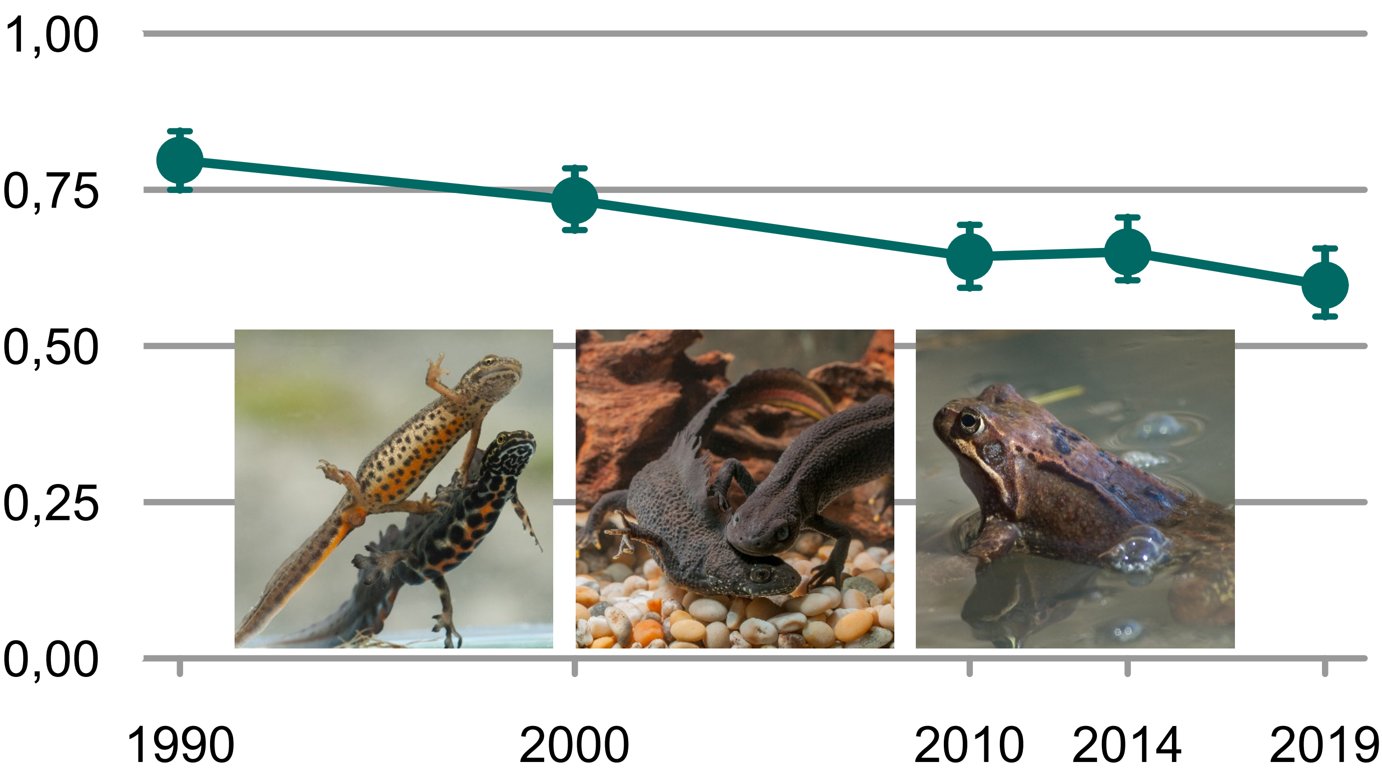
www.nina.no

NINA Report 1990

Nature Index system documentation

Mathematical framework, database, web-portals, scripts and API

Siw Elisabeth Berge
Bård Pedersen



NINA
Norwegian Institute for Nature Research

NINA Publications

NINA Report (NINA Rapport)
This is NINA's ordinary form of reporting completed research, monitoring or review work to clients. In addition, the series will include much of the institute's other reporting, for example from seminars and conferences, results of internal research and review work and literature studies, etc. NINA

NINA Special Report (NINA Temahefte)
Special reports are produced as required and the series ranges widely: from systematic identification keys to information on important problem areas in society. Usually given a popular scientific form with weight on illustrations.

NINA Factsheet (NINA Fakta)
Factsheets have as their goal to make NINA's research results quickly and easily accessible to the general public. Fact sheets give a short presentation of some of our most important research themes.

Other publishing.
In addition to reporting in NINA's own series, the institute's employees publish a large proportion of their research results in international scientific journals and in popular academic books and journals.

Nature index system documentation

Mathematical framework, database, web-portals, scripts and API

Siw Elisabeth Berge
Bård Pedersen

Norwegian Institute for Nature Research

Berge, S.E. & Pedersen, B. 2021. Nature index system documentation. Mathematical framework, database, web-portals, scripts and API. NINA Report 1990. Norwegian Institute for Nature Research.

Trondheim, April 2021

ISSN: 1504-3312
ISBN: 978-82-426-4769-6

AVAILABILITY
Open

PUBLICATION TYPE
Digital document (pdf)

QUALITY CONTROLLED BY
Roald Vang

SIGNATURE OF RESPONSIBLE PERSON
Research Director Signe Nybø (sign.)

CLIENT(S)/SUBSCRIBER
Norwegian Environment Agency

CLIENT REFERENCE
M-2024|2021

CLIENTS/SUBSCRIBER CONTACT PERSONS
Ragnvald Larsen, Eirin Bjørkvoll

COVER PICTURE
Thematic index amphibia for Eastern Norway 1990 – 2019.
Common newt, great crested newt and common frog
© Børre Dervo

KEY WORDS
Nature Index, thematic index, Nature Index database,
Nature Index data entry portal, http://naturindeks.nina.no,
public Nature Index web-page, www.naturindeks.no, NIcalc.

NØKKELORD
Naturindeksen, temaindeks, Naturindeks database,
Innlesingsportal for Naturindeksen, http://naturindeks.nina.no,
www.naturindeks.no, NIcalc.

CONTACT DETAILS

NINA head office	NINA Oslo	NINA Tromsø	NINA Lillehammer	NINA Bergen:
P.O.Box 5685 Torgarden	Sognsveien 68	P.O.Box 6606 Langnes	Vormstuguvegen 40	Thormøhlens gate 55
NO-7485 Trondheim	0855 Oslo	NO-9296 Tromsø	NO-2624 Lillehammer	NO-5006 Bergen.
Norway	Norway	Norway	Norway	Norway
P: +47 73 80 14 00	P: +47 73 80 14 00	P: +47 77 75 04 00	P: +47 73 80 14 00	P: +47 73 80 14 00

www.nina.no

Abstract

Berge, S.E. & Pedersen, B. 2021. Nature Index system documentation. Mathematical framework, database, web-portals, scripts and API. NINA Report 1990. Norwegian Institute for Nature Research

The Nature Index measures the state of biological diversity in Norway. It gives an overview over how this state changes in ecosystems, and over developments in selected species groups and themes. This report aims to document the mathematical framework and theoretical concepts of the Nature Index, and the technical solutions that the Nature Index relies upon.

To set the context for the systems, the mathematical framework and the dataset that the Nature Index is built upon is explained. The database is a relational database where tables are linked with keys. The database tables, it's structure, content and logic are listed together with the technical specifications. Two web-based portals interact with the database; the data entry portal (naturindeks.nina.no) which is used by the indicator experts to deliver data for their indicators, and the public data portal (naturindeks.no) which gives a public friendly presentation of the indicators, ecosystem and thematic indices with the calculated Nature Index values. Both the web-portals's main functionalities and the technical platforms and frameworks are documented. The organization of programming code is also listed for the codebase for the two web applications.

The Nature Index project has developed various scripts to perform a series of tasks in relation to calculations of the Nature Index, and they are bundled in the NIcalc package. In addition, the package facilitates a more automated process for experts to deliver their indicator data. All communication with the database goes through the Nature Index API (application programming interface), which is a set of methods to import and export data to the database.

Siw Elisabeth Berge (siw.berge@nina.no) and Bård Pedersen (bard.pedersen@nina.no), Norwegian Institute for Nature Research, P.O. Box 5685 Torgarden, NO-7485 Trondheim.

Sammendrag

Berge, S.E. & Pedersen, B. 2021. System dokumentasjon for Naturindeks. Matematisk rammeverk, database, web-portaler, R-kode og API. NINA Rapport 1990. Norsk institutt for naturforskning.

Naturindeksen måler tilstanden til det biologiske mangfoldet i Norge, og gir en oversikt over utviklingen i økosystemene, for utvalgte artsgrupper og tema. Denne rapporten har som mål å dokumentere det matematiske rammeverket og de teoretiske konsepter som ligger til grunn for Naturindeksen og også dokumentere de tekniske løsningene som Naturindeksen er bygget på.

Innledningsvis blir det matematiske rammeverket og datasettet til Naturindeksen gjennomgått. Databasen er en relasjonsdatabase der en rekke tabeller er forbundet til hverandre med henvisninger. Databasens tabeller, innhold, struktur og utviklet logikk er beskrevet. Naturindeksen har to web-løsninger som kommuniserer med databasen; en data-innleggingsportal som ekspertene bruker for å levere data om sine indikatorer, og en innsynsløsning for publikum som presenterer indikatorer, økosystemer og temaindekser og den beregnede Naturindeksen for disse i kart og grafer. For begge disse løsningene er funksjonaliteten beskrevet sammen med de tekniske plattformene og rammeverkene som er benyttet. Strukturen til programkoden til de to web-applikasjonene er også dokumentert.

Naturindeks-prosjektet har utviklet en samling script for å utføre ulike oppgaver i forbindelse med beregninger av Naturindeksen og disse er samlet i en pakke kalt NIcalc. Pakken støtter i tillegg en mer automatisert prosess for oppdatering av indikatordata. All kommunikasjon med databasen skjer igjennom Naturindeks API (application programming interface) som er et sett med metoder for import og eksport av data til databasen.

Siw Elisabeth Berge (siw.berge@nina.no) og Bård Pedersen (bard.pedersen@nina.no), Norsk institutt for naturforskning, Postboks 5685 Torgarden, NO-7485 Trondheim.

Contents

Abstract 3

Sammendrag 4

Contents 5

Foreword 7

1 Introduction 8

2 Mathematical framework and the Norwegian Nature Index dataset 10
- 2.1 Mathematical framework 10
  - 2.1.1 Definition 10
  - 2.1.2 Indicators 10
  - 2.1.3 Spatial units 10
  - 2.1.4 Indicator observations 11
  - 2.1.5 Scaling 11
  - 2.1.6 Weights 12
  - 2.1.7 Calculation 14
  - 2.1.8 Missing indicator observations – multiple imputations 14
  - 2.1.9 Statistical properties 16
- 2.2 Thematic indices and indicator indices 17
- 2.3 Data 18

3 Nature Index database 22
- 3.1 The Nature Index Main Tables 22
- 3.2 The result tables from R calculations 26
- 3.3 Functions and stored procedures 27
- 3.4 Database content 28

4 Web-interface for data entry 29
- 4.1 Functionality – a brief overview 29
- 4.2 Technologies, frameworks and code organization 29
  - 4.2.1 Technologies and frameworks 29
  - 4.2.2 Code organization 30

5 Public Nature Index web-site 34
- 5.1 Background 34
- 5.2 The pages 34
- 5.3 Technologies and frameworks 35
- 5.4 Code organization 36

6 R-scripts and NIcalc 39

7 Nature Index API 47

8 References 49

9 Appendix 1 - Nicalc - example runs 51
- 9.1 Calculating the Nature Index for mountains 51
- 9.2 Calculating a thematic index: amphibia 55
- 9.3 Distribution objects 56
- 9.4 Updating indicator observations, and communicating with the database 59

10 Appendix 2 - S3 classes defined within NIcalc 63
10.1 Class *niDataImport* 63
10.2 Class *niInput* 63
10.3 Class *niImputations* 64
10.4 Results: Classes *niOutput*, *niSeries* and *niValue* 65
10.5 class *indicatorData* 66

Foreword

The Nature Index was launched the first time in 2010. Since then, the index has been reported every fifth year, in 2015 and 2020. The Norwegian Environment Agency coordinates the work with the Norwegian Nature Index, commissioned by the Norwegian Ministry of Climate and Environment. The Norwegian Institute for Nature Research (NINA) has been responsible for developing and revising the Nature Index frameworks, for establishing and updating a central database storing data and results, for developing information systems for online results visualization, and for calculating the index. The work and the present report has been financed by the Norwegian Environment Agency.

In this technical report we document the various components of the information systems, the procedures and scripts for calculating the index, as well as the mathematical framework behind the Nature Index, as of November 2020.

The intended reader is not the general public, rather, the report addresses those who in the future will be involved in revising and improving the systems.

The systems described here have been developed and over a ten year period. In addition to the authors, many persons have provided large and small contributions to the development of the systems and frameworks described here. Especially, we want to acknowledge the significant contributions from Jens Åström, Grégoire Certain, Simon Jakobsson, Pål Kvaløy, Signe Nybø, Stein Are Sæther, Olav Skarpaas, and Øystein Solberg. During this period, Else Løbersli, Eirin Bjørkvoll, and Ragnvald Larsen have been our contact persons at the Norwegian Environment Agency.

23 April 2021 Siw Elisabeth Berge and Bård Pedersen

1 Introduction

The Nature Index is a framework for condensed reporting of the state of nature (Certain *et al*. 2011, Pedersen *et al*. 2016). The composite index synthesizes and communicates knowledge about states and trends in nature. The Nature Index does this by summarizing measurements and assessments of the state of indicators, which, together, represent biodiversity. The Nature Index is calculated as a weighted average of scaled indicator states (Chapter 2). Experts from various research institutions provide estimated and observed indicator values.

Figure 1.1 describes the IT information system that has been developed for data input, storage and presentation of the Nature Index. The database store indicator observations, indicator metadata and other data for the calculation and presentation of the Nature Index. It also stores the calculations for the different indicators, ecosystems and indices. The purpose of the website naturindeks.nina.no is that the experts can update information about the indicators for which they are responsible. Based on the data provided via this website, statistical calculations are performed in the R-package and results are written to the database. The purpose of the web-portal www.naturindeks.no is to present the results of the calculations and the background data to the public.

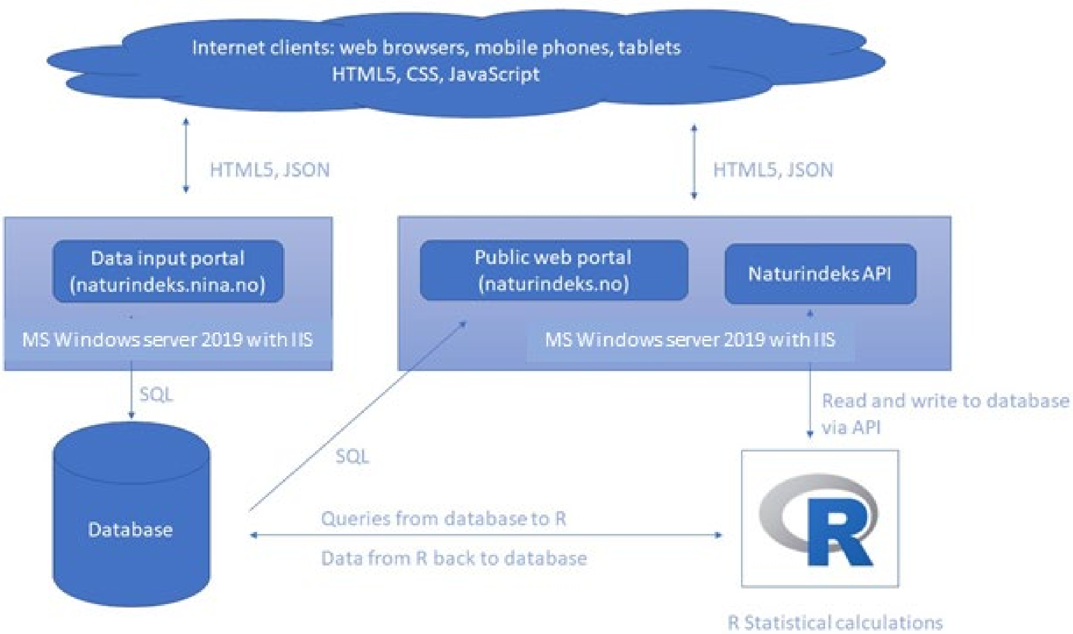


Figure 1.1 Overview of the Nature Index IT system

In this technical report we document the various components of the IT-system, the procedures and scripts for calculating the index, as well as the mathematical framework behind the Nature Index and statistical properties documented so far. Details of the systems and frameworks are revised for each launching of the index. The current report describes the system of the launching of the Norwegian Nature Index in November 2020.

Chapter 2 contains a detailed account of the mathematical framework for calculating the nature index. The framework determines the necessary data input in the NI calculations. These data are stored in the NI-database. The mathematical framework provides the necessary background for understanding the purpose and operation of the R-functions involved in the calculation of NI. These functions are bundled together in the R-package *NIcalc* (Chapter 6). The framework also introduces the terminology associated with indicators, indicator measurements, spatial units, scaling and weights. Other accounts of the framework are found in Certain and Skarpaas (2010), Certain *et al*. (2011) and Pedersen *et al*. (2016). Chapter 2 also discusses methods for dealing with missing values in time series of indicator measurements, and the technicalities of how to implement such methods in the calculation of the Nature Index. In addition, the chapter reviews some statistical issues related to the Nature Index.

Chapter 2 also gives an overview of the information contained in the database. The Nature Index database contains not only the necessary information for calculating the Nature Index for Norway, but also pictures and supplemental data that characterize the various indicators, and which are presented on the public Nature Index web-page. It also contains descriptions of the datasets behind each indicator and the procedures followed to determine reference values and to adapt the data to the Nature Index framework. Background documents and links to such documentation available from other sources are also stored in the database. It also contains various other types of data, e.g. map polygons and information necessary to administer experts' access to the database.

Chapters 3 describes the Nature Index database; the tables, the stored procedures and functions as well as technical information are outlined. Chapter 7 document the Nature Index Application Programming Interface (API), a set of methods that provides import and export functionality from the NI database.

Chapter 4 and chapter 5 is respectively about the Nature Index data management portal and the public Nature Index web site. Both chapters give a brief overview of the websites' functionality and the technical frameworks the sites are built upon. The code is documented by all the endpoints and the belonging functionality.

Chapter 6, appendix 1 and appendix 2 describes the R-code developed for calculating the Nature Index. R is an open source programming language and software environment developed for statistical computing and graphics (R Core Team 2020). Functions developed in R also perform a series of additional tasks within the Nature Index IT information system. These functions are assembled in an open source R-library called *NIcalc*.

2 Mathematical framework and the Norwegian Nature Index dataset

2.1 Mathematical framework

2.1.1 Definition

The Nature Index (NI) is a weighted average of scaled indicator states

$$(1) \qquad NI_{jKt} = \sum_{i=1}^{n} w_{ijK} S_{ijKt}$$

where $S_{ijKt}, i = 1, \ldots, n$ are the n scaled indicator states included in the calculation. The weights (w_{ijK}) obey the condition $\sum_{i=1}^{n} w_{ijK} = 1$, while the indicator states are scaled to range between zero and 1. The Nature Index is calculated for a specified major habitat (j) in a defined spatial unit (K, here referred to as an *NIunit*) and for a particular year (t) from a set of measurements of indicator states.

2.1.2 Indicators

In the Norwegian implementation of NI, indicators are typically the state of individual species. Most often abundance is used to represent state. In addition, indicators are various types of community indices that represent groups of species with similar ecological function. For a few indicators, substitutes for species and species groups are used to represent state. Such substitutes may be dominating environmental or biological factors with a negative impact on abundance or community composition, or important resources. Jakobsson & Pedersen (2020) contains lists of indicator sets for each major habitat in the Norwegian implementation of the NI-framework as of 17.11.2020. The open web portal http://www.naturindeks.no/ presents each indicator in detail.

2.1.3 Spatial units

Measurements of indicator states are collected from indicator-specific spatial units (here referred to as *ICunits*) that may have a different spatial extent than the NIunit. 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 measurements are collected. However, each ICunit and NIunit must consist of one or more basic spatial units (*BSunit*). The set of BSunits varies among implementations of the Nature Index framework. The Norwegian implementation of the Nature Index uses municipalities as of 01.01.2010 as BSunits, while the pilot implementation in Costa Rica used a hexagonal grid of spatial units.

Thus, in order to calculate the index, the delineation of each IC- and NIunit must be provided. That is, each ICunit and NIunit must be specified in terms of the BSunits they consist of. Further, all indicator measurements must be linked to the correct ICunit.

2.1.4 Indicator observations

In the Norwegian implementation of the Nature Index, indicator measurements (values) are generated in three different ways. Either they are estimated from monitoring programs, by expert judgment, or as predictions from models (Pedersen *et al*. 2016).

All indicators are supposed to be nonnegative variables with zero as the minimum value.

Indicator measurements may be associated with errors and are therefore uncertain. Thus, in the context of the Nature Index, measurements are specified as probability distributions where the dispersion of the distribution represents this uncertainty, and the central tendency of the distribution represents the magnitude of the indicator value. Thus, each measurement is treated as a stochastic variable.

Probability distributions representing indicator measurements may be provided in different formats. Measurements generated from model predictions are typically specified as standard probability distributions (e.g. lognormal distribution) together with estimates of the distributions' parameters; or as large sets/samples of observations from the distributions, generated from the models through e.g. simulations or MCMC runs.

Probability distributions representing expert judgments are typically provided through an elicitation process. 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. In the first step, experts estimate the distribution's expected value and its lower and upper quartiles for all indicator values, where the interquartile distance measures the uncertainty in the measurements. In the second step, a probability distribution is fitted from among a number of model distributions for each indicator value. The fitting is based on a least squares criterion. Model distributions are non-negative, univariate distributions. They are the 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.

2.1.5 Scaling

The indicators are measured or observed using measurement scales that are specific to the individual indicator. Scaling indicator values to a common scale is however necessary to calculate a meaningful average. This is done using nonlinear (piecewise linear) scaling functions. 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' range is the interval [0,1].

There are two types of scaling functions, LOW and MAX:

LOW:

$$(2) \qquad S_i = \begin{cases} \dfrac{U_i}{Ref_i}, & 0 \le U_i \le Ref_i \\ 1, & U_i > Ref_i \end{cases}$$

MAX:

$$(3) \qquad S_i = \begin{cases} 1, & 0 \le U_i \le Ref_i \\ 2 - \dfrac{U_i}{Ref_i}, & Ref_i \le U_i \le 2 * Ref_i \\ 0, & U_i > 2 * Ref_i \end{cases}$$

U_i is the state of indicator *i* measured on the original, indicator specific scale.

The choice of scaling function is determined by whether the indicator relates positively or negatively to the aspect of biodiversity that the indicator represents. The LOW model is used when there is a positive relation between the indicator and biodiversity. This holds true for most indicators. The MAX model is used when there is a negative relation between the indicator and biodiversity. This model applies only to certain indirect indicators that represent a negative effect that the measured indicator has on other components of biodiversity.

2.1.6 Weights

When calculating the Nature Index, state values are weighted in relation to each other (equation 1). Some measurements of indicator states are given more weight than others. The rationale behind these weights is to correct for biases in the data from which the index is calculated (cf. Certain and Skarpaas 2010).

In the calculation of NI for a *BSunit*, the weights assigned to indicator states depend on the indicators' specificity / fidelity to the respective major ecosystem and the indicators' ecological function. They can be written as a product of two factors, a *trophic weight* ($w^{trophic}$) and a *fidelity weight* ($w^{fidelity}$). A grouping of the indicators into trophic groups and key indicators according to their ecological function (cf. Certain and Skarpaas 2010), is the basis for calculating these weights.

Since the first launching of the Nature Index in 2010, indicator fidelities have been set in two different ways. In 2010 and 2015 versions of the index, indicators' *fidelities* (φ_{ij}) to the various major habitats were determined by the individual expert based on the degree to which the underlying data for the indicator reflected the state of one or more of the major habitats. An indicator's total fidelity over all major habitats was supposed to be 100%, i.e. $\sum_{j=1}^{p} \varphi_{ij} = 100\%$, where *p* is the number of such habitats.

Many indicators belong to only one of the major habitats; for example, the blue mussel is an indicator for only the coast major ecosystem and have a fidelity of 100% to the coast, with a fidelity of 0% to the other major ecosystems. Conversely, marine fish species will often belong to both marine major habitats, ocean and coastal water. 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 habitats 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 habitat. In some cases, ecosystem-specific data is available for a species, and then separate indicators with 100% fidelity could be used.

In the calculation of the 2020 index, *fidelities* were either 0% or 100%. I.e. indicator reflecting the state of two or more of the major habitats were not downweighted in the calculations compared to those reflecting the state of one single major habitat.

The *fidelity weight* assigned to a measurement of indicator *i* belonging to trophic group g in the calculation of an index for major habitat *j* in BSunit *k*, is given by

$$(4) \qquad w_{ijk}^{fidelity} = \frac{\varphi_{ij}}{\sum_{m \in g} \varphi_{mj}}$$

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

Trophic weights ($w^{trophic}$) depend on the presence of key indicators and the number of other functional (trophic) groups represented in the BSunit:

$$(5) \qquad w_{ijk}^{trophic} = \begin{Bmatrix} \frac{1}{2}, & if\ i\ is\ a\ key\ indicator, nonkey\ indicators\ present \\ \frac{1}{1}, & if\ i\ is\ a\ key\ indicator, nonkey\ indicators\ not\ present \\ \frac{1}{2r_{jk}}, & if\ i\ is\ a\ nonkey\ indicator, key\ indicators\ present \\ \frac{1}{r_{jk}}, & if\ i\ is\ a\ nonkey\ indicator, key\ indicators\ not\ present \end{Bmatrix}$$

r_{jk} is the number of functional groups represented with indicator measurements in BSunit k.

The product of an indicator measurement's fidelity weight and trophic weight is the total weight (w^{BSunit}) assigned to the measurement in a calculation of the Nature Index for BSunit k,

$$(6) \qquad w_{ijk}^{BSunit} = w_{ijk}^{fidelity} * w_{ijk}^{trophic}$$

and the Nature Index for major habitat j in the unit at time t is

(7) $$NI_{jkt} = \sum_{i=1}^{n} w_{ijk}^{BSunit} S_{ijkt}$$

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

The consequences of these assignments are first, in most cases where both key- and non-key indicators are present, that the sum of the weights w_{ijk}^{BSunit} for key indicators is $\frac{1}{2}$. So key indicators weigh 50% in the Nature Index for a municipality or BSunit. Second, the different trophic groups represented in the BSunit are weighted equally, regardless of the number of indicators represented within each group.

Equation (7) is used to calculate the Nature Index when NIunits correspond to single BSunits. Aggregated indices for NIunits (K) consisting of several BSunits are calculated as weighted averages of municipal indices, e.g. for country parts or the whole country:

(8) $$NI_{jKt} = \sum_{k \in K} w_{jk}^{area} NI_{jkt} = \sum_{k \in K} \sum_{i=1}^{n} w_{jk}^{area} w_{ijk}^{BSunit} S_{ijkt}$$

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 BSunit k, and $\sum^{k \in K} w_{jk}^{area} = 1$. Thus, the weight assigned to a measurement of indicator i from ICunit C, where C covers several BSunits, is $\sum_{k \in K \cap C} w_{jk}^{area} w_{ijk}^{BSunit}$.

2.1.7 Calculation

Since the Nature Index is calculated from indicator measurement 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 measurements included in the calculation. The Nature Index value is then calculated as a weighted average of the sample of draws. The index's distribution is simulated by repeating this procedure $nsim = 1000$ 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 in the index estimate.

2.1.8 Missing indicator observations – multiple imputations

The Nature Index is often calculated as a time series in order to describe how the state of biodiversity changes over the period covered. For example, the latest edition of the Norwegian Nature Index was calculated for the period from 1990 to 2019 (Jakobsson & Pedersen 2020). During such a period, monitoring programs are terminated, new ones are added, the preconditions for making expert judgments will change, etc. This means that the indicator set and which time series are documented with data change throughout the period. These and many other factors lead to gaps in the series.

Such gaps are problematic for several reasons. First, as a consequence of such gaps, weights assigned to particular measurement series in index-calculations will vary throughout the period (cf. chapter 2.1.6). This will cause the Nature Index to vary accordingly, even in the absence of any variation over time in the documented time series (cf. equation 1). Second, variation among indicator measurements are primarily among indicators, and not so much among geographical areas or years for the same indicator. Furthermore, a common pattern of missing values in the data is that all data for some indicators are missing for some years. Thus, the risk of bias occurring among years in the sample of indicator measurements, and consequently in the calculated index values, is relatively high, especially when many indicators lack measurements from the same year. Third, missing data should increase the uncertainty in the calculated index. Unfortunately, the calculation of the index, as described above, only accounts for measurement uncertainty in the data, not sampling uncertainty with respect to indicators.

Multiple imputations (Van Buuren 2018) are therefore calculated for all missing values. Like measurements, each imputation is a probability distribution (cf. chapter 2.1.4). In practice, the imputation method calculates an expected value, a lower quartile and the interquartile distance (ID) for each imputed value. The upper quartile is calculated from the lower quartile and the interquartile distance. The same elicitation procedure as described in chapter 2.1.4 is used to determine the imputed distribution. By imputing indicator measurements, the first issue raised above is remedied, irrespective of the imputation method used.

Two general approaches for imputing multivariate data have emerged: joint modeling (JM, Schafer 1997, Honaker & King 2010) and fully conditional specification (FCS), also known as multivariate imputation by chained equations (MICE, Van Buuren 2007). JM involves specifying a multivariate distribution for the missing data, and drawing imputation from their conditional distributions by Markov chain Monte Carlo (MCMC) techniques. This methodology is attractive if the multivariate distribution is a reasonable description of the data. FCS specifies the multivariate imputation model on a variable-by-variable basis by a set of conditional densities, one for each incomplete variable. Starting from an initial imputation, FCS draws imputations by iterating over the conditional densities.

A JM approach was applied for missing data in NI2015 using the R-package "Amelia" (Honaker *et al*. 2011). This approach was, however, not robust when implemented as a general method for all indicators. The routine often crashed (e.g. when the joint multivariate normal distribution model was not suitable) and sometimes led to fatal errors in the CPU. Currently, therefore, the FCS approach is applied, using predictive mean matching implemented in the R-package "mice" (Van Buuren & Groothuis-Oudshoorn 2011), as imputation method.

Indicator measurements are normalized against their corresponding reference value and thereafter log-transformed before imputation modelling. The imputation model includes five variables in the order logmean, loglower, logID, year, indicator. Note that the aim of the model is to remedy the second issue associated with missing values described above, i.e. to impute “neutral” values, typical for the indicator and year in question, that do not add new patterns to the time series data. With neutral or typical values we here mean measurements with an expected deviation from the (grand) mean equal to the sum of 1) the mean deviation for measurements from the year in question, and 2) the mean deviation from annual means for measurements of the indicator in question. In some instances, especially when working with very unbalanced data, the imputation model may not produce neutral values as desired. In such cases, *post hoc* adjustment of imputed values is necessary to avoid introducing new patterns in the data set.

$nsim$ probability distributions are imputed for each missing value. For each of the $nsim$ simulations of the Nature Index (chapter 2.1.7), one distribution is randomly chosen for each missing value, and one single observation is randomly drawn from each of the selected distributions and entered to the calculation of the weighted mean. This two-stage sampling procedure adds the extra uncertainty associated with missing indicator measurements to the distribution of the Nature Index.

2.1.9 Statistical properties

Location displacement: With uncertain data, the expected value of the Nature Index, μ_{NI}, will, in general, differ from the weighted mean of scaled expected values of indicator measurements (Pedersen & Skarpaas 2012), i.e.

$$(9) \qquad \mu_{NI_{jKt}} \neq \sum_{i=1}^{n} w_{ijK} S_i\left(\mu_{U_{ijKt}}\right),$$

where $S(\mu_U)$ is the scaled expected values of indicator measurements. The same also holds for the median, m_{NI}, which is used as a point estimate for the Nature Index (cf. chapter 2.1.7):

$$(10) \qquad m_{NI_{jKt}} \neq \sum_{i=1}^{n} w_{ijK} S_i\left(m_{U_{ijKt}}\right).$$

This non-intuitive phenomenon has been referred to as location displacement. Expected values are “displaced” during scaling, i.e. $\mu_{S_{ijKt}} \neq S_i\left(\mu_{U_{ijKt}}\right)$, because both scaling models LOW and MAX are nonlinear, while medians are “displaced” during the summation of scaled indicator values, $m_{NI_{jKt}} \neq \sum_{i=1}^{n} w_{ijK} m_{S_{ijKt}}$. Location displacement in a calculated index, b_{NI}, is measured as

$$(11) \qquad b_{NI_{jKt}} = m_{NI_{jKt}} - \sum_{i=1}^{n} w_{ijK} S_i\left(\mu_{U_{ijKt}}\right).$$

The size of location displacement depends on scaling model, location and dispersion of the indicator measurements’ distributions, and the number of measurements (n). Displacement may be both positive and negative. It increases in absolute value with increasing dispersion of the measurements’ distributions, and when the measurements’ expected values approach the reference values. When the index is calculated from a small number of measurements, location displacement will increase in absolute value before approaching a limit as new measurements are added to the calculation (Pedersen & Skarpaas 2012).

Uncertainty in reference values: The Nature Index calculated for Norway seems to be robust with respect to errors in the reference values (Pedersen & Skarpaas 2012, Pedersen *et al*. 2016, Schartau *et al*. 2016). This robustness is due to several factors. First, the index is an average of many scaled indicator measurements. Second, most indicators in the data set are scaled with the LOW model which is relatively robust with respect to such errors. Measurements larger and somewhat smaller than the reference value are little or not affected by errors in the reference value (Schartau *et al*. 2016). Third, a large dispersion in the location of unscaled measurements relative to the reference values, together with a relatively high level of uncertainty in individual measurements, which both are characteristic of the Norwegian data set (cf. Pedersen *et al*. 2018, 2019, Johansen *et al*. 2019), contribute to this robustness.

Weights and uncertainty in the Nature Index: The Nature Index is calculated under the assumption that indicator measurements are independent with respect to measurement error (cf. chapter 2.1.7). Expressed as a variance, σ^2, the uncertainty in the index estimate is

$$(12) \qquad \sigma^2_{NI_{jKt}} = \sum_{i=1}^{n} w^2_{ijK} \sigma^2_{S_{ijKt}}$$

Thus, the uncertainty in the index estimate depends on how weights are distributed over indicator measurements in addition to the uncertainty in the measurements themselves. As explained in chapter 2.1.6, weights are calculated according to an *a priori* defined system. Unlike algorithms that calculate weights *a posteriori* to minimizes the uncertainty in the estimated average, the Nature Index system often produces a very uneven distribution of weights over indicator measurements which in turn results in an increased uncertainty in the index compared to an unweighted average. Instead of producing weights that are inversely proportional to the measurements' variance, the Nature Index system may produce weights that are unrelated or even positively correlated to the uncertainty in the measurements (e.g. Pedersen *et al*. 2018). As a result, the uncertainty of the Nature Index estimate depends on the uncertainty of the few, most heavily weighted measurements, most often measurements of key indicators (chapter 2.1.6, Pedersen *et al*. 2018, 2019, Johansen *et al*. 2019). As shown by Pedersen *et al*. (2019), adding new observations to the data set will hardly reduce the uncertainty of the index unless weights assigned to key indicator observations are reduced as a consequence. Pedersen *et al*. (2019) discusses strategies for revising and extending input data sets to Nature Index calculations.

2.2 Thematic indices and indicator indices

The Nature Index dataset is also used as a basis for calculation of so-called thematic indices. Such indices reflect particular management themes, environmental pressures, and/or the development in groups of species or nature types of particular interest (Pedersen *et al*. 2013, Framstad 2015, Jakobsson & Pedersen 2020) etc. A thematic index is similar in construction as the Nature Index, but is most often composed of a smaller selection of indicators that form part of the Nature Index. The system for weighting indicator measurements, described in Chapter 2.1.6, has often little or no relevance for the calculation of thematic indices. The system for weighting indicator measurements varies among the various indices, depending on the selection of indicators used to construct them. Most often, fidelities and functional groupings are ignored when calculating the weights for thematic indices.

Indicator indices give an area-weighted summary of the state for individual indicators. Indicator indices are weighted averages of “normalized” measurements (N):

$$(13) \qquad N_i = \frac{U_i}{Ref_i}, \qquad U_i \geq 0$$

$$(14) \qquad N_i = \begin{cases} 2 - \frac{U_i}{Ref_i}, & 0 \leq U_i \leq 2 * Ref_i \\ 0, & U_i > 2 * Ref_i \end{cases},$$

where equation (13) is used when there is a positive relation between the indicator and biodiversity, and equation (14) is used when there is a negative relation.

Framstad (2015), Jakobsson & Pedersen (2020), and the public Nature Index web-page present examples of thematic- and indicator indices.

2.3 Data

The Nature Index database contains information on a series of different types of objects. They are listed in table 2.1 together with the attributes that characterize each object. In addition to input data to Nature Index calculations, the database stores information necessary to control and manage web-based data entry by experts, and to present results from calculations and other information on the public web-page.

As input to Nature Index calculations, each *indicator* is characterized by a name and/or an ID, which type of scaling model that should be used to scale its values, which trophic group it belongs to, and whether it is a key indicator or not. In addition, one must provide the indicators’ fidelity to each major habitat. Additional info about indicators, which is not used in the calculations is also available in the database, i.e., descriptions of the indicator, red list status, taxonomy, pictures, documentation and links to external resources, responsible experts and sensitivity to pressure factors. Indicators may be “active” or “passive”. Passive indicators are neither included in index calculation nor presented on the public Nature Index web-page. Passive indicators are mostly former, active indicators that no longer is updated with new data.

Indicator measurements (values) may be stored in two ways (cf. chapter 2.1.4). Either as distribution objects generated in R (Ruckdeschel *et al*. 2006, chapter 6) and imported to the database as text documents, or as expected values together with lower- and upper quartiles, optionally together with the results from the elicitation process described in chapter 2.1.4, i.e. distribution families and estimated model parameters. In the calculations indicator values must be related to the indicator being measured, year of measurement, and indicator area. Further attributes of indicator measurement are the results from the elicitation process where model probability distributions are fitted to each measurement; i.e. distribution families and estimated model parameters. In addition, the database stores information about datatype (monitoring, expert judgment, model prediction) and the original measurement unit for each indicator value.

Table 2.1 Objects in the Nature Index database and their attributes.

	Object type	Attribute
Input objects to Nature Index calculations	Indicators	Name and ID Type of scaling model Trophic group Key- or non-key indicator Fidelity to major habitats Sensitivity to pressure factors Indicator description Description of data collection Reference value estimation Red list status Taxonomy Pictures Documentation Responsible experts Active or passive
	Indicator measurements	ID Indicator ICunit Year Type Expected values Lower- and upper quartiles Model distribution family and parameters Distribution object Measurement unit
	Model probability distributions	Name and ID
	Scaling functions	Name and ID
	Reference values	ID Indicator ICunit Type Expected values Lower- and upper quartiles Model distribution family and parameters Distribution object Measurement unit
	Functional groups	Name and ID
	BSunits	Name and ID Area covered by each major habitat Map polygon
	ICunits	Name and ID Indicator List of BSunits
	NIunits	Name and ID List of BSunits
	Major habitats	Name and ID
	Years	Name and ID

Table 2.1 continued.

	Object type	Attribute
Objects that store results from index calculations	Indices	Name and ID Type of index Description List of indicators
	Nature- and thematic in-dex values	ID Index Calculation run NIunit Major habitat (Nature Index values) Year Point estimate 95% confidence interval
	Calculation runs	Name and ID Index Date and time Number of bootstrap simulations Uncertainty in reference values Uncertainty in indicator measurements Missing value treatment Description
Objects not involved in index calculations	Definition area	ID Indicator List of BSunits
	Predefined spatial units	ID and name List of BSunits
	Taxonomic groups	Name and ID
	Red list categories	Name and ID
	Experts	Name and ID Email Institution List of indicators Login information

Reference values have almost the same properties as indicator measurements. However, they are not related to a particular year. A reference value applies to a specific ICunit and has the same unit of measurement as indicator measurements from the same area.

All spatial units should have an ID and/or name. *Basic spatial units* should in addition be characterized by the area covered by each major habitat. The Norwegian implementation of the Nature Index uses municipalities as of 01.01.2010 as basic units. One must also provide the delineation of all *IC- and NIunits* in terms of the basic spatial units they consist of. In the Norwegian implementation, one has so far calculated the Nature Index for country regions and in some cases for counties, in addition to the whole country. The municipality ID-codes implicitly give the delineation of these NIunits. Information about NIunits are therefore not stored separately in the Nature Index database.

There are two additional types of spatial units defined in the Nature Index database. A *definition area* is defined for each indicator and consists of all BSunits where it is possible to document the indicator with measurements. The definition area often corresponds to the indicator's geographical range. There is also a set of so-called *predefined spatial units* intended to aid the experts when they delineate ICunits on the area-page (chapter 4.1).

The database includes short lists with IDs and names of *major habitats*, *functional groups*, *scaling functions*, *probability distributions*, *years* covered with indicator measurements, *taxonomic groups*, and *red list categories*. The Norwegian implementation lists nine major habitats: *ocean bottom*, *ocean pelagic*, *coast bottom*, *coast pelagic*, *open lowland*, *mires and wetland*, *freshwater*, *forest*, and *mountain*; and eight functional groups: *specialist-* and *generalist top predators*, *specialist-* and *generalist intermediate predators*, *herbivores* (including filter feeders), *specialist-* and *generalist primary producers*, and *decomposers* in addition to key indicators. Major habitats and functional groups may be merged during calculation of the Nature Index (e.g. Jakobsson & Pedersen 2020). There are two levels of informal, taxonomic groupings. Indicators based on single species are grouped into *plants* (with the subgroups *algae*, *liverwort*, *moss*, *fern*, and *seed plant*), *fungi* (*ascomycete*, *basidiomycete*, *lichen*), *invertebrates* (*cnidarian*, *crustacean*, *echinoderm*, *insect*, *mollusk*, *spider*, *sponge*, *other*), and *vertebrates* (*amphibian*, *bird*, *bony fish*, *cartilaginous fish*, *mammal*, *reptile*).

Each *index* is characterized by the selection of indicators that forms the index, together with an explanation of its background, motivation and/or interpretation. The index *calculations* themselves are characterized through the way missing values are handled, the number of bootstrap simulations, whether uncertainties in reference values and indicator measurements are accounted for, date and time of calculation, together with an explanatory text. Results from the calculations consist of estimates of *index values* for each year, each major habitat, and each NIunit covered, together with estimates of their 95% confidence intervals (chapter 2.1.7).

The public web portal of Nature Index uses different map polygons to visualize the state of indicators, ecosystems and thematic indices. For ecosystems and thematic indices, the polygons used are the Norwegian mainland regions, pelagic and coastal areas i.e. *Norwegian Sea, Barents Sea, North Sea, Skagerrak, Eastern Norway, Southern Norway, Mid-Norway, Northern Norway, Western Norway, coastal Eastern Norway, coastal Southern Norway, coastal Mid-Norway, coastal Northern Norway, coastal Western Norway.* For the indicators, the smallest unit used is the Norwegian municipalities. Other map polygons used are the former Norwegian counties.

The database contains information about *experts,* which are the users responsible for the different indicators and who enters indicator measurements and metadata for each indicator to the database. The system stores information about the expert's name, email and the institution that the expert belongs to. It also says which expert has access to, and is responsible for the different indicators.

3 Nature Index database

The Nature Index database is an SQL relational database. The database is of type MS SQL Server 2019 (https://www.microsoft.com/en-us/sql-server/sql-server-2019) and runs on MS Windows Server 2016 (https://www.microsoft.com/en-us/cloud-platform/windows-server).

The database consists of a set of main tables containing data on the most important object types for which information is stored in the database. Further, it consists of several lookup tables that contain mostly information and names of the features that characterize the objects in the main tables, such as tables about sampling time, major ecosystem, trophic group, Red List categories, type of probability distribution, data type, and scaling model. All objects in the main and lookup tables are assigned unique ID codes

The database also consists of tables that link information in the other tables via the objects' ID codes, and that also include information that characterizes the connection. For example, the link table "Indikator_Okosystem", which connects the main table "Indicator" and the lookup table "T_Okosystem" contains data on the indicator's connections to the various major ecosystems.

The database consists of many tables, some of them are no longer in use. In the following, the description of the tables is grouped in two parts;

- the tables containing input data from experts, municipality data and other background information. Information in these tables is used in the calculation of the NI.
- the tables used to store the result of the NI calculations. The tables are updated and maintained from R.

3.1 The Nature Index Main Tables

Figure 3.1 gives an overview over the most important tables in the Nature Index database and the relationships between them.

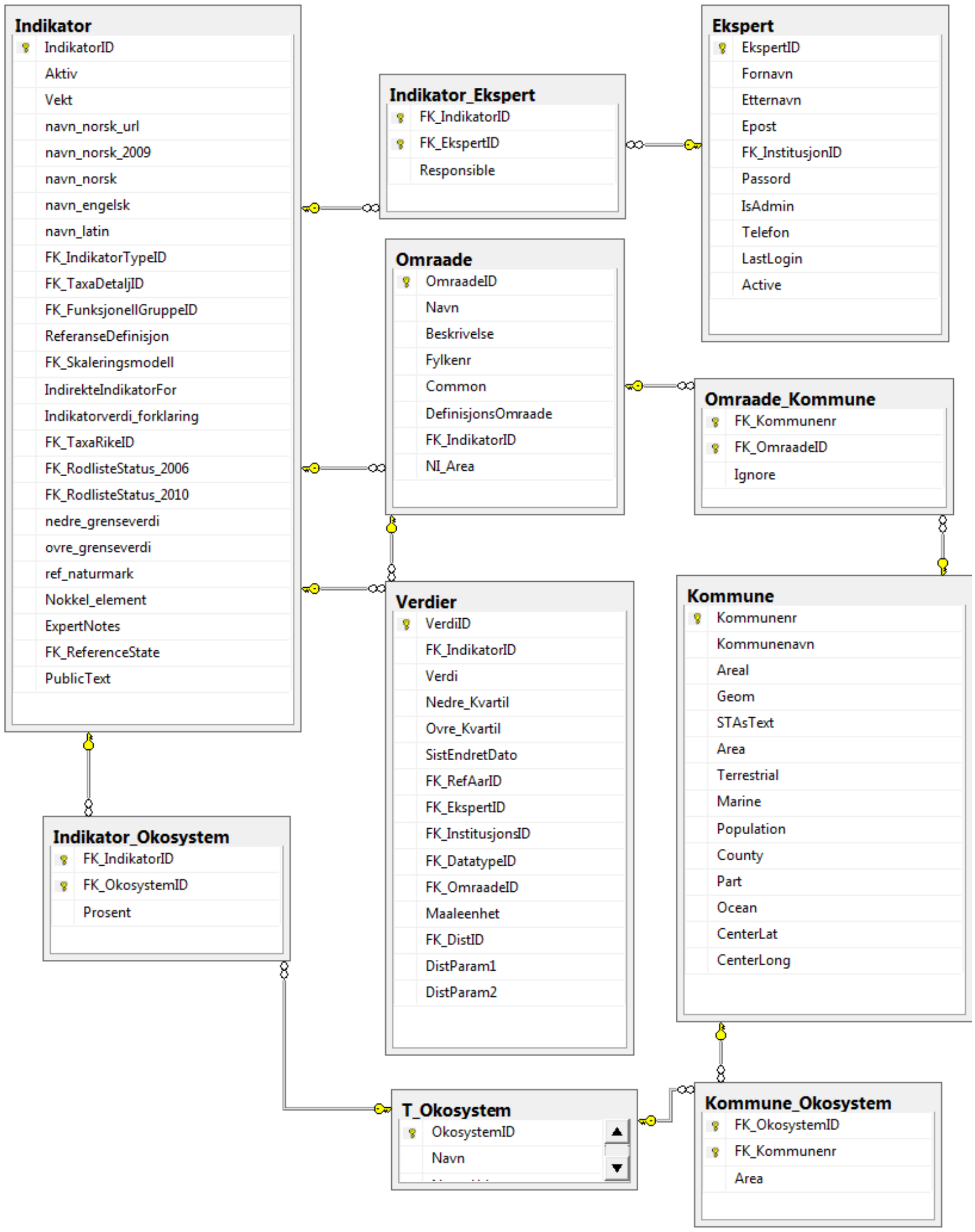


Figure 3.1: The most important tables storing input data from experts, municipality data and other background information.

Table 3.1. A brief overview of all tables in the Nature Index database and description of content.

Name of table (NOR/ENG)	Purpose
Area	Historical data, formerly used in index calculations
Changelog	Changelog messages displayed on the front page of the system.
Dokumentasjon/ Documentation	Metadata about documents that are uploaded via the input interface are stored here. The physical documents and pictures are stored in a dedicated file area on the web server or in the database.
Ekspert/ Expert	Contains metadata about experts.
FellesOmraade_Kommune	Table used for input to calculations in R
FellesOmraade	Table used for input to calculations in R
FileUpload	This table is used to store files when e.g. Nature Index is deployed in Azure.
Indikator/ Indicator	Contains metadata about the indicators.
Indikator_Datatype	Specifies the indicator's datatype(s)
Indikator_Ekspert/ Indicator_Expert	Information that governs experts' access to indicator data via the input interface.
Indikator_Okosystem/ Indicator_Ecosystem	Contains data on the indicator's "fidelity" to major ecosystems.
Insektdata	Historical data
Institusjon/ Institution	Contains metadata about the institutions.
Kommune/Municipality	The geographical polygons for Norwegian municipalities (as of 01.05.2010) and ocean areas.
Kommune_Okosystem/Municipality_Ecosystem	Areal coverage of major ecosystems within each municipality or the individual body of water is stored in this connection table.
Language	Translations of texts
NI_indicators	Result table R calculations (chapter 3.2)
NI_results	Result table R calculations (chapter 3.2)
NI_runs	Result table R calculations (chapter 3.2)
NI_thematicindexes	Result table R calculations (chapter 3.2)
NI_verdier_2012	Result table R calculations (chapter 3.2)
Omraade/Area	Contains names and IDs for ICunits and the IDs of the corresponding indicators.
Omraade_Kommune/ Area_Municipality	The municipalities included in each ICunit.
Parameters	Values controlling the different installations of the system, i.e. title, country, values for maps, store documents in database or on disc etc.
Publ_Kommune/ Publ_Municipality	Public municipalities of Norway with polygons and municipality number.
Paavirkning_Indikator/ Driver_Indicator	Contains the indicators' sensitivity to antropogenic pressures.

Table 3.1. Continued.

Name of table (NOR/ENG)	Purpose
T_Datatype	Lookup table for the different datatypes.
T_Distributions	Lookup table for the set of model probability distribution families fitted to indicator measurements in the elicitation process.
T_FunksjonellGruppe/ T_FunctionalGroup	Lookup table, functional group.
T_IndikatorType/ T_IndicatorType	Lookup table for indicator type.
T_NI_RunType	Lookup table for index types.
T_Okosystem/ T_Ecosystem	Lookup table, ecosystems.
T_Paavirkningsfaktorer/ T_Antropogenic_drivers	Lookup table over antropogenic pressures
T_PaavirkningsVerdi/T_Driver_effect	Lookup table over categories for indicators' sensitivity to antropogenic pressures
T_ReferenceState	Lookup table, reference state (habitat type)
T_RefAar	Lookup table for coding of reference values and of indicator measurements by year.
T_Rodlistestatus/ T_Redlist category	Lookup table, redlist category.
T_Skaleringsmodell/ T_Scaling model	Lookup table, scaling models.
TaxaDetalj/ Type of organism (informal subgrouping)	Lookup table, type of organism (informal subgrouping).
TaxaRike/ Type of organism (informal grouping)	Lookup table, type of organism (informal grouping).
TemaIndeks/ ThematicIndex	Contains names and descriptions of the different thematic indices.
Temaindeks_indikator/ ThematicIndex_Indicator	The indicators included in the different thematic indices.
Todo	Contain todo tasks for admin
Verdier/Values	The table consists of one object for each indicator measurement and reference value. In addition to the measurements' expected values, lower and upper quartiles, and/or distribution objects (chapter 2.1.4), the table also includes the indicator's ID, the refaarID (see table T_RefAar above), the area ID, data type (expert assessment, model-based or monitoring data) and measurement unit. Information about the probability distribution that is fitted to some of the values (cf. chapter 2.1.4), such as the type of distribution and the distribution's parameter values, are also stored in this table.

3.2 The result tables from R calculations

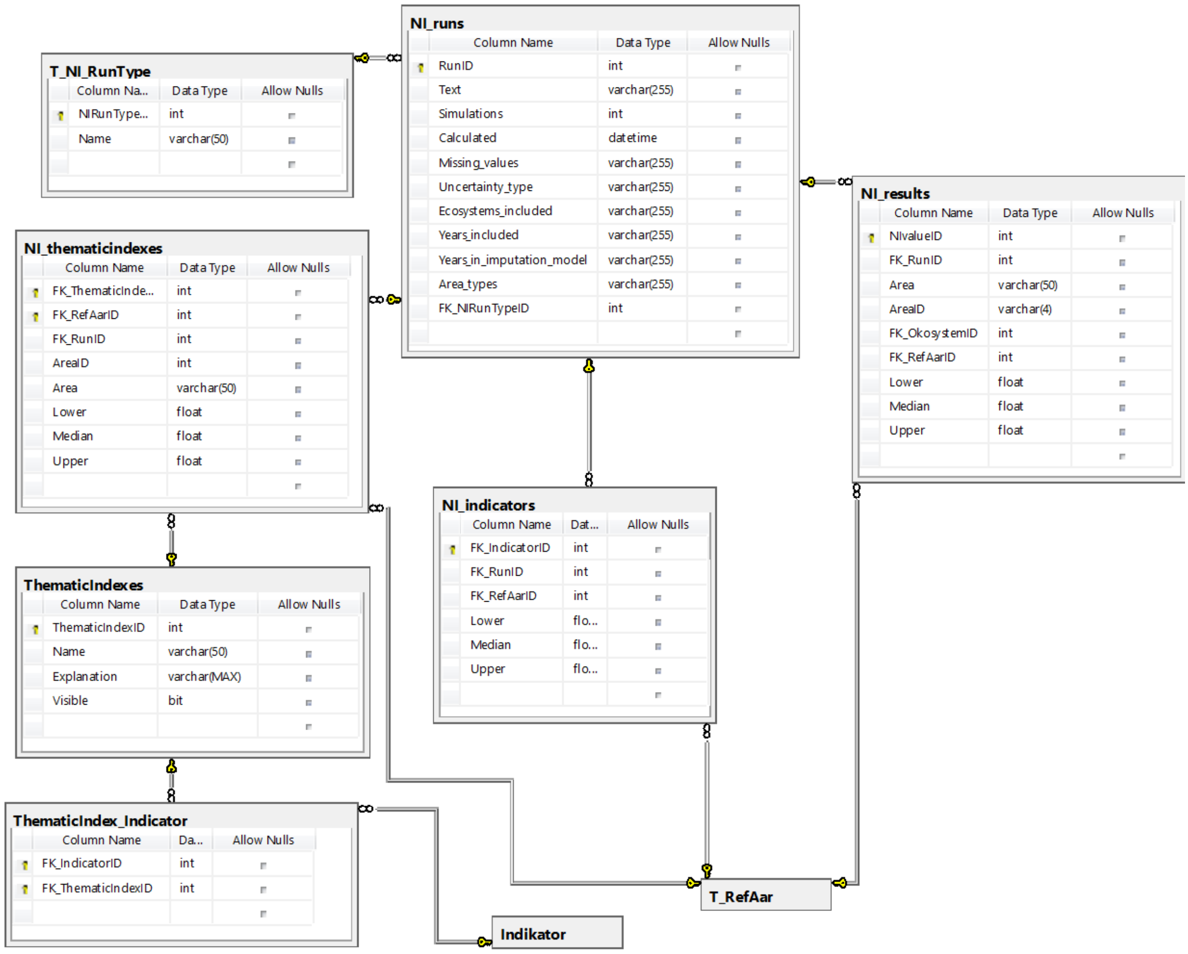


Figure 3.2 The most important tables storing results from index calculations.

Results from calculations of the Nature Index, thematic indices and national trend-lines for individual indicators from R are also stored in the database. They are stored in the tables NI_results and NI_thematicindexes and NI_indicators. For example, the table "NI_results" contains the calculated Nature Index values as point estimates and confidence intervals, and information about the year, the major ecosystem and area the values apply to, as well as a run_ID.

There are several additional tables that contain “bookkeeping information” about the calculations. For example, the table "NI_runs", contains background information and technical details describing the scope and conditions for the various calculations. All these tables – both those storing results and those storing bookkeeping info are updated and maintained by R-scripts that read results to the database. They may also, of course, be accessed directly.

Table 3.2 A brief overview of tables in the Nature Index database storing results from index calculations.

Name of table	Purpose
NI_results	Results from calculations of the Nature Index for ecosystems
NI_thematicindexes	Results from calculations of thematic indices (cf. chapter 2.2).
NI_indicators	Results from calculations of indicator indices (cf. chapter 2.2).
NI_runs	Contains background information, parameter values and other technical details describing the scope and premises for the various index calculations.
T_NI_RunType	Lookup table for index types (chapter 2.2).

3.3 Functions and stored procedures

The database has several stored procedures and some functions that provides functionality to the web applications. They are listed in tables 3.3 and 3.4.

Table 3.3 Stored Procedures in the Nature Index database

Norweagian name	English name
Get_Definisjonsomrade	Get definition area.
Get_kommuner_for_omrade	Get municipalities for area.
Get_ledige_kommuner_for_definisjonsomrade	Get available municipalities for definition area.
Get_ledige_kommuner_for_omrade	Get available municipalities for area.
Get_omraader	Get areas.
SP_Delete_Indikator_For_Ekspert	Delete indicator for expert.
Get_DB_Name	Get_DB_Name.
SP_Get_EcosystemAreas_For_Kommune	Get ecosystem areas for municipalities.
SP_Get_Ecosystemvalues_For_Indicator	SP_Get_Ecosystemvalues_For_Indicator.
SP_Get_Indikatorer_For_Ekspert	Get indicators for expert.
SP_Get_Omraade	Get area.
SP_Get_Omraade_As_Geography	Get area as geography.
SP_Get_Pressures_For_Indicator	Get pressures for indicator.
SP_Get_RefAar_Med_Verdi_For_Indikator	Get years with values for indicator.
SP_Get_Values_For_Indicator_Area	Get values for indicator area.
SP_Get_Verdier_Omraader	Get values for areas.
SP_Region_Kommuner	Region municipalities.
SP_Public_GetKommuner	Get municipalities.

Table 3.4 Functions in the Nature Index database

Norwegian name	English
CamelCase	Camel case.
GetOmradeGeoAsText	Get area geometry as text.
GetReferanseverdiForOmrade	Get reference value for area.

3.4 Database content

Table 3.5 Content of the Nature Index database per 15.11.2020. Active indicators are those included in the last update of the Nature Index (NI2020). In addition to active indicators, the database stores data for 134 additional indicators included in earlier updates of the index.

Objects	#
Active indicators	260
ICunits (active indicators)	11 120
Institutions	6
Indicator observations	116 889
Documents, images, links to external resources	1 271

Table 3.6 Number of active indicators for each ecosystem per 15.11.2020. Some indicators are included in two or more ecosystems.

Ecosystem	Number of indicators
Coast	46
Freshwater	38
Mountain	30
Ocean	28
Open lowland	22
Wetland	28
Woodland	89

4 Web-interface for data entry

4.1 Functionality – a brief overview

The website https://naturindeks.nina.no is a data entry portal where experts can register and edit observations and other information about the indicators that the expert is responsible for. It is also possible to upload documents, images and links to external resources. The information provided in the data entry portal will be available for everyone in the public web solution https://naturindeks.no, which will be described in Chapter 5. The experts must log in with a username and a password and will then be given access only to the indicators that they are responsible for. The data entry portal can be opened and closed for data editing by an administrator.

The system consists of six main pages:

The front page contains information about new versions and other important messages to the users. A link to the user manual for the web portal (Pedersen & Kvaløy 2014) is available here.

The indicator page is used to select indicator and to enter different data about it. The different fields and functions are described in detail in the *User Manual, Chapter 3*.

The area page is used to define and edit ICunits (chapter 2.1.3), the geographical areas indicator measurements refers to. Details about the area page can be found in the *User Manual, Chapter 4*.

The value page contains tools for entering indicator- and reference values to the database either directly on the page or by importing the data from Excel work-sheets. There is also a tool for importing data from Excel-files, a tool for visualizing probability distributions fitted to measurements and reference values by elicitation (cf. chapter 2.1.4), and a proofreading tool that visualizes stored data on a map. Explanation on how the page works can be found in the *User Manual, Chapter 5.*

The user page contains functions for editing user information such as name and contact information. The page is further described in the *User Manual, Chapter 2.*

The admin page is only available for users with administrative privileges. On this page, users, indicators and municipalities can be administered and the portal can be opened or closed for the editing of data for different years.

4.2 Technologies, frameworks and code organization

4.2.1 Technologies and frameworks

Server side:

- Database: Microsoft SQL Server installed on a Microsoft Windows Server 2016 platform (references in Chapter 3).

- Webserver: Microsoft Internet Information Server (IIS) 10 (https://www.iis.net) running on a Windows Server 2019 platform (Windows Server 2019 | Microsoft).
- Framework: ASP.NET Framework 4.5 (https://www.microsoft.com/en-us/download/details.aspx?id=30653) and uses .NET .aspx master/content pages (https://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx).
- Programming languages: C# (https://docs.microsoft.com/en-us/dotnet/csharp)
- O/RM: Entity Framework is used as an abstraction layer between code and data-base (https://docs.microsoft.com/en-us/ef/ef6)
- The data is retrieved from the database by Linq (https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/language-reference/linq-to-entities). Linq is the query language for Entity Framework.
- An Entity Model Code Generator (https://docs.microsoft.com/en-us/ef/ef6/modeling/designer/codegen) is used to generate the objects representing the tables in the database and thereby the domain objects. The code generation can be managed from the file *Naturindeks.edmx*

Client side:

- Programming languages:
  - CSS and HTML5 (https://www.w3.org/standards/webdesign/htmlcss)
  - JavaScript (https://www.w3schools.com/whatis/whatis_js.asp)
- JavaScript framework: Knockout (https://knockoutjs.com/)
- Data interchange format: Data is transported on the Java Script Object Notation - JSON format (https://www.json.org) between client and server with help of Web API (https://www.asp.net/web-api)
- Map engine: Leaflet (https://leafletjs.com) is used to display the web maps. The Wicket.js framework (https://github.com/arthur-e/Wicket) is used to read and write the geometric objects to and from Leaflet.

4.2.2 Code organization

The server-side code is mainly organized in different Web API Controllers (https://www.asp.net/web-api) which handle the different requests from the client. The necessary logic is then executed, and the result is returned to the client as JSON or as a page view with the relevant information. The data is retrieved from the database either by Linq-queries in code or by calling stored procedures in the database. Most of the requests for different polygons used to draw municipalities or indicator area layers on the map rely upon stored procedures. Table 4.1 shows the different endpoints and the functionality associated with it.

Table 4.1. Endpoints and functionality

Endpoint	Functionality
Admin/Eksperter	Get all registered experts
Admin/Institusjon	Get all registered institutions
Admin/Indicators	Get all indicators
Admin/Municipalities	Get all municipalities in Norway
Admin/GetAreasForMunicipality/kommunenr	Get ecosystem areas for selected municipality
Admin/SaveEcosystemAreasToDB/ kommunenr	Saves or updates ecosystem areas for selected municipality
Admin/GetEcosystems	Get all ecosystems
Admin/GetRefAarValues	Get id for reference values and ids for all years
Admin/UpdateRefAarValues	Update ids for reference values and all years
Admin/SaveExpertToDB	Save new expert or update existing expert
Admin/GetExpertsForIndicator/indikatorID	Get experts for selected indicator
Admin/GetIndicatorsForExpert/EkspertID	Get indicators for selected expert
Admin/UpdateIndicatorsForExpert/EkspertID	Update indicator for selected expert
Admin/SaveIndicatorToDB	Save new indicator or update existing indicator.
Omraade	Get all ICunits
Omraade/Id	Get ICunit by id
Omraade/Common	Get predefined (default) areas
Omraade/GetCommonKommuneList/AreaIds	Get municipality info and polygons for one or more ICunits
Omraade/omraadeID/kommuner	Get municipalities, municipality info, and polygons for selected ICunit
Omraade/omraadeID/STAsText	Get info and polygons for selected ICunit
Omraade/indikatorID/def	Get definition area info and polygons for selected indicator
Omraade/indikatorID/all	Get all ICunits with info and polygons for selected indicator
Omraade/indikatorID/ LedigeKommunerForDefinisjonsomraade	Get free municipalities for definition area for selected indicator
Omraade/indikatorID/ LedigeKommunerForOmraade	Get free municipalities for ICunit for selected indicator
Omraade/indikatorID/CreateNewArea/ AreaName	Create new area (either definition area or ICunit) for selected indicator
Omraade/OmraadeID/AddToOmraade	Add municipalities to existing ICunit
Omraade/OmraadeID/RemoveFromOmraade	Remove municipalities from existing ICunit
Omraade/OmraadeID/RenameOmraade	Rename ICunit
Omraade/OmraadeID/Delete	Delete ICunit

Table 4.1. Continued

Endpoint	Functionality
Default/GetDBName	Get database name
Default/Changelog	Get the changelog
Default/Todo	Get todos
Default/Changelog/New	Create new changelog entry
Default/Todo/Save	Create new todo entry
Default/Changelog/Delete/Id	Delete changelog entry by id
Default/Todo/Delete/Id	Delete todo entry by id
Documentation/GetDocument/id	Get document from database by id
Excel/Id	Get all data as excel for selected indicator
Excel/Upload/id	Upload and update data from excel for selected indicator
Indicator	Get list of all indicators
Indicator/User/Id	Get indicators for selected user
Indicator/Id	Get indicator by id
Indicator/T_IndikatorType	Get indicator types
Indicator/T_RefAar	Get id for years and for reference values
Indicator/T_FunksjonellGruppe	Get functional groups
Indicator/TaxaRike	Get types of organism (informal grouping)
Indicator/TaxaDetalj	Get types of organism (informal subgrouping)
Indicator/TaxaDetalj/{TaxaRikeID:int}	Get type of organism (informal subgrouping) for selected type of organism (informal grouping)
Indicator/T_Rodlistestatus	Get red list statuses
Indicator/T_ReferenceState	Get references states
Indicator/T_Skaleringsmodell	Get scaling models
Indicator/indikatorID/RefAarMedVerdi	Get years with values for selected indicator
Indicator/indikatorID/Okosystemer	Get ecosystems for selected indicator
Indicator/indikatorID/Pressures	Get pressures for selected indicator
Indicator/indikatorID/UpdatePressures	Update pressures for selected indicator
Indicator/indikatorID/UpdateEcosystems	Update ecosystems for selected indicator
Indicator/indikatorID/Documentation	Get documentation for selected indicator
Indicator/indikatorID/UpdatePublicText	Update public text for selected indicator
Indicator/UpdateDocumentation	Update documentations for indicator
Indicator/DeleteDocumentation/docid	Delete documentation by id
Indicator/T_Paavirkningsfaktorer	Get name of all pressures
Indicator/T_PaavirkningsVerdi	Get all values for pressures
Indicator/UpdateIndicator	Save or update indicator
Login	Login user
Parameters/Getvalue/Name	Get value for given parameter

Table 4.1. Continued

Endpoint	Functionality
Statistikk/GetAntVerdier	Get number of values in the NI database
Statistikk/GetAntOmrader	Get number of ICunits in the NI database
Statistikk/GetAntEkspert	Get number of experts in the NI database
Statistikk/GetAntInstitusjon	Get number of institutions in the NI database
Statistikk/GetAntIndikator	Get number of indicators in the NI database
Statistikk/GetAntVerdierOvervaking	Get number of values from monitoring in the NI database
Statistikk/GetAntVerdierEkspert	Get number of values from expert judgment in the NI database
Statistikk/GetAntVerdierModeller	Get number of values from models in the NI database
Statistikk/GetAntIndikatorerPrOkosystem	Get number of indicators per ecosystem in the NI database
Upload/File	Upload file to database or file system
Upload//ExcelFile	
User/GetInstitutions	Get all institutions
User/GetUser/userid	Get user by id
User/UpdateUser/userid	Update information for selected user
User/UpdatePassword/userid/newpwd	Update password for selected user
Values	Get all values
Values/DataTypes	Get datatypes
Values/RefAar	Get id for years and for reference values
Values/indikatorID/def	Get definition area for selected indicator
Values/indikatorID/all	Get all areas with info and polygons for selected indicator
Values/IndikatorID/RefAarID	Get all values, info and polygons for selected indicator and selected year
Values/id	Get values for selected indicator
Values/indikatorid/Omraade/omraadeid	Get values for selected indicator and selected area
Values/SaveValues	Save values for selected indicator and selected area

5 Public Nature Index web-site

5.1 Background

The public Nature Index website is available at www.naturindeks.no/. The goal of the website is to provide users and the public at large insight into what the Nature Index is, the purpose it is meant to serve, how it is calculated and what kind of data it is based on. The website provides information about – and hopefully creates interest in – the state of biodiversity in Norway, how this state has developed in the recent past and what is known and not known about these questions.

The web portal is included as part of the Norwegian Environment Agency's online website directed at the public. The site layout therefore follows the same template as the Agency's other web sites. The website largely replaces previous public outreach efforts about the Nature Index and its underlying data in the form of reports.

As for the web interface for data entry (previous chapter), the website is developed and hosted by the Norwegian Institute for Nature Research. It's first version was completed in the autumn of 2015.

The web portal is developed with a so-called responsive design, so that it can be viewed on all types of devices such as mobile phones, tablets, PCs and Macs. The solution is designed to support multiple languages, and currently the web portal is available in English and Norwegian.

5.2 The pages

The web portal consists of seven main pages:

The **home page** contains a general introduction to the Nature Index.

The **species / indicators** page contains general descriptions of each indicator, their underlying data and procedure for setting reference values. The page also shows a map of each indicator that depicts scaled indicator values for the years for which there are data, and a graph showing the trend of the indicator's state at a national level (with confidence intervals). The page also contains a list of anthropogenic factors that put negative pressure on the indicators, links to background documents, and contact details for the experts that are responsible for the indicators.

The **ecosystems** page presents the index values for each major ecosystem in maps and graphs (with confidence intervals) in the period from 1990 until the present day. The maps portray index values for each region and ocean area.

The **thematic indices** page presents thematic indices in text, graphs and maps for the whole country as well as each region.

The page **about the Nature Index** introduces the Nature Index framework and how it is calculated.

The **key numbers** page for the Nature Index contain number of indicators in each major ecosystem.

The **pressures** page discusses how anthropogenic pressures might affect the Nature Index in each major ecosystem.

5.3 Technologies and frameworks

Server side:

- Database: Microsoft SQL Server installed on a Microsoft Windows Server 2016 platform (references in Chapter 3).
- Webserver: Microsoft Internet Information Server (IIS) 10 (https://www.iis.net) running on a Windows Server 2019 platform (Windows Server 2019 | Microsoft).
- Programming languages: C# (https://docs.microsoft.com/en-us/dotnet/csharp)
- O/RM: Entity Framework is used as an abstraction layer between code and data-base (https://docs.microsoft.com/en-us/ef/ef6)
- The data is retrieved from the database by Linq (https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/language-reference/linq-to-entities). Linq is the query language for Entity Framework.
- An Entity Model Code Generator (https://docs.microsoft.com/en-us/ef/ef6/modeling/designer/codegen) is used to generate the objects represent-ing the tables in the database and thereby the domain objects. The code genera-tion can be managed from the file *Naturindeks.edmx*

Client side:

- Programming languages:
  - CSS and HTML5 (https://www.w3.org/standards/webdesign/htmlcss)
  - JavaScript (https://www.w3schools.com/whatis/whatis_js.asp)
- JavaScript framework: Knockout (https://knockoutjs.com/)
- Data interchange format: Data is transported on the Java Script Object Notation - JSON format (https://www.json.org) between client and server with help of Web API (https://www.asp.net/web-api)
- Map engine: Leaflet (https://leafletjs.com) is used to display the web maps. The Wicket.js framework (https://github.com/arthur-e/Wicket) is used to read and write the geometric objects to and from Leaflet.
- Highcharts (https://www.highcharts.com/) is the chosen framework for displaying interactive charts.

Miscellaneous:

- The geometry used to draw on the map is either found in the regions.js-file as GeoJson (http://geojson.org) for Ecosystem and Thematic Indices, or in the database for Indicators.
- The different polygons for drawing areas such as municipalities and ICunits are returned from server to the client as Well-Known Text (WKT). WKT is an Open Geospatial Consortium (OGC) representation of a geography instance.
- The solution has support for multiple languages. All texts are replaced by variables. The name of the variables and the corresponding translations are stored in resource files. English and Norwegian are the currently available languages.

5.4 Code organization

The server-side code is mainly organized in different controller classes which handle the different requests from the client. The necessary logic is then executed, and the result is returned to the client as JSON or as a page view with the relevant information. Table 5.1 shows the different endpoints and the functionality associated with it.

Table 5.1. Endpoints and functionality

Endpoint	Functionality
Indicator/AllInfo/indicatorid	Get indicator index values, indicator observations for charts, indicator meta information, experts, documentation and images for selected indicator.
Indicator/IndikatorMap/indikatorID	Get ICunits as polygons, indicator observations for all years and information about selected indicator to show in map.
Indicator	Get list of all active indicators
Indicator/Id	Get selected indicator
Indicator/TaxaDetalj	Get TaxaDetalj information for selected indicator.
Indicator/ T_Rodlistestatus	Get red list status for selected indicator
Indicator/IndikatorID/RefAarID	Get indicator observations for ICunits, and ICunits as polygons for the selected indicator and selected year
Indicator/indikatorID/Okosystemer	Get ecosystems for selected indicator
Indicator/indikatorID/Paavirkninger	Get pressure factors for selected indicator
Indicator /indikatorID/Datatyper	Get datatypes for selected indicator
Indicator /GetEcosystems	Get all ecosystems
Indicator/GetIndicatorsForEcosystem/ ecosystemid	Get all indicators for selected ecosystem
Indicators/name	Get indicator page by name
Indicators/id	Get indicator page by id

Table 5.1. Continued

Endpoint	Functionality
Ecosystem/GetEcosystems	Get list of all ecosystems
Ecosystem/GetAreas/ecosystemid	Get names and ids of NIunits for selected ecosystem.
Ecosystem/GetChartValues/areacode/ ecosystemid	Get NI values for chart for selected NIunit and ecosystem.
Ecosystem/GetFillValues/OkosystemID	Get NI values displayed as colours in map for selected ecosystem.
Ecosystems/name	Get ecosystem page by name.
Themes/id?	Get all thematic indices or information about selected thematic indices
Themes/GetAreas/id	Get NIunits for selected thematic indices
Themes/GetFillValues/id	Get index values displayed as colours in map for selected thematic indices.
Themes/GetChartValues/areacode/themeId	Get index values for chart for selected area and thematic indices.
Themes/GetExplanationThemaIndex/id	Get textual description of selected thematic indices
Themes/GetIndicatorsForThematicIndex/id	Get all indicators for selected thematic indi-ces
Statistikk/GetAntVerdier	Get number of values in NI database
Statistikk/GetAntOmrader	Get number of ICunits in NI database
Statistikk/GetAntEkspert	Get number of experts registered in the NI database
Statistikk/GetAntInstitusjon	Get number of institutions registered in the NI database
Statistikk/GetAntIndikator	Get number of indicators in NI database
Statistikk/GetAntVerdierOvervaking	Get number of observations that are based on monitoring.
Statistikk/GetAntVerdierEkspert	Get number of observations that are expert based
Statistikk/GetAntVerdierModeller	Get number of observations that are model based
Statistikk/GetAntIndikatorerPrOkosystem	Get number of indicators per ecosystem in the NI database
Statistikk/GetAntIndikatorerPrOkosystem/ OkosystemID	Get number of indicators for selected eco-system.
IndexMap/Id	Get the map from naturindeks.no for the se-lected ecosystem. Can be shown in iframe on another web-page (ie. Miljøstatus)

Table 5.1. Continued

Endpoint	Functionality
IndexMap/Indicator/Id	Get the map from naturindeks.no for the selected indicator. Can be shown in iframe on another web-page (ie. Miljøstatus)
IndexMap/ThematicIndices/Id	Get the map from naturindeks.no for the selected thematic indices. Can be shown in iframe on another web-page (ie. Miljøstatus)
IndexChart/Id	Get the chart from naturindeks.no for the selected ecosystem. Can be shown in iframe on another web-page (ie. Miljøstatus)
IndexChart/Indicator/Id	Get the chart from naturindeks.no for the selected indicator. Can be shown in iframe on another web-page (i.e. Miljøstatus)
IndexChart/ThematicIndices/Id	Get the chart from naturindeks.no for the selected thematic indices. Can be shown in iframe on another web-page (ie. Miljøstatus)
Home/SetCulture	Set the selected language for the site.
Kontakt	Returns page with contact information.
Keynumbers	Returns page with key numbers and statistic for the NI database.

6 R-scripts and NIcalc

Scripts for calculation of Nature-, thematic- and indicator indices are coded in R (appendix 1). Data are imported from the Nature Index database through the Nature Index Application Programming Interface described in Chapter 7. Results from calculations are exported to the database through an ODBC interface. R code for posting the results via the Nature Index API remains to be developed.

Additional scripts perform a series of tasks within the Nature Index IT information system. For example, the tool for visualizing probability distributions (chapter 4.1) runs an R code that does the calculations and draws the final figure. Experts may also develop their own R-scripts for entering indicator measurements to the database via the API.

Such scripts use a number of utility functions tailored for the various tasks within the Nature Index IT information system. These functions are bundled together in an R-package called *NIcalc* (table 6.1). *NIcalc* represents a complete revision and extension of earlier published R-scripts (Certain *et* al. 2011) for calculating the Nature Index.

There are several aspects of the Nature Index framework that makes it necessary to code functions and define object classes that are specifically tailored for entering data to the Nature Index database, for calculation of indices, and for preparing input data to such calculations, instead of just using general purpose functions already available within R. First, in the context of the mathematical framework, indicator measurements are probability distributions. The NI-system is somewhat flexible in how distributions are entered to and stored in the database in order to facilitate the inclusion of both expert judgments and model predictions (chapter 2.1.4 and 2.3). R code should accommodate and account for all the formats applied for specifying such distributions.

Second, in order to calculate the Nature Index, characteristics of indicators, indicator measurements, reference values, as well as BS-, IC- and NIunits must be provided. The relevant data are found distributed over several tables in the Nature Index database. The information is not easily combined into simple data structures or objects. *NIcalc* therefore contains a set of utility functions especially adapted to harvest the Nature Index database. These functions return data objects structured according to the requirements set by the framework for calculating the index. They include procedures for checking whether candidate data sets are consistent and contain all the necessary information.

Third, calculation of weights requires an analysis of the structure of input datasets (cf. chapter 2.1.6). Fourth, the flexible system of defining indicator specific sets of ICunits (chapter 2.1.3) entails the maintenance of a bookkeeping system for such units during the calculations. Such a system is required not only for selecting indicator measurements from a particular NIunit, but also to avoid underestimating the uncertainty in indices for NIunits covering two or more BSunits.

Table 6.1 lists and describes the functions included in *NIcalc*. S3 classes defined within the package and their methods are listed in appendix 2 and table 6.1. The functions *importDatasetApi()*, *assembleNiObject()*, *calculateIndex()*, *getIndicatorValues()*, *setIndicatorValues()*, and *writeIndicatorValues()* represent the core of *NIcalc*.

getIndicatorValues(), *setIndicatorValues()*, and *writeIndicatorValues()* may be used in scripts developed by the experts to update the NI database with new or revised measurements for the indicators they are responsible for. *getIndicatorValues()* retrieves the current values for a given indicator from the NI database as an S3 object of class *indicatorData* (appendix 2), *setIndicatorValues()* updates *indicatorData* objects with new indicator measurements, and *writeIndicatorValues()* posts objects with updated values to the 'Verdier' table via the Nature Index API.

importDatasetApi() harvests datasets from the database via the Nature Index API. It calls a series of functions that retrieves data from the specific tables in the database. *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. If necessary, it completes the elicitation process for expert opinions. *importDatasetApi()* and *assembleNiObject()* return S3 objects of class *niDataImport* and *niInput* respectively. These are comprehensive lists of several dataframes and matrices (appendix 2). *niInput* lists contain all the necessary information for calculating indices, including indicator observations, reference values, indicator- and BSunit characteristics for calculating weights, as well as the delineation of ICunits and NIunits in terms of BSunits.

calculateIndex() calculates the Nature Index, thematic indices, as well as indicator indices (chapter 2.2) from *niInput* lists. *calculateIndex()* does the whole procedure of sampling and scaling indicator observations, calculating weights and the weighted average. It calls a series of other functions from the *NIcalc* package; *sampleObsMat()* draws the bootstrap samples, *scaleObsMat()* does the scaling of indicator measurements, while *calculateWeights()* and *indexCalculation()* calculate weights and the weighted average respectively. Each of these calls yet another sets of functions within the package. *calculateIndex()* returns an object of class *niOutput* (appendix 2). It produces an extensive output for each index value to facilitate further analyses of the results.

Other functions central for implementing the Nature Index framework are *elicitation()* and *makeDistribution()*. *elicitation()* fits two-parameter distributions to indicator observations given as expected value and lower and upper quartiles, while *makeDistribution()* generates distribution objects from model predictions (chapter 2.1.4).

NIcalc contains documentation and explanations for all functions included. Three vignettes are also included: *NatureIndexCalculation* describes the mathematical framework (i.e. chapter 2.1.1 – 2.1.7), *objectsInNIcalc* describes the S3 classes defined within NIcalc (cf. appendix 2), while *Distributions* describes how experts may interact with the Nature Index database in their scripts when updating their indicators with new data, and how to work with distribution objects in this context.

Table 6.1 Functions in NIcalc

Function name	Explanation	Comment
Data entry to NI data-base		
getToken()	Get a connection token from the NI database	Tries to connect to the NI database API and retrieve a token that is later used to communicate with the database.
getIndicators()	List indicators	List the indicators that the user is per-mitted to alter
getIndicatorValues()	Get present indicator values	Retrieves the current values for a given indicator from the NI database as an indicatorData object.
makeDistribution()	Create a distribution object	Formats various representations of in-dicator uncertainty into a common structure for further processing
setIndicatorValues()	Set indicator values	Updates an indicatorData object with new indicator measurements.
writeIndicatorValues()	Write indicator values	Writes an indicatorData object with updated values for a given indicator to the Nature Index database via the Na-ture Index API.
Reading data from the NI database		
importDatasetApi()	Import dataset	Imports and assembles a candidate da-taset from the NI database via Nature Index API
getBSunits()	Reads the 'Kommune' table in the NI database via the Nature Index API.	Characteristics for all BSunits.
getEcosystems()	Reads the 'T_Okosystem' ta-ble.	Ecosystem names and ids.
getEcosystemB-SunitData()	Reads the 'Kommune_Okosys-tem' table.	Area of each major ecosystem per BSunit.

Table 6.1 Continued

Function name	Explanation	Comment
getFunction-alGroupData()	Reads the 'T_FunksjonellGruppe' table.	Names of functional groups
getICunits()	Reads the 'Omraade' table.	ICunits for each indicator
getICunitBSunit()	Reads the 'Kommune_Okosys-tem' table.	BSunits within each ICunit
getIndicators2()	Reads the 'Indikator' table.	Indicator characteristics.
getIndicatorEcosys-temData()	Reads the 'Indikator_Okosys-tem' table.	Fidelity data for each indicator per ma-jor ecosystem.
getIndicatorValues2()	Reads 'Verdier' table.	Reference values and indicator obser-vations
getRefYearData()	Reads the 'T_RefAar' table.	Ids for observation years and reference values.
getScalingModelData()	Reads the 'T_Skalerings-modell' table.	Names of scaling models
extractContentFromNi-api()	Extract data from the Nature Index API	A general purpose utility function that retrieves data from specified tables in the Nature Index database
assembleNiObject()	Assemble dataset	Assembles and structures data into a complete and consistent dataset for calculating the Nature Index or the-matic indices.
checkInputData()	Check dataset	Controls the content of a candidate da-taset for calculating the Nature Index.

Table 6.1 Continued

Function name	Explanation	Comment
Elicitation		
elicitation()	Fit probability distributions	Fits probability distributions to a set of indicator measurements.
estim()	Fit probability distribution	Selects, for both continuous and dis-crete cases and by using the least square criterion, the distribution among a predetermined set of model distribution families that best fits to an expected value and two quantiles
estimlight()	Fit probability distribution - reduced version	Called if estim() fails.
qdev	Sum of squared differences between distribution parame-ters	Set of functions for calculating sum of squared differences between "ob-served" parameter values and model distribution parameters: qdev.TNO, qdev.LOGNO, qdev.WEI, qdev.ZEXP, qdev.GA, qdev.PO, qdev.NBII, qdev.ZIP
Imputations for missing observations		
imputeDiagnostics()	Imputation diagnostics	Performs multivariate imputation by chained equations and returns a set of diagnostic plots and statistics together with the imputations.
imputeData()	Multiple imputations	Multivariate imputation by chained equations. Adds multiple imputations for missing indicator observations in a Nature Index dataset.
impStand()	Standardize imputations	Optional standardizations of imputa-tions

Table 6.1 Continued

Function name	Explanation	Comment
Index calculation		
calculateIndex()	Calculate index	Calculate the Nature Index for a major habitat, or calculate a thematic index. A time series of index values are calculated for each of one or more NIunits.
sampleobs()	Random draws	Random draws from two-parameter probability distributions.
sampleDistribution()	Random draws	Random draws from distribution objects.
sampleObsMat()	Draw bootstrap samples	Bootstrap samples of a set of distributions and distribution objects representing indicator observations or reference values.
scaleObsMat()	Scaling of indicator measurements	Scales a set of indicator observations according to chosen scaling model and associated reference values.
calculateWeights()	Weighting per NIunit	Returns weights for the calculation of the Nature Index for a set of NIunits.
calculateBSunitWeights()	Weighting per BSunit	Returns weights used in the calculation of indices for a set of BSunits.
calculateNIunitWeights()	NIunit weights	Returns weight factors used in the calculation of indices for a set of NIunits.
indexCalculation()	Weighted average per NIunit of scaled indicator observations	Calculates, for each unit in a set of NIunits, a sample of draws from the distribution of the Nature Index or a thematic index.
indexCalculationPerBSunit()	Weighted average per BSunit of scaled indicator observations	Calculates, for each unit in a set of BSunits, a sample of draws from the distribution of the Nature Index or a thematic index.

Table 6.1 Continued

Function name	Explanation	Comment
Classes and methods		
niDataImport() is.niDataImport()	Create or test for objects of class niDataImport	Appendix 2
niInput() is.niInput()	Create or test for objects of class niInput	Appendix 2
niImputations() is.niImputations()	Create or test for objects of class niImputations	Appendix 2
niOutput() is.niOutput()	Create or test for objects of class niOutput	Appendix 2
niSeries() is.niSeries()	Create or test for objects of class niSeries	Appendix 2
niValue() is.niValue()	Create or test for objects of class niValue	Appendix 2
plot.niSeries()	Plot method for class niSeries	Plots a time series of index values
plot.niValue()	Plot method for class niValue	Plots the distribution of an index
plotWeights()	Weight plot method for class niValue	Barplot showing weights per indicator, trophic group, BSunit or ICunit
summary.niOutput()	Summary for class niOutput	Summarizes index value estimates for a set of NIunits
summary.niSeries()	Summary for class niSeries	Summarizes a time series of index value estimates
summaryWeights()	Weight summary method for class niSeries	Summarizes weights per indicator, trophic group, BSunit or ICunit

Table 6.1 Continued

Function name	Explanation	Comment
Miscellaneous		
fitAndPlotDistribution()	Fit and plot distribution functions	Calls functions for elicitation with in-puts and plots the resulting distribution
plotDistribution()	Plot function for known distributions	
normal2Lognormal() logNormal2normal()		Transform normal distribution to log-normal, and vice versa

The open source library and its documentation is available under version 3 of the GNU General Public License at https://github.com/NINAnor/NIcalc and can be installed by the R command devtools::install_github("NINAnor/NIcalc", build_vignettes = T). *NIcalc* includes a trial dataset. It depends on a series of other R-packages: distr (Ruckdeschel *et al*. 2006), gamlss.dist (Stasinopoulos & Rigby 2021), graphics and grDevices (R Core Team 2020), httr (Wickham 2020), jsonlite (Ooms 2014), lattice (Sarkar 2008), mice (Van Buuren & Groothuis-Oudshoorn 2011), msm (Jackson 2011), plyr (Wickham 2011), stats (R Core Team 2020), tibble (Müller & Wickham 2021), truncnorm (Mersmann *et al*. 2018), uuid (Urbanek & Ts'o 2020). Code and documentation is authored by Bård Pedersen and Jens Åström. Functions *sampleobs*(), *qdev*(), *estim*() and *estimlight*() are revisions of code provided by Nigel Yoccos.

7 Nature Index API

The Nature Index Application Programming Interface (API) is a set of methods that provides import and export functionality from the NI database. Some of the methods in the API is open for the public and others requires authentication.

The open methods of the API can be used to read information about the different species groups (indicators), ecosystems and themes (thematic indices). The API provides metadata, the calculated Nature Index values and spatial data (GeoJson). The information available is the same information presented at the public Nature Index web-site, *naturindeks.no* (Chapter 5).

The API also offers methods to read from and write to the database about the indicators. These methods are only available for users who are experts in the Nature Index database. They can deliver their values for the indicators that they are responsible for. To authenticate themselves, experts must obtain a security token that must be sent in the header of every request to the API. The NI Calc package (Chapter 5) uses the API to facilitate a more automated data delivery process to the NI database.

There are also methods in the API that is used by the NI calc to calculate the NI values. These methods require both authentication and authorization; the user must have admin rights to use these API methods.

As of 01.01.2021, the Nature Index API offers the following methods:

TYPE	Endpoint	Auth
GET	/api/Ecosystem/All	
GET	api/Ecosystem/{id}	
GET	/api/Ecosystem/{id}/GetValues	
GET	/api/Ecosystem/{id}/Areas	
GET	/api/Ecosystem/Areas	
GET	/api/Indicator/All	
GET	/api/Indicator/{id}	
GET	/api/Indicator/{id}/GetValues	
GET	/api/Indicator/{id}/Areas	
GET	/api/ThematicIndices/All	
GET	/api/ThematicIndices/{id}	
GET	/api/ThematicIndices/{id}/GetValues	
GET	/api/ThematicIndices/{id}/Areas	
GET	/api/ThematicIndices/Areas	
GET	/api/Area	x
GET	/api/Area/indicator/{id}	x
GET	/api/Area/municipalities	x
GET	/api/Area/{areaId}/municipalities	x
GET	/api/Calculation/indicators	x
GET	/api/Calculation/indicators/{indicatorId}	x
GET	/api/Calculation/ecosystems	x
GET	/api/Calculation/municipalityEcosystems	x

GET	/api/Calculation/municipalityEcosystems/ecosystem/{ecosystemId}	x
GET	/api/Calculation/indicatorEcosystems	x
GET	/api/Calculation/indicatorEcosystems/ecosystem/{ecosystemId}	x
GET	/api/Calculation/indicatorEcosystems/indicator/{indicatorId}	x
GET	/api/Calculation/distributions	x
GET	/api/Calculation/functionalGroups	x
GET	/api/Calculation/scalingModels	x
GET	/api/Calculation/refYears	x
GET	/api/Calculation/thematicIndices	x
GET	/api/Calculation/thematicIndices/{id}	x
GET	/api/Calculation/thematicIndices/indicators	x
GET	/api/Calculation/thematicIndices/{id}/indicators	x
GET	/api/Calculation/indicator/{id}/values	x
GET	/api/Calculation/indicator/{indicatorId}/values/years/{year}	x
GET	/Indicators/{indicatorId}/values	x
GET	/Indicators/{indicatorId}/values/years/{year}	x
POST	/Indicators/values	x
GET	/Indicators/values/customDistributions/{id}	x
POST	/token	x

8 References

Certain, G. & Skarpaas, O. 2010. Nature Index: General framework, statistical method and data collection for Norway. NINA Report. 542, Norwegian Institute for Nature Research.

Certain, G., Skarpaas, O., Bjerke, J.-W., Framstad, E., Lindholm, M., Nielsen, J.-E., Norderhaug, A., Oug, E., Pedersen, H.-C., Schartau, A.-K., Storaunet, K. O., Van der Meeren, G. I., Aslaksen, I., Engen, S., Garnåsjordet P.-A., Kvaløy, P., Lillegård, M., Yoccoz N.G. & Nybø, S. 2011. The Nature Index: A General Framework for Synthesizing Knowledge on the State of Biodiversity. - PLoS ONE 6: e18930.

Framstad, E. (ed.) 2015. Naturindeks for Norge 2015. The Norwegian Nature Index 2015 – state and trends of biodiversity. Norwegian Environment Agency. (In Norwegian)

Honaker J. & King, G. 2010. What to Do about Missing Values in Time-Series Cross-Section Data. American Journal of Political Science 54: 561–581.

Honaker, J., King, G. & Blackwell, M. 2011. Amelia II: A Program for Missing Data. Journal of Statistical Software 45 (7): 1–47.

Jackson, C.H. 2011. Multi-State Models for Panel Data: The msm Package for R. Journal of Statistical Software 38 (8): 1–29.

Jakobsson, S. & Pedersen, B. (red.) 2020. Naturindeks for Norge 2020. Tilstand og utvikling for biologisk mangfold. NINA Rapport 1886, Norwegian Institute for Nature Research. (In Norwegian)

Johansen, L., Carlsen, T., Hassel, K., Kallioniemi, E., Staverløkk, A., Pedersen, B. & Wehn, S. 2019. Naturindeks for Norge: Evaluering av indikatorer innen åpent lavland. NIBIO Rapport 84, Norwegian Institute of Bioeconomy Research. (In Norwegian)

Mersmann, O., Trautmann, H., Steuer, D. & Bornkamp, B. 2018. truncnorm: Truncated Normal Distribution. R package version 1.0-8. https://CRAN.R-project.org/package=truncnorm

Müller, K. & Wickham, H. 2021. tibble: Simple Data Frames. R package version 3.0.6. https://CRAN.R-project.org/package=tibble

Ooms, J. 2014. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805 [stat.CO] URL https://arxiv.org/abs/1403.2805.

Pedersen, B. & Skarpaas, O. 2012. Statistical properties of the Nature index for Norway. Measurement error and sensitivity. NINA Report 797, Norwegian Institute for Nature Research. (In Norwegian)

Pedersen, B. & Kvaløy, P. 2015: Manual for entering data to the nature index database. Version 2.1. NINA Report 1139, Norwegian Institute for Nature Research.

Pedersen, B., Nybø, S. and Skarpaas, O. 2013. Ecological framework for the Nature Index. A more rigorous approach to the determination of reference values and selection of indicators. NINA Minireport 428, Norwegian Institute for Nature Research.

Pedersen, B., Nybø, S. & Sæther, S. A. (eds.) 2016. Nature Index for Norway 2015. Ecological framework, computational methods, database and information systems. NINA Report 1226, Norwegian Institute for Nature Research.

Pedersen, B., Bjerke, J.W., Pedersen, H.C., Brandrud, T.E., Gjershaug, J.O., Hanssen, O., Lyngstad, A. & Øien, D.-I. 2018. Naturindeks for Norge – fjell og våtmark. Evaluering av eksisterende indikatorsett, dets datagrunnlag og behovet for ytterligere tilfang av datakilder. NINA Report 1462, Norwegian Institute for Nature Research. (In Norwegian)

Pedersen, B., Schartau, A.K., Kielland, Ø.N., Kjærstad, G. & Persson, J. 2019. Nature index for Norway – freshwater. Evaluation of indicators and datasets and suggestions for adjustments. NINA Report 1683, Norwegian Institute for Nature Research. (In Norwegian)

R Core Team 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Ruckdeschel, P., Kohl, M., Stabla, T. & Camphausen, F. 2006. S4 Classes for Distributions. R News 6 (2): 2–6. URL https://CRAN.R-project.org/doc/Rnews/

Sarkar, D. 2008. Lattice: Multivariate Data Visualization with R. Springer, New York.

Schafer, J. L. 1997. Analysis of Incomplete Multivariate Data. London: Chapman & Hall, London.

Schartau, A.K., Gundersen, H. & Pedersen, B. 2016. Likheter og forskjeller i vannforskriftens klassifisering og naturindeksen for ferskvann/kystvann. Evaluering av metodikk og datagrunnlag. NINA Kortrapport 5, Norwegian Institute for Nature Research. (In Norwegian)

Stasinopoulos, M. & Rigby, R. 2021. gamlss.dist: Distributions for Generalized Additive Models for Location Scale and Shape. R package version 5.3-2. https://CRAN.R-project.org/package=gamlss.dist

Urbanek, S. & Ts'o, T. 2020. uuid: Tools for Generating and Handling of UUIDs. R package version 0.1-4. https://CRAN.R-project.org/package=uuid

Van Buuren, S. 2007. Multiple Imputation of Discrete and Continuous Data by Fully Conditional Specification. Statistical Methods in Medical Research 16: 219–242.

Van Buuren, S. 2018. Flexible Imputation of Missing Data, 2nd ed., CRC Press, Boca Raton, FL.

Van Buuren, S. & Groothuis-Oudshoorn, C. G. M. 2011. ‘Mice‘: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software 45 (3): 1–67.

Wickham, H. 2011. The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software 40 (1): 1–29.

Wickham, H. 2020. httr: Tools for Working with URLs and HTTP. R package version 1.4.2. https://CRAN.R-project.org/package=httr

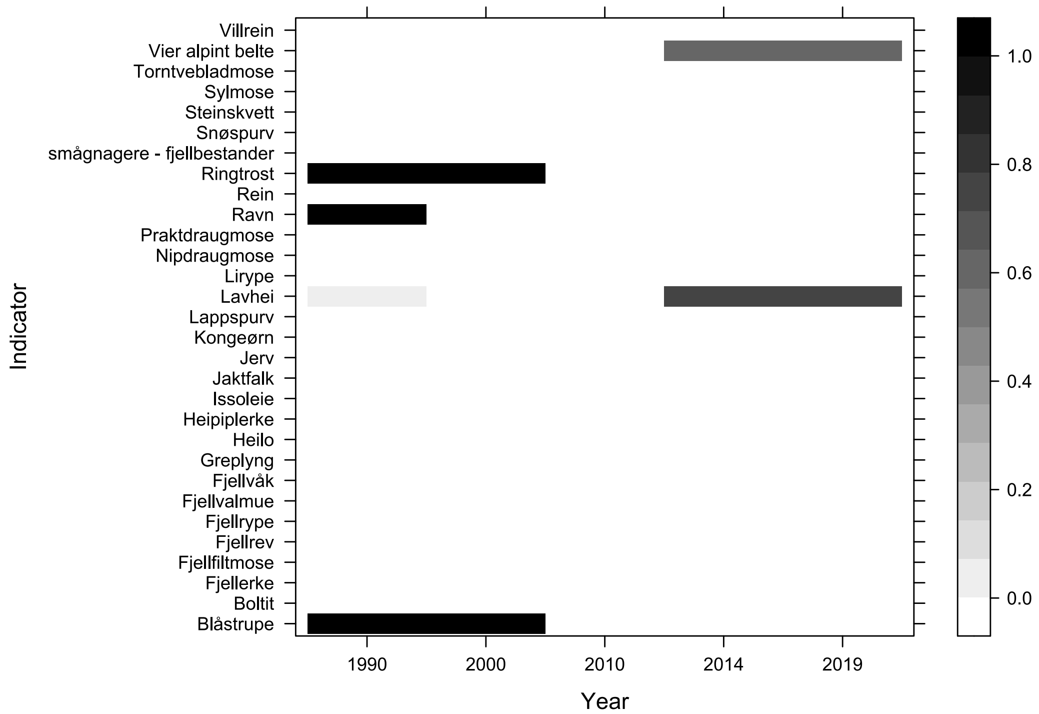
9 Appendix 1 - Nicalc - example runs

Here we present some example runs using functions in the NIcalc package. Partial console output and comments are in grey and green respectively.

9.1 Calculating the Nature Index for mountains

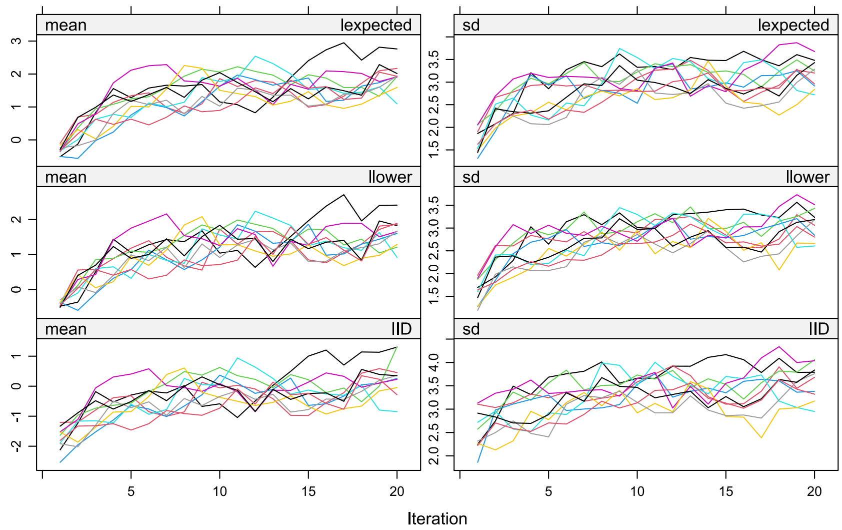
```
# Import data set for Mountains from Nature Index database. Include data from
# 1990, 2000, 2010, 2014, and 2019. Use Norwegian names on ecosystems and indicators.
> mountainImport <- NIcalc::importDatasetApi(
                             username = "Bard.Pedersen@nina.no",
                             password = "...",
                             eco = "Fjell",
                             indic = NULL,
                             year = c("1990","2000","2010","2014","2019"),
                             norwegian = TRUE,
                             refYearCode = 0)
Token retrieved from https://www8.nina.no/NaturindeksNiCalc !
Importing from T_Okosystem table .....
Importing from Kommune_Okosystem table .....
...
Importing from 'Verdier' table .....
Reading values for indicator 288
...
Reading values for indicator 52
# Assemble dataset and check for data completeness and consistency.
# Specify whole land area of Norway and all five country parts as Niunits.
# Only include Bsunits where the mountain ecosystem covers more than 20% of the
# terrestrial area.
> mountainInput <- NIcalc::assembleNiObject(
                             inputData = mountainImport,
                             predefNIunits = c(allArea = T, parts = T,
                                               counties = F),
                             indexType = "ecosystem",
                             part = "ecosystem",
                             total = "terrestrial",
                             partOfTotal = 0.2)
Check input data .....
Elicitate incomplete reference values and indicator observations .....
First attempt to estimate quartiles (fast routine using lapply) .....
Assemble dataset .....
Warning message:
In checkInputData(inputData = inputData) :
Checking the dataset generated 1 warning(s):
Time series from areas 132, 133, 134, .... 257, 258, 259, 260 contain at least one missing
indicator observation each.
```

```
# Reduce the number of functional groups in the data set in order to avoid
# unintentionally emphasizing single non-key indicators. I.e. merge
# generalists and specialists into one group for each trophic level.
> xxx <- yyy <- mountainInput$indicators$functionalGroup
> xxxId <- yyyId <- mountainInput$indicators$functionalGroupId
> yyy[xxxId %in% c(1,2)] <- "Mellompredator"
> yyyId[xxxId %in% c(1,2)] <- 1
> yyy[xxxId %in% c(6,7)] <- "Primærprodusent"
> yyyId[xxxId %in% c(6,7)] <- 6
> yyy[xxxId %in% c(8,9)] <- "Topp-predator"
> yyyId[xxxId %in% c(8,9)] <- 8
> mountainInput$indicators$functionalGroup <- yyy
> mountainInput$indicators$functionalGroupId <- yyyId
# Imputations for missing values.
> mountainDiagnostics <- NIcalc::imputeDiagnostics(x = mountainInput,
                                              nSim = 10,
                                              transConst = 0.01,
                                              maxit = 20)
Multiple imputations:
m = 10 imputations for each of 83 missing indicator observations .....
> mountainDiagnostics$diagnostics$mdDistribuiton
```

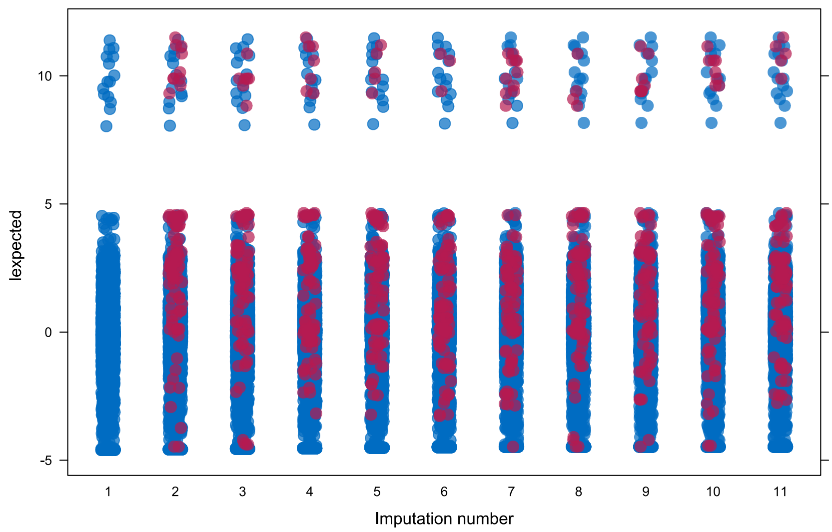



```
> mountainDiagnostics$diagnostics$mdPattern
     fyears findicators lexpected llower lID
3272      1           1         1      1   1   0
83        1           1         0      0   0   3
          0           0        83     83  83 249
```

```
> mountainDiagnostics$diagnostics$convergencePlot
```

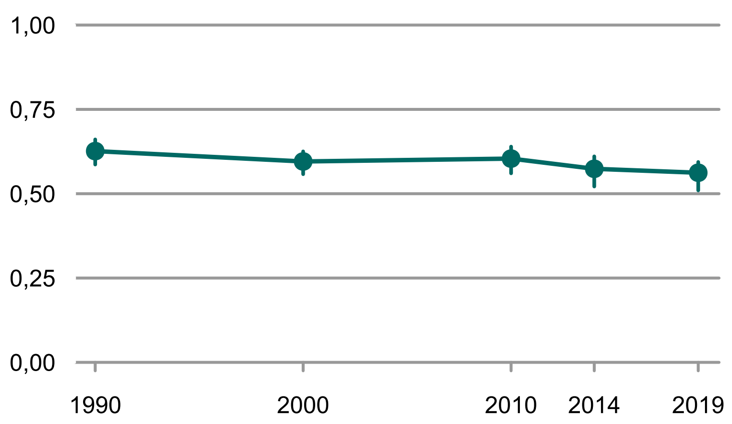



```
> mountainDiagnostics$diagnostics$stripplotExpected
```




```
> mountainImputes <- NIcalc::imputeData(x = mountainInput,
                                        nSim = 1000,
                                        transConst = 0.01,
                                        maxit = 20,
                                        printFlag = TRUE)
Multiple imputations:
m = 1000 imputations for each of 83 missing indicator observations .....
...
Sampling n = 1 draw from each of 1000 imputed distributions for each of 83 missing indicator
observations ...
> mountainImputesCorr <- NIcalc::impStand(x = mountainInput,
                                          imputations = mountainImputes)
```

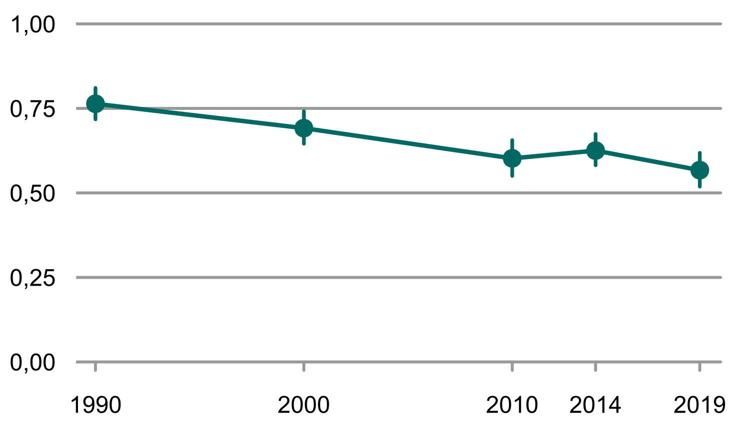
```
# Calculate Nature Index for mountains
> mountainIndex <- NIcalc::calculateIndex(x = mountainInput,
                                imputations = mountainImputesCorr,
                                awBSunit = "Fjell",
                                nsim = 1000,
                                fids = FALSE,
                                tgroups = TRUE,
                                keys = "specialWeight",
                                w = 0.5)
Indices for NIunits 'wholeArea', 'E', 'S', 'W', 'C', 'N'
and years '1990', '2000', '2010', '2014', '2019' will be calculated.
The 30 index distributions will each be based on  1000 simulations.
There are 671 ICunits with observations in data set 'mountainInput'.
Calculating weights that are the same for all years .....
Sampling reference values .....
Sampling and scaling indicator observations from  1990 .....
Sampling and scaling indicator observations from  2000 .....
Sampling and scaling indicator observations from  2010 .....
Sampling and scaling indicator observations from  2014 .....
Sampling and scaling indicator observations from  2019 .....
> NIcalc::summary(mountainIndex$wholeArea)
                    2.5%    median     97.5% displacement
wholeArea 1990 0.5865024 0.6263073 0.6608267  -0.03200836
wholeArea 2000 0.5579075 0.5955603 0.6253488  -0.02505816
wholeArea 2010 0.5604902 0.6040597 0.6394702  -0.01505446
wholeArea 2014 0.5213839 0.5738119 0.6106023  -0.03700540
wholeArea 2019 0.5098596 0.5621753 0.5939236  -0.03387043
> NIcalc::plot(mountainIndex$wholeArea, main = "",
               cex = 2, whiskerEnds = FALSE)
```



9.2 Calculating a thematic index: amphibia

```
# Import data set for amphibian thematic index from Nature Index database.
# Include data from 1990, 2000, 2010, 2014, and 2019. Use Norwegian names on indicators.
> amphibiaImport <- NIcalc::importDatasetApi(
                        username = "Bard.Pedersen@nina.no",
                        password = "...",
                        eco = NULL,
                        indic = c("Småsalamander","Buttsnutefrosk","Storsalamander"),
                        year = c("1990","2000","2010","2014","2019"),
                        norwegian = TRUE,
                        refYearCode = 0)
Token retrieved from https://www8.nina.no/NaturindeksNiCalc !
Importing from T_Okosystem table .....
Importing from Kommune_Okosystem table .....
...
Importing from 'Verdier' table .....
Reading values for indicator 173
Reading values for indicator 188
Reading values for indicator 212
# Assemble dataset and check for data completeness and consistency.
# Specify whole land area of Norway and all five country parts as Niunits.
amphibiaInput <- NIcalc::assembleNiObject(
                        inputData = amphibiaImport,
                        predefNIunits = c(allArea = T, parts = T, counties = F),
                        indexType = "thematic")
Check input data .....
Elicitate incomplete reference values and indicator observations .....
First attempt to estimate quartiles .....
Assemble dataset .....
# I.e. no missing values
# Calculate thematic index
amphibiaIndex <- NIcalc::calculateIndex(
                        x = amphibiaInput,
                        nsim = 1000,
                        fids = FALSE,
                        tgroups = FALSE,
                        keys = "ignore",
                        w = 0,
                        awbs = TRUE,
                        awBSunit = "Ferskvann")
Indices for NIunits 'wholeArea', 'E', 'S', 'W', 'C', 'N'
and years '1990', '2000', '2010', '2014', '2019' will be calculated.
The 30 index distributions will each be based on  1000 simulations.
There are 35 ICunits with observations in data set 'amphibiaInput'.
Calculating weights that are the same for all years .....
Sampling reference values .....
Sampling and scaling indicator observations from  1990 .....
```

```
Sampling and scaling indicator observations from  2000 .....
Sampling and scaling indicator observations from  2010 .....
Sampling and scaling indicator observations from  2014 .....
Sampling and scaling indicator observations from  2019 .....
# Results for Eastern Norway:
> NIcalc::summary(amphibiaIndex$E)
           2.5%    median     97.5% displacement
E 1990 0.7175502 0.7637120 0.8106270 -0.003518250
E 2000 0.6454153 0.6915935 0.7416636 -0.005168597
E 2010 0.5502890 0.6022126 0.6561287 -0.001209135
E 2014 0.5817651 0.6251688 0.6742140 -0.001770768
E 2019 0.5186727 0.5673496 0.6185993 -0.001187170
> NIcalc::plot(amphibiaIndex$E,cex=2,whiskerEnds=F)
```

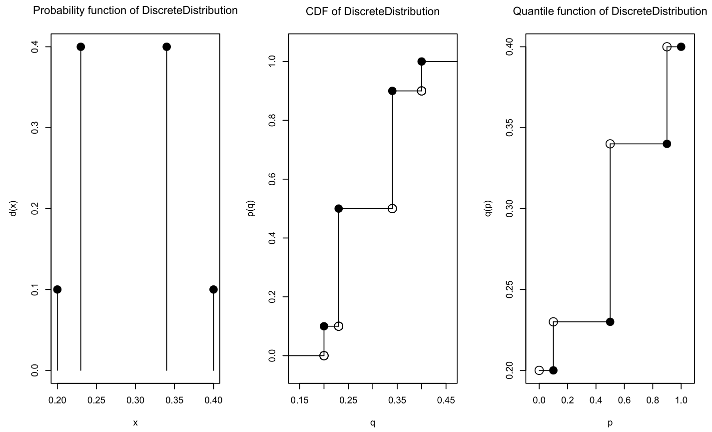



```
> NIcalc::summaryWeights(amphibiaIndex$E)
                  E 1990    E 2000    E 2010    E 2014    E 2019
Buttsnutefrosk 0.5247579 0.5247579 0.5247579 0.5247579 0.5247579
Småsalamander  0.3089460 0.3089460 0.3089460 0.3089460 0.3089460
Storsalamander 0.1662961 0.1662961 0.1662961 0.1662961 0.1662961
```

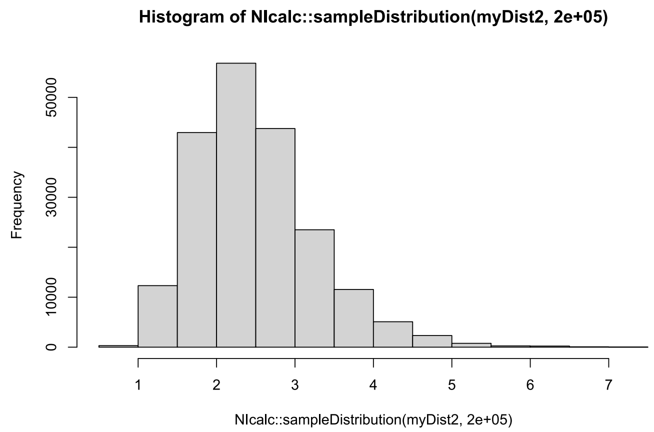
9.3 Distribution objects

```
# Create distribution objects
# Example 1: User defined descrete distribution
> myProbs <- data.frame("est" = c(0.2, 0.23, 0.34, 0.4),
                        "probs" = c(0.1, 0.4, 0.4, 0.1 ))
> myDist1 <- NIcalc::makeDistribution(myProbs)
> NIcalc::sampleDistribution(myDist1, 20)
  [1] 0.34 0.20 0.23 0.40 0.34 0.23 0.34 0.34 0.23 0.34 0.34 0.23 0.23 0.34 0.23 0.34
 [17] 0.34 0.23 0.34 0.34
```

```
> distr::plot(myDist1)
```

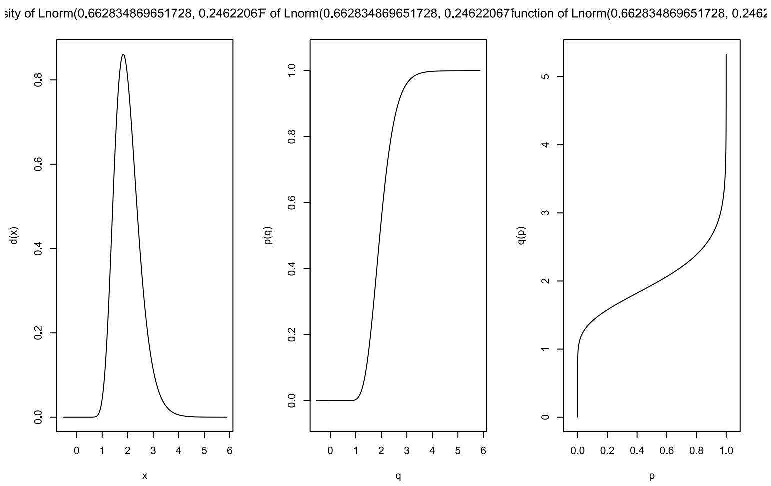



```
# Example 2: Based on a large sample of observations (e.g. a MCMC sample).
> codaSamples <- rlnorm(100000, mean = 0.87, sd = 0.3)
> myDist2 <- NIcalc::makeDistribution(codaSamples)
> hist(NIcalc::sampleDistribution(myDist2, 200000))
```




```
# Example 3: Model distributions. Poisson.
> myDist3 <- NIcalc::makeDistribution(input = "Poisson", distParams = list("lambda" = 3))
# Example 4: Two non-negative point estimates with standard error. Lognormal distribution
# fitted to both.
> myData <- data.frame(estim = c(1, 2),
                       stErr = c(2, 0.5))
> logNormalParams <- NIcalc::normal2Lognormal(mean = myData$estim,
                                               sd = myData$stErr)
> myData$muLogNorm <- logNormalParams$mean
> myData$sigLogNorm <- logNormalParams$sd
> ddd <- NULL
> for (i in 1:dim(myData)[[1]]) {
   ddd[i] <- list(NIcalc::makeDistribution(
     input = "logNormal",
     distParams = list(mean = myData$muLogNorm[i],
                       sd = myData$sigLogNorm[i])))
 }
```

```
> ddd
[[1]]
Distribution Object of Class: Lnorm
 meanlog: -0.80471895621705
 sdlog: 1.26863624117952
[[2]]
Distribution Object of Class: Lnorm
 meanlog: 0.662834869651728
 sdlog: 0.24622067706924
> myData$distrObjects <- ddd
> myData
  estim stErr   muLogNorm  sigLogNorm                                       distrObjects
1     1   2.0  -0.8047190   1.2686362 <S4 class 'Lnorm' [package "distr"] with 12 slots>
2     2   0.5   0.6628349   0.2462207 <S4 class 'Lnorm' [package "distr"] with 12 slots>
> distr::plot(myData$distrObjects[[2]])
```




```
# Example 5: Elicitation of 10 expert judgments
> x <- NIcalc::elicitate(expected.value = 2:11,lower = c(0,2:10), upper = 5:14)
> x
       FK_DistID         mu       sig       ssq
1  ZIExponential  0.3204295 0.2298763 1.3366874
2    TruncNormal  2.3070778 3.5404686 1.0330797
3    TruncNormal  4.2423157 3.0984178 0.8686658
4    TruncNormal  5.5054391 2.9765415 0.7769487
5    TruncNormal  6.6026195 2.9468103 0.7232179
6    TruncNormal  7.6420052 2.9458566 0.6932889
7    TruncNormal  8.6578506 2.9522500 0.6779806
8        Weibull 10.6865554 3.8241768 0.6490866
9        Weibull 11.7100458 4.2162205 0.6193232
10       Weibull 12.7297232 4.6074579 0.5946387
```

9.4 Updating indicator observations, and communicating with the database

```
# get access to the database
> NIcalc::getToken(username="bard.pedersen@nina.no",
                   password="...",
                   url = "https://www8.nina.no/NaturindeksNiCalc")
Token retrieved from https://www8.nina.no/NaturindeksNiCalc !
# List accessible indicators (will differ among users)
> myIndicators <- NIcalc::getIndicators()
> myIndicators
   id                          name
1   3                 Alge på bjørk
2  11 Begroing elver eutrofierings indeks
3  19                  Blåsteinbit
4  27                        Brosme
5 359                 Humler i skog
# Read indicator observations to be updated or revised from the database
> indicatorData <- NIcalc::getIndicatorValues(
  indicatorID = myIndicators$id[myIndicators$name == "Humler i skog"],
  years = c("Referanseverdi","2014","2019"))
> indicatorData
$indicatorValues
   indicatorId indicatorName areaId           areaName yearId       yearName
1          359 Humler i skog   7053    Østfold Vestfold      0 Referanseverdi
9          359 Humler i skog   7053    Østfold Vestfold      8           2014
10         359 Humler i skog   7053    Østfold Vestfold      9           2019
11         359 Humler i skog   7054           Trøndelag      0 Referanseverdi
19         359 Humler i skog   7054           Trøndelag      8           2014
20         359 Humler i skog   7054           Trøndelag      9           2019
21         359 Humler i skog   7055 Vest-Agder Rogaland      0 Referanseverdi
29         359 Humler i skog   7055 Vest-Agder Rogaland      8           2014
30         359 Humler i skog   7055 Vest-Agder Rogaland      9           2019
   verdi nedre_Kvartil ovre_Kvartil datatypeId     datatypeName unitOfMeasurement
1   1.00          0.00         0.00          1 Ekspertvurdering         Enhetsløs
9   0.59          0.45         0.54          2  Overvåkingsdata         Enhetsløs
10    NA            NA           NA         NA             <NA>              <NA>
11  1.00          0.00         0.00          1 Ekspertvurdering         Enhetsløs
19  0.74          0.63         0.74          2  Overvåkingsdata         Enhetsløs
20    NA            NA           NA         NA             <NA>              <NA>
21  1.00          0.00         0.00          1 Ekspertvurdering         Enhetsløs
29  0.55          0.49         0.55          2  Overvåkingsdata         Enhetsløs
30    NA            NA           NA         NA             <NA>              <NA>
   customDistributionUUID distributionName distributionId distParam1  distParam2
1                      NA            Gamma              6  1.0000000 12.35169525
9                      NA        LogNormal              2 -0.6473736  0.13709669
10                     NA             <NA>             NA         NA          NA
11                     NA            Gamma              6  1.0000000 12.35169525
```

```
19                    NA        LogNormal            2 -0.3565711  0.11997121
20                    NA             <NA>           NA         NA          NA
21                    NA            Gamma            6  1.0000000 12.35169525
29                    NA        LogNormal            2 -0.6372550  0.08596413
30                    NA             <NA>           NA         NA          NA

$customDistributions
named list()

attr(,"class")
[1] "indicatorData" "list"
> updatedIndicatorData <- indicatorData # Just in case - make a copy
# Update with distrbution objects from Appendix 1 section 9.3
> updatedIndicatorData <- NIcalc::setIndicatorValues(updatedIndicatorData,
                                areaId = 7054,
                                years = 2014,
                                distribution = myDist1,
                                datatype = 3, # Model prediction
                                unitOfMeasurement = "Enhetsløs")
updatedIndicatorData <- NIcalc::setIndicatorValues(updatedIndicatorData,
            areaId = 7054,
            years = 2019,
            distribution = NIcalc::makeDistribution(
                    input = "logNormal",
                    distParams = NIcalc::normal2Lognormal(mean = 1, sd = 2)),
            datatype = 3,
            unitOfMeasurement = "Enhetsløs")
# Update expert judgment
updatedIndicatorData <- NIcalc::setIndicatorValues(updatedIndicatorData,
            areaId = 7055,
            years = 2019,
            est = 2, lower = 1, upper = 3,
            datatype = 1, # Expert judgment
            unitOfMeasurement = "Enhetsløs")
# Updating several values in a for-loop
> myData$areaIDs <- c(7053,7053)
> myData$years <- c("Referanseverdi","2019")
> for (i in 1:dim(myData)[[1]]) {
      updatedIndicatorData <- NIcalc::setIndicatorValues(updatedIndicatorData,
                 areaId = myData$areaIDs[i],
                 years = myData$years[i],
                 distribution = myData$distrObjects[[i]],
                 datatype = 3,
                 unitOfMeasurement = "Enhetsløs")  }
```

```
> updatedIndicatorData
$indicatorValues
$indicatorValues
   indicatorId indicatorName areaId           areaName yearId       yearName
1          359 Humler i skog   7053    Østfold Vestfold      0 Referanseverdi
9          359 Humler i skog   7053    Østfold Vestfold      8           2014
10         359 Humler i skog   7053    Østfold Vestfold      9           2019
11         359 Humler i skog   7054           Trøndelag      0 Referanseverdi
19         359 Humler i skog   7054           Trøndelag      8           2014
20         359 Humler i skog   7054           Trøndelag      9           2019
21         359 Humler i skog   7055 Vest-Agder Rogaland      0 Referanseverdi
29         359 Humler i skog   7055 Vest-Agder Rogaland      8           2014
30         359 Humler i skog   7055 Vest-Agder Rogaland      9           2019
       verdi nedre_Kvartil ovre_Kvartil datatypeId          datatypeName  UnitOfMeasurement
1  1.0000000            NA           NA          3 Beregnet fra modeller          Enhetsløs
9  0.5900000          0.45         0.54          2       Overvåkingsdata          Enhetsløs
10 2.0000000            NA           NA          3 Beregnet fra modeller          Enhetsløs
11 1.0000000          0.00         0.00          1      Ekspertvurdering          Enhetsløs
19 0.2879238            NA           NA          3 Beregnet fra modeller          Enhetsløs
20 1.0000000            NA           NA          3 Beregnet fra modeller          Enhetsløs
21 1.0000000          0.00         0.00          1      Ekspertvurdering          Enhetsløs
29 0.5500000          0.49         0.55          2       Overvåkingsdata          Enhetsløs
30 2.0000000          1.00         3.00          1       Ekspervurdering          Enhetsløs
                customDistributionUUID distributionName distributionId distParam1  distParam2
1  d8488f50-bcae-4d2f-8518-4e8862031b93             <NA>             NA         NA          NA
9                                 <NA>        LogNormal              2 -0.6473736  0.13709669
10 8042461e-347e-475e-be12-7624b78f4169             <NA>             NA         NA          NA
11                                <NA>            Gamma              6  1.0000000 12.35169525
19 2d0e1336-11f6-4f91-bc32-3e4b2772c6e2             <NA>             NA         NA          NA
20 cdbf2fd1-a28c-4752-896c-b164b523f657             <NA>             NA         NA          NA
21                                <NA>            Gamma              6  1.0000000 12.35169525
29                                <NA>        LogNormal              2 -0.6372550  0.08596413
30                                <NA>             <NA>             NA         NA          NA

$customDistributions
$customDistributions$`2d0e1336-11f6-4f91-bc32-3e4b2772c6e2`
Distribution Object of Class: DiscreteDistribution

$customDistributions$`cdbf2fd1-a28c-4752-896c-b164b523f657`
Distribution Object of Class: Lnorm
 meanlog: -0.80471895621705
 sdlog: 1.26863624117952

$customDistributions$`d8488f50-bcae-4d2f-8518-4e8862031b93`
Distribution Object of Class: Lnorm
 meanlog: -0.80471895621705
 sdlog: 1.26863624117952
```

```
$customDistributions$`8042461e-347e-475e-be12-7624b78f4169`
Distribution Object of Class: Lnorm
 meanlog: 0.662834869651728
 sdlog: 0.24622067706924

attr(,"class")
[1] "indicatorData" "list"
# Update the database with the new values
> NIcalc::writeIndicatorValues(updatedIndicatorData)
```

10 Appendix 2 - S3 classes defined within NIcalc

S3 classes defined within NIcalc are lists containing input data to calculations of the Nature Index and objects containing the results from the calculation.

10.1 Class *niDataImport*

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	Indicator data: $id, $name, $active, $keyElement, $functionalGroupId, $functionalGroup, $scalingModelId, $scalingModel, and one or more variables with fidelities to major ecosystems.
[[2]]	referenceValues $referenceValues $customDistributions	list of 2 data.frame list	Reference values $indId, $indName, $ICunitId, $ICunitName, $yearId, $yearName, $expectedValue, $lowerQuantile, $upperQuantile, $customDistributionUUID (name of corresponding distribution object), $distributionFamilyId, $distributionFamilyName, $distParameter1, $distParameter2 list of named distribution objects
[[3]]	indicatorObservations $indicatorValues $customDistributions	list of 2 data.frame list	Indicator measurements Same variables as for reference values. list of named distribution objects
[[4]]	ICunits	data.frame	Delimitation of ICunits: $id, $name, $BSunitId, $indId
[[5]]	BSunits	data.frame	BSunit data: $id, $name, and additional variables describing BSunits.
[[6]]	ecosystems	data.frame	$id, $name

10.2 Class *niInput*

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	Indicator data: $id, $name, $keyElement, $functionalGroup, $functionalGroupId, and 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 ICunit that includes the corresponding BSunit.
[[3]]	BSunits	data.frame	BSunit data: $id, $name, $area, and optional variables describing BSunits.
[[4]]	referenceValues	data.frame	Reference values: $indId, $indName, $ICunitId, $ICunitName, $yearId, $yearName, $expectedValue, $lowerQuantile, $upperQuantile, $customDistributionUUID (name of corresponding distribution object), $distributionFamilyId, $distributionFamilyName, $distParameter1, $distParameter2, $customDistribution
[[5]]	indicatorValues	list	Indicator measurements: Each element is a data.frame with the same variables as [[referenceValues]] + $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.

10.3 Class *niImputations*

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. *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, $indName, $year, $refss, $stringsAsFactors
[[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.

10.4 Results: Classes *niOutput*, *niSeries* and *niValue*

The function *calculateIndex()* calculates indices from *niInput* and *niImputations* lists and produces an extensive output for each index value to facilitate further analyses of the results. *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. Each element is a list of class *niSeries*. Lists of class *niSeries* contain the results from the calculation of a (time) series of index values for a single NIunit. Each element is a list of class *niValue*. Lists of class *niValue* contain 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	numeric vector	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	integer matrix	BSunit x indicator matrix giving the delineation of each ICunit in terms of BSunits.
[[10]]	imputations	data.frame	indicators and ICunits with imputed values
[[11]]	BSunits	character vector	BSunit names
[[12]]	BSunitData	data.frame	Input BSunit data
[[13]]	BSunitWeights	numeric matrix	BSunit x indicator matrix of BSunit weights
[[14]]	NIunitWeights	numeric vector	NIunit weights for each BSunit
[[15]]	BSunitIndices	numeric matrix	BSunit x nsim matrix of index values BSunit
[[16]]	BSunitbbb	numeric vector	bbb statistic for each BSunit. Used to calculate location displacement
[[17]]	indexWeights	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").
[[4]]	www	numerical	trophic weight given to key indicators when keys="special-Weight".
[[5]]	awbs	logical	Are weights based on a BSunit variable, or are all BSunits within the NIunit given equal weight?
[[6]]	stochastic	character	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 vary among years in the time series?
[[10]]	nsim	numerical	number of bootstrap simulation

10.5 class *indicatorData*

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 database table *Verdier* 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, $distributionName – character, $distributionId – integer, $distParam1 – numeric, $distParam2 – numeric.
[[2]]	$customDistributions	list	list of named distribution objects

www.nina.no

1990

NINA Report

The Norwegian Institute for Nature Research, NINA,
is as an independent foundation focusing on environmental research, emphasizing the interaction between human society, natural resources and biodiversity.

NINA was established in 1988. The headquarters are located in Trondheim, with branches in Tromsø, Lillehammer, Bergen and Oslo. In addition, NINA owns and runs the aquatic research station for wild fish at Ims in Rogaland and the arctic fox breeding center at Oppdal.

NINA's activities include research, environmental impact assessments, environmental monitoring, counselling and evaluation. NINA's scientists come from a wide range of disciplinary backgrounds that include biologists, geographers, geneticists, social scientists, sociologists and more. We have a broad-based expertise on the genetic, population, species, ecosystem and landscape level, in terrestrial, freshwater and coastal marine ecosystems.

ISSN: 1504-3312
ISBN: 978-82-426-4769-6

Norwegian Institute for Nature Research
NINA head office
Postal address: P.O. Box 5685 Torgard,
NO-7485 Trondheim, NORWAY
Visiting address: Høgskoleringen 9, 7034 Trondheim
Phone: +47 73 80 14 00
E-mail: firmapost@nina.no
Organization Number: 9500 37 687
http://www.nina.no

NINA

Cooperation and expertise for a sustainable future