1.00000000
```

Now scaled.bootmatyear2only includes one observation per indicator area.

```
dim(scaled.bootmatyear2only)
```

```
## [1] 59 1000
```

However, to keep track of which observation in scaled.bootmatyear2 corresponds to the various indicator areas we may want to create a new data frame with indicator observations corresponding to scaled.bootmatyear2

```
Observationsyear2only <- Observationsyear2[Observationsyear2$ReferenceYearID !=
  0, ]
dim(Observationsyear2only)
```

```
## [1] 59 10
```


areaweights.fct

areaweights.fct calculates relevant area weights w_{jk}^{area} for chosen spatial resolution of the NI output. I.e. areaweights.fct calculates w_{jk}^{area} for a set of aggregated, nonoverlapping, spatial units each consisting of one to many basic spatial units (municipalities). In the Norwegian implementation w^{area} are weights based on the area the major habitat in question covers in each basic spatial unit. An alternative is to use the total area of the spatial unit, etc.

w^{area} is the relevant municipality area divided by the corresponding total area of the aggregate. Municipalities outside the aggregate receive weights equal zero

Area weights are relevant when NI is calculated for aggregates of basic spatial units.

Note that the weights are stored in a three-dimensional array to be included in the calculation of the nature index in a subsequent call to Nicalculation.fct. In the array weights are repeated for each indicator included in the current calculation of NI. This in order to simplify the coding of Nicalculation.fct.

Input arguments:

Name	Object type	Mode	Dim	Explanation
Municipalities	vector	character	$N.kom$	names of all basic spatial units included in the calculation
Regions	vector	character	$N.kom$	names of aggregated spatial units that the corresponding municipality is a part of.
Area.municipality	vector	double	$N.kom$	relevant basic spatial unit areas
Indicators	vector	character	$N.ind$	names of all indicators included in the calculation

Output:

Name	Object type	Mode	Dim	Explanation
Weights.reg.area	array	double	$N.kom \times N.ind$ $\times N.reg$	$N.ind$ copies of area weights for each NI area.

Examples:

To calculate area weights for “All area”

```
areaweights1 <- areaweights(Municipalities = BSunits$Basicunit, Regions = BSunits$NIarea1,  
  Area.municipality = BSunits$Area, Indicators = Indic$Indicator_name)  
areaweights1[, 1:6, ]
```

```
##          Ind 1      Ind 2      Ind 3      Ind 4      Ind 5      Ind 6  
## Mun 1 0.28708586 0.28708586 0.28708586 0.28708586 0.28708586 0.28708586  
## Mun 2 0.56156202 0.56156202 0.56156202 0.56156202 0.56156202 0.56156202  
## Mun 3 0.12625807 0.12625807 0.12625807 0.12625807 0.12625807 0.12625807  
## Mun 4 0.02509405 0.02509405 0.02509405 0.02509405 0.02509405 0.02509405
```

To calculate area weights for “Part 1” and “Part2”

```
areaweights2 <- areaweights(Municipalities = BSunits$Basicunit, Regions = BSunits$NIarea2,
  Area.municipality = BSunits$Area, Indicators = Indic$Indicator_name)
areaweights2[, 1:6, ]
```

```
## , , Part 1
##
##      Ind 1      Ind 2      Ind 3      Ind 4      Ind 5      Ind 6
## Mun 1 0.6945448 0.6945448 0.6945448 0.6945448 0.6945448 0.6945448
## Mun 2 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## Mun 3 0.3054552 0.3054552 0.3054552 0.3054552 0.3054552 0.3054552
## Mun 4 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##
## , , Part 2
##
##      Ind 1      Ind 2      Ind 3      Ind 4      Ind 5      Ind 6
## Mun 1 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## Mun 2 0.95722528 0.95722528 0.95722528 0.95722528 0.95722528 0.95722528
## Mun 3 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## Mun 4 0.04277472 0.04277472 0.04277472 0.04277472 0.04277472 0.04277472
```

To check if weights sum to 1

```
colSums(areaweights2)[1:6, ]
```

```
##      Part 1 Part 2
## Ind 1      1      1
## Ind 2      1      1
## Ind 3      1      1
## Ind 4      1      1
## Ind 5      1      1
## Ind 6      1      1
```

If you want to give equal weights to all basic spatial units in the calculation of NI, one could do as follows:

```
areaweights(Municipalities = BSunits$Basicunit, Regions = BSunits$NIarea2, Area.municipality = BSunits$Area,
  Indicators = Indic$Indicator_name)[, 1:6, ]
```

```
## , , Part 1
##
##      Ind 1 Ind 2 Ind 3 Ind 4 Ind 5 Ind 6
## Mun 1  0.5  0.5  0.5  0.5  0.5  0.5
## Mun 2  0.0  0.0  0.0  0.0  0.0  0.0
## Mun 3  0.5  0.5  0.5  0.5  0.5  0.5
## Mun 4  0.0  0.0  0.0  0.0  0.0  0.0
##
## , , Part 2
##
##      Ind 1 Ind 2 Ind 3 Ind 4 Ind 5 Ind 6
## Mun 1  0.0  0.0  0.0  0.0  0.0  0.0
## Mun 2  0.5  0.5  0.5  0.5  0.5  0.5
## Mun 3  0.0  0.0  0.0  0.0  0.0  0.0
## Mun 4  0.5  0.5  0.5  0.5  0.5  0.5
```

munweights.fct

munweights.fct calculates fidelity weights ($w^{fidelity}$), trophic weights ($w^{trophic}$) and their product w^p for each combination of municipality and indicator. $w^p = 0$ for indicators not observed in a municipality.

Input arguments:

Name	Object type	Mode	Dim	Explanation
Municipalities	vector	character	$N.kom$	names of all basic spatial units included in the calculation
Indicators	vector	character	$N.ind$	names of all indicators included in the calculation
FK_TrophicgroupID	vector	integer	$N.ind$	trophic group IDs that the corresponding indicator belongs to
Key.indicators	vector	logical	$N.ind$	key indicator or not
Fidelity	vector	double	$N.ind$	indicator fidelities to the major habitat in question
Areaind.Name	vector	integer	$N.omr$	unique indicator area IDs for all areas included in the calculation of NI.
Areaind.Indicator	vector	character	$N.omr$	indicator names corresponding to the entries in Areaind.Name
Area.Name	vector	integer	$\text{length}(\text{Area.Name})$	indicator area IDs, where each indicator area ID in Areaind.Name is repeated for each spatial unit it consists of
Area.Municipality	vector	character	$\text{length}(\text{Area.Name})$	names of basic spatial units corresponding to the entries in Area.Name

Output

Name	Object type	Mode	Dim	Explanation
Weights.trof	matrix	double	$N.kom \times N.ind$	w^p .

Examples:

```
munweights <- munweights(Municipalities = BSunits$Basicunit, Indicators = Indic$Indicator_name,
  FK_TrophicgroupID = Indic$TrophicgroupID, Key.indicators = Indic$Key.indicators,
  Fidelity = Indic$Fidelity, Areaind.Name = Indicator.area.ind$IndicatorareaID,
  Areaind.Indicator = Indicator.area.ind$Indicator_name, Area.Name = Indicator.area$IndicatorareaID,
  Area.Municipality = Indicator.area$Basicunit)
```

```
munweights[, 1:6]
```

```
##           Ind 1      Ind 2      Ind 3      Ind 4      Ind 5      Ind 6
## Mun 1 0.00500000 0.05333333 0.04666667 0.00000000 0.02000000 0.00000000
## Mun 2 0.01082251 0.05333333 0.04666667 0.00000000 0.04329004 0.06666667
## Mun 3 0.00000000 0.06666667 0.05833333 0.03676471 0.03472222 0.00000000
```

```
## Mun 4 0.00000000 0.00000000 0.12500000 0.03676471 0.03472222 0.00000000
```

```
rowSums(munweights)
```

```
## Mun 1 Mun 2 Mun 3 Mun 4
```

```
##      1      1      1      1
```

If you want to ignore trophic groups in the calculation of w^p , set all entries in FK_TrophicgroupID equal to each other. If you want to ignore the distinction between key- and non-key indicators, set Key.indicators to FALSE. If you want to ignore differences in fidelity, set all entries in Fidelity equal 1.0. If you want to give equal weights to all indicators observed in a community, enter all three of these settings for FK_TrophicgroupID, Key.indicators and Fidelity.

```
munweights(Municipalities = BSunits$Basicunit, Indicators = Indic$Indicator_name,
  FK_TrophicgroupID = rep(4, length(Indic$TrophicgroupID)), Key.indicators = rep(F,
    length(Indic$Key.indicators)), Fidelity = rep(1, length(Indic$Fidelity)),
  Areaind.Name = Indicator.area.ind$IndicatorareaID, Areaind.Indicator = Indicator.area.ind$Indicator,
  Area.Name = Indicator.area$IndicatorareaID, Area.Municipality = Indicator.area$Basicunit)[,
  1:6]
```

```
##      Ind 1      Ind 2      Ind 3      Ind 4      Ind 5      Ind 6
## Mun 1 0.03846154 0.03846154 0.03846154 0.00000000 0.03846154 0.00000000
## Mun 2 0.07142857 0.07142857 0.07142857 0.00000000 0.07142857 0.07142857
## Mun 3 0.00000000 0.07142857 0.07142857 0.07142857 0.07142857 0.00000000
## Mun 4 0.00000000 0.00000000 0.07692308 0.07692308 0.07692308 0.00000000
```

NIcalculation.fct

Function for calculating nsim draws of NI from nsim sets of (N.omr) scaled observations for (N.ind) indicators and corresponding areaweights, trophic weights and fidelity weights for a set of (N.kom) basic spatial units or a set of (N.reg) aggregated NI-areas each consisting of one or several basic spatial units.

Scaled indicator observations are calculated in scaleobs.fct, while area weights are calculated in areaweights.fct, and the product of trophic weights and fidelity weights in munweights.fct.

NI is calculated for a set of NI-areas, while scaled indicator observations, on the other hand, refer to indicator specific areas that may vary in geographical extent among indicators, and will in general not correspond to the NI-areas. Thus, NIcalculation.fct needs input about the spatial extent of each indicator area and NI-area.

Area weights should not be entered to NIcalculation.fct if NI is calculated for basic spatial units.

NIcalculate returns a matrix of nsim draws of NI for each NI-area (or basic spatial unit), of class NIresults.

Input arguments:

Name	Object type	Mode	Dim	Explanation
Areaind.Name	vector	integer	$N.omr$	unique indicator area IDs
Areaind.Indicator	vector	character	$N.omr$	indicator names corresponding to the entries in Areaind.Name
Area.Name	vector	integer	$\text{length}(\text{Area.Name})$	indicator area IDs, where each indicator area ID is repeated for each basic spatial unit it consists of
Area.Municipality	vector	character	$\text{length}(\text{Area.Name})$	vector of names of corresponding basic spatial units
FK_OmraadeID	vector	integer	$N.omr$	indicator area IDs corresponding to the sequence of indicator observations in scaled.bootmat
nsim	vector	integer	1	number of draws (simulations) in the estimation of NI
scaled.bootmat	matrix	double	$N.omr \times nsim$	nsim random draws of scaled indicator observations or nsim copies of scaled expected values
Weights.trof	matrix	double	$N.kom \times N.ind$	product of fidelity weights and trophic weights w^p
Weights.reg.area	array	double	$N.kom \times N.ind \times N.reg$	area weights w^{area}

Output:

Name	Object type	Mode	Dim	Explanation
NI	NIresult, matrix	double	$(N.kom \text{ or } N.reg) \times nsim$	matrix of NI draws

summary.NIresults

The summary function for NIresults provides point estimates of the Nature Index together with confidence intervals for a set of *N.reg* basic spatial units or aggregated NI-areas from a *N.regxnsim* matrix of *nsim* simulations of NI for each area.

As default parameters returns the median simulated value as the point estimate and the 0.025- and 0.975-quantiles as limits for the confidence interval.

Input arguments:

Name	Object type	Mode	Dim	Explanation
NI	NIsummary	double	$(N.kom \text{ or } N.reg) \times nsim$	NI draws
kvantiler	vector	double	3	Quantiles used as basis for point estimate and limits of confidence interval.

Output:

Name	Object type	Mode	Dim	Explanation
NIest	matrix	double	$(N.kom \text{ or } N.reg) \times 3$	Estimates of NI

Examples

To estimate the Nature Index for the area “All area”, enter the areaweights stored in areaweights1 which was calculated above.

```
summary(NIcalculate(Areaind.Name = Indicator.area.ind$IndicatorareaID, Areaind.Indicator = Indicator.ar
  Area.Name = Indicator.area$IndicatorareaID, Area.Municipality = Indicator.area$Basicunit,
  FK_OmraadeID = Observationsyear2$IndicatorareaID[Observationsyear2$ReferenceYearID !=
    0], nsim = 1000, scaled.bootmat = scaled.bootmatyear2[Observationsyear2$ReferenceYearID !=
    0, ], Weights.trof = munweights, Weights.reg.area = areaweights1))
```

```
##           0.025      0.5      0.975
## All area 0.5740089 0.6904668 0.7729667
## attr(,"class")
## [1] "NIsummary" "matrix"
```

To estimate the Nature Index for the NI-areas “Part1” and “Part2”, weights stored in areaweights2.

```
summary(NIcalculate(Areaind.Name = Indicator.area.ind$IndicatorareaID, Areaind.Indicator = Indicator.ar
  Area.Name = Indicator.area$IndicatorareaID, Area.Municipality = Indicator.area$Basicunit,
  FK_OmraadeID = Observationsyear2$IndicatorareaID[Observationsyear2$ReferenceYearID !=
    0], nsim = 1000, scaled.bootmat = scaled.bootmatyear2[Observationsyear2$ReferenceYearID !=
    0, ], Weights.trof = munweights, Weights.reg.area = areaweights2))
```

```
##           0.025      0.5      0.975
## Part 1 0.6045511 0.6961416 0.7724760
## Part 2 0.5024862 0.6916946 0.8008551
## attr(,"class")
```

```
## [1] "NIsuSummary" "matrix"
```

To estimate the Nature Index for basic spatial units, do not enter area weights.

```
summary(NIcalculate(Areaind.Name = Indicator.area.ind$IndicatorareaID, Areaind.Indicator = Indicator.ar
  Area.Name = Indicator.area$IndicatorareaID, Area.Municipality = Indicator.area$Basicunit,
  FK_OmraadeID = Observationsyear2$IndicatorareaID[Observationsyear2$ReferenceYearID !=
    0], nsim = 1000, scaled.bootmat = scaled.bootmatyear2[Observationsyear2$ReferenceYearID !=
    0, ], Weights.trof = munweights))
```

```
##           0.025           0.5           0.975
## Mun 1 0.6095452 0.7252579 0.8135143
## Mun 2 0.4992053 0.6953875 0.8080408
## Mun 3 0.5020530 0.6298139 0.7435781
## Mun 4 0.4877304 0.6080267 0.7162214
## attr("class")
## [1] "NIsuSummary" "matrix"
```

To change to eg. a 90% confidence interval, specify nondefault values for kvantiler.

```
summary(NIcalculate(Areaind.Name = Indicator.area.ind$IndicatorareaID, Areaind.Indicator = Indicator.ar
  Area.Name = Indicator.area$IndicatorareaID, Area.Municipality = Indicator.area$Basicunit,
  FK_OmraadeID = Observationsyear2$IndicatorareaID[Observationsyear2$ReferenceYearID !=
    0], nsim = 1000, scaled.bootmat = scaled.bootmatyear2[Observationsyear2$ReferenceYearID !=
    0, ], Weights.trof = munweights))
```

```
##           0.025           0.5           0.975
## Mun 1 0.6095452 0.7252579 0.8135143
## Mun 2 0.4992053 0.6953875 0.8080408
## Mun 3 0.5020530 0.6298139 0.7435781
## Mun 4 0.4877304 0.6080267 0.7162214
## attr("class")
## [1] "NIsuSummary" "matrix"
```

More examples of calculations based on the test data set

Calculate NI for basic spatial units for all four years.

In the test data set there are no missing indicator observations, i.e. indicator observations have been recorded in all indicator areas in each of the four years included. This means that weights used in the calculation of NI will be the same for all years.

In the example below, uncertainties in reference values are not accounted for.

```
Observations <- data.frame(Observations, elicitate(expected.value = Observations$Expected.value,
  lower = Observations$Lower, upper = Observations$Upper)[, 1:3])

years <- unique(Observations$ReferenceYearID[Observations$ReferenceYearID !=
  0])
Alt_DistID <- Observations$FK_DistID
Alt_DistID[Observations$ReferenceYearID == 0] <- "NoBoot"

bootmatall <- samplebootmat(ValueID = Observations$ValueID, Value = Observations$Expected.value,
  RefobsID = Observations$ReferenceYearID, DistID = Alt_DistID, mu = Observations$mu,
  sig = Observations$sig, nsim = 1000)

scaled.bootmatall <- scaleobs(ValueID = Observations$ValueID, FK_OmraadeID = Observations$IndicatorareaID,
  FK_IndicatorID = Observations$IndicatorID, FK_RefAarID = Observations$ReferenceYearID,
  nsim = 1000, bootmat = bootmatall, ref.value.code = 0, IndicatorID = Indic$IndicatorID,
  FK_Scalingmodel = Indic$Scalingmodel)

munweightsforall <- munweights(Municipalities = BSunits$Basicunit, Indicators = Indic$Indicator_name,
  FK_TrophicgroupID = Indic$TrophicgroupID, Key.indicators = Indic$Key.indicators,
  Fidelity = Indic$Fidelity, Areaind.Name = Indicator.area.ind$IndicatorareaID,
  Areaind.Indicator = Indicator.area.ind$Indicator_name, Area.Name = Indicator.area$IndicatorareaID,
  Area.Municipality = Indicator.area$Basicunit)

NIresult <- NULL

for (i in years) {

  scaled.bootmatyeari <- scaled.bootmatall[Observations$ReferenceYearID ==
    i, ]

  NIi <- summary(Nicalculate(Areaind.Name = Indicator.area.ind$IndicatorareaID,
    Areaind.Indicator = Indicator.area.ind$Indicator_name, Area.Name = Indicator.area$IndicatorareaID,
    Area.Municipality = Indicator.area$Basicunit, FK_OmraadeID = Observations$IndicatorareaID[Observations$ReferenceYearID == i], nsim = 1000, scaled.bootmat = scaled.bootmatyeari, Weights.trof = munweightsforall))

  year <- rep(i, dim(NIi)[1])
  iresult <- cbind(year, NIi)
  NIresult <- rbind(NIresult, iresult)

}

NIresult

##      year      0.025      0.5      0.975
## Mun 1      2 0.6090815 0.7254112 0.8145972
```



```
## Mun 2      2 0.4951396 0.6929235 0.8095687
## Mun 3      2 0.4986721 0.6286155 0.7408326
## Mun 4      2 0.4804884 0.6112681 0.7188868
## Mun 1      3 0.5861226 0.6966156 0.7936411
## Mun 2      3 0.4339003 0.6380504 0.7652685
## Mun 3      3 0.4167306 0.5516883 0.6748226
## Mun 4      3 0.4730459 0.5918962 0.6916610
## Mun 1      4 0.6095103 0.7051773 0.8132393
## Mun 2      4 0.5066938 0.6993006 0.8172746
## Mun 3      4 0.4622512 0.5768286 0.7090273
## Mun 4      4 0.4703545 0.5569165 0.6828366
## Mun 1      8 0.5646740 0.6480436 0.7335862
## Mun 2      8 0.4646412 0.6524077 0.7771531
## Mun 3      8 0.4226794 0.5333869 0.6134751
## Mun 4      8 0.4561206 0.5126608 0.5848853
```

Calculate NI with uncertain reference values

The same code as above, however without the assignment: `Alt_DistID[Observations$ReferenceYearID==0]`
`<- "NoBoot"`. The result is as follows:

```
##      year      0.025      0.5      0.975
## Mun 1      2 0.6121056 0.7250319 0.8162222
## Mun 2      2 0.5032822 0.6859948 0.8075258
## Mun 3      2 0.4978399 0.6269383 0.7413811
## Mun 4      2 0.4768842 0.6005820 0.7151140
## Mun 1      3 0.5858127 0.6973288 0.7896410
## Mun 2      3 0.4481523 0.6268951 0.7583476
## Mun 3      3 0.4134075 0.5536416 0.6751149
## Mun 4      3 0.4755392 0.6008897 0.6938488
## Mun 1      4 0.6067570 0.7049894 0.8195012
## Mun 2      4 0.4964149 0.7036561 0.8172108
## Mun 3      4 0.4537210 0.5747991 0.7121291
## Mun 4      4 0.4661841 0.5543717 0.6819990
## Mun 1      8 0.5648341 0.6482423 0.7311045
## Mun 2      8 0.4652264 0.6528622 0.7801291
## Mun 3      8 0.4227668 0.5305285 0.6181426
## Mun 4      8 0.4553128 0.5097342 0.5790670
```

Note, however, that several reference values in the test data set are entered with zero uncertainty.

Estimate the effect of uncertainty in reference values

One may want to estimate the uncertainty in the calculated NI values that results from uncertainty in the reference values only.

The code for doing this is again the same as above, however this time the assignment

```
Alt_DistID[Observations$ReferenceYearID==0] <- "NoBoot"
```

should be replaced with

```
Alt_DistID[Observations$ReferenceYearID!=0] <- "NoBoot"
```

```
##      year      0.025      0.5      0.975
## Mun 1      2 0.7681888 0.7681888 0.7681888
```

```
## Mun 2      2 0.7376662 0.7376662 0.7376662
## Mun 3      2 0.6423352 0.6423352 0.6423352
## Mun 4      2 0.6173134 0.6173134 0.6173134
## Mun 1      3 0.7252688 0.7252688 0.7252688
## Mun 2      3 0.6553767 0.6553767 0.6553767
## Mun 3      3 0.5672605 0.5672605 0.5672605
## Mun 4      3 0.6336953 0.6336953 0.6336953
## Mun 1      4 0.7296070 0.7296070 0.7296070
## Mun 2      4 0.7368540 0.7368540 0.7368540
## Mun 3      4 0.5965567 0.5965567 0.5965567
## Mun 4      4 0.5726695 0.5726695 0.5726695
## Mun 1      8 0.6574743 0.6574743 0.6574743
## Mun 2      8 0.7100785 0.7100785 0.7100785
## Mun 3      8 0.5615443 0.5615443 0.5615443
## Mun 4      8 0.5155545 0.5155545 0.5155545
```

The result illustrates what was noted above, that many reference values in the test data set lack estimates of uncertainty.

Calculate NI without uncertainties

Again, the same code as above, however this time all entries in Alt_DistID are set to “NoBoot”.

```
##      year      0.025      0.5      0.975
## Mun 1      2 0.7681888 0.7681888 0.7681888
## Mun 2      2 0.7376662 0.7376662 0.7376662
## Mun 3      2 0.6423352 0.6423352 0.6423352
## Mun 4      2 0.6173134 0.6173134 0.6173134
## Mun 1      3 0.7252688 0.7252688 0.7252688
## Mun 2      3 0.6553767 0.6553767 0.6553767
## Mun 3      3 0.5672605 0.5672605 0.5672605
## Mun 4      3 0.6336953 0.6336953 0.6336953
## Mun 1      4 0.7296070 0.7296070 0.7296070
## Mun 2      4 0.7368540 0.7368540 0.7368540
## Mun 3      4 0.5965567 0.5965567 0.5965567
## Mun 4      4 0.5726695 0.5726695 0.5726695
## Mun 1      8 0.6574743 0.6574743 0.6574743
## Mun 2      8 0.7100785 0.7100785 0.7100785
## Mun 3      8 0.5615443 0.5615443 0.5615443
## Mun 4      8 0.5155545 0.5155545 0.5155545
```

The difference between NI calculated with uncertainties and NI calculated without, is an estimate of the so-called location displacement mentioned on page 37 in Pedersen *et al.* (2016)² and discussed in depth in Pedersen and Skarpaas (2012)³.

```
##      year      0.025      0.5      0.975
## Mun 1      0 -0.15910728 -0.042777644 0.04640837
## Mun 2      0 -0.24252659 -0.044742726 0.07190250
## Mun 3      0 -0.14366310 -0.013719718 0.09849744
## Mun 4      0 -0.13682498 -0.006045307 0.10157343
## Mun 1      0 -0.13914619 -0.028653244 0.06837223
```

²Pedersen, B., Nybø, S., Sæther, S. A. (eds.) 2016. Nature Index for Norway 2015. Ecological framework, computational methods, database and information systems – NINA Report 1226. 84 pp.

³Pedersen, B., Skarpaas, O. 2012. Statistiske egenskaper til Naturindeks for Norge. Usikkerhet i datagrunnlaget og sensitivitet. - NINA Rapport 797, Trondheim. 55 pp. (In Norwegian)

## Mun 2	0	-0.22147636	-0.017326349	0.10989180
## Mun 3	0	-0.15052991	-0.015572187	0.10756206
## Mun 4	0	-0.16064939	-0.041799143	0.05796567
## Mun 1	0	-0.12009677	-0.024429727	0.08363227
## Mun 2	0	-0.23016017	-0.037553364	0.08042057
## Mun 3	0	-0.13430541	-0.019728064	0.11247064
## Mun 4	0	-0.10231504	-0.015753055	0.11016706
## Mun 1	0	-0.09280038	-0.009430786	0.07611189
## Mun 2	0	-0.24543732	-0.057670802	0.06707462
## Mun 3	0	-0.13886485	-0.028157352	0.05193083
## Mun 4	0	-0.05943395	-0.002893717	0.06933075