Package: openNCAI (via r-universe)

October 28, 2026

Type Package

Title Calculates a Natural Capital Assets Index

Version 0.2.0

Date 2026-09-02

Description Calculates a regional natural capital assets index (NCAI) following the methodology designed by NatureScot for Scotland as described in Albon, Balana, Brooker & Eastwood (2014) <https://www.nature.scot/sites/default/files/2025-06/naturescot-commissioned-report-751.pdf> and McKenna et al. (2019) <doi:10.1016/J.ECOLIND.2019.105645>. Processes habitat extent and condition data alongside metadata and weighting systems to produce a yearly single figure indexed relative to a base-year value of 100.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

LazyData true

URL https://github.com/oharakate/openNCAI

BugReports https://github.com/oharakate/openNCAI/issues

Imports dplyr, janitor, openxlsx, readxl, rlang, slider, stats, stringr, tidyr

RoxygenNote 7.3.3

VignetteBuilder knitr

Suggests ggplot2, knitr, rmarkdown, rprojroot, testthat (>= 3.0.0), tibble

Config/testthat/edition 3

Depends R (>= 4.1.0)

Config/pak/sysreqs libicu-dev

Repository https://oharakate.r-universe.dev

Date/Publication 2026-09-28 15:41:25 UTC

RemoteUrl https://github.com/oharakate/openncai

RemoteRef HEAD

RemoteSha e9780b3519cef1e9806d4a31b9565c90dd726718

Contents

calc_ncai_subset . . . 2
check_missing . . . 4
create_ncai_template . . . 5
get_ncai . . . 7
index_and_smooth . . . 10
ns_between_importance_scores . . . 11
ns_ci_relevance_matrices . . . 12
ns_ci_scores . . . 12
ns_custom_divisor_matrix . . . 13
ns_display_ci_names . . . 13
ns_display_es_label_tree . . . 14
ns_display_habitats_label_tree . . . 14
ns_es_label_tree . . . 15
ns_habitat_extent . . . 15
ns_habitats_label_tree . . . 16
ns_indicator_directory . . . 16
ns_provision_per_unit_scores . . . 17
ns_within_importance_scores . . . 17
ns_year_list . . . 18
read_ncai_template . . . 18
show_missing . . . 20

Index **21**

`calc_ncai_subset` *Calculate the NCAI for a Custom Subset of Habitats and Services*

Description

Indexes the NCAI over any combination of habitats (rows of the yearly NCAI matrices) and ecosystem services (columns), for example cultural services from woodland habitats. Labels are checked against the matrices before indexing, so a mistyped label stops with an error rather than being silently left out of the total.

Usage

```
calc_ncai_subset(
  yearly_ncai_matrices,
  habitats = NULL,
  services = NULL,
```

```
  habitats_label_tree = NULL,
  es_label_tree = NULL,
  year_one = names(yearly_ncai_matrices)[[1]],
  smoothing_weights = c(0.2, 0.4, 0.6, 0.8, 1)
)
```

Arguments

yearly_ncai_matrices	A named list of yearly NCAI matrices, as returned by `get_ncai(return = "yearly_ncai_matrices")`. The names of the list must be the years.
habitats	Optional. A character vector of habitat labels (row names) to include. May also contain group names from `habitats_label_tree`, which are expanded to every habitat in that group. Defaults to all habitats.
services	Optional. A character vector of ecosystem service labels (column names) to include. May also contain group names from `es_label_tree`, which are expanded to every service in that group. Defaults to all services.
habitats_label_tree	Optional. A named list defining the hierarchy of habitats. Needed only if `habitats` contains group names.
es_label_tree	Optional. A named list defining the hierarchy of ecosystem services. Needed only if `services` contains group names.
year_one	The baseline year for indexing (where index = 100). Defaults to the first year in `yearly_ncai_matrices`. Set this to match the `year_one` given to `get_ncai()` if that was not the first year.
smoothing_weights	Numeric vector of weights for 5-year trailing smoothing. Defaults to `c(0.2, 0.4, 0.6, 0.8, 1.0)`.

Details

A habitat or service that is selected more than once, directly or through a group, is counted once.

The total of the selected cells in `year_one` must be non-zero, as the index is expressed relative to it.

Value

A data frame with one row per year (years as row names) and columns `raw_total`, `raw_index` and `smoothed_index`, as returned by `index_and_smooth`.

See Also

`get_ncai`, `index_and_smooth`

Examples

```
yearly_ncai_matrices <- get_ncai(
  habitat_extent = ns_habitat_extent,
  ci_scores = ns_ci_scores,
```

```
  habitats_label_tree = ns_habitats_label_tree,
  es_label_tree = ns_es_label_tree,
  year_list = ns_year_list,
  provision_per_unit_scores = ns_provision_per_unit_scores,
  custom_divisor_matrix = ns_custom_divisor_matrix,
  between_importance_scores = ns_between_importance_scores,
  within_importance_scores = ns_within_importance_scores,
  ci_relevance_matrices = ns_ci_relevance_matrices,
  indicator_directory = ns_indicator_directory,
  return = "yearly_ncai_matrices"
)

# Cultural services from woodland habitats, selected by group name
calc_ncai_subset(
  yearly_ncai_matrices,
  habitats = "g_woodland_forest_and_other_wooded_land",
  services = "cultural",
  habitats_label_tree = ns_habitats_label_tree,
  es_label_tree = ns_es_label_tree
)

# All services from broadleaved woodland only
calc_ncai_subset(
  yearly_ncai_matrices,
  habitats = "g1_broadleaved_deciduous_woodland"
)
```

`check_missing` *Check pipeline inputs for missing data*

Description

Checks one or more openNCAI pipeline objects for missing data and reports a summary via `message()`. Data frames are checked cell by cell. Named lists (label trees and weight trees, up to two levels of nesting, e.g. `habitats_label_tree` or `within_importance_scores`) are checked element by element, and blank or `NA` *names* are treated as missing alongside blank or `NA` *values*, since openNCAI matches on these labels downstream. Bare vectors (e.g. `year_list`) are checked element by element. `ci_relevance_matrices`-shaped input (a named list of data frames) is checked per element, with its own header line.

`check_missing()` is run automatically at the start of `get_ncai`; if any input has missing data, it stops the pipeline before any calculation runs. It can also be called directly on any named list of objects.

Usage

```
check_missing(inputs, label_cols = NULL)
```

Arguments

inputs	A named list of objects to check.
label_cols	Optional named character vector mapping names in inputs to the column that should be used as the row label in the report, for data frames whose row names aren't meaningful (e.g. c(indicator_directory = "ci_id")).

Value

Invisibly, TRUE if no missing data was found. Stops with an error if any input has missing data.

Examples

```
check_missing(list(habitat_extent = ns_habitat_extent))
```

---

create_ncai_template *Create an NCAI Data Entry Template*

Description

Generates a highly formatted, protected Excel (.xlsx) workbook containing the necessary sheets for a user to populate an openNCAI account. The template includes internal logic for habitat and ecosystem service trees, automated styling, and cell locking to ensure data integrity for subsequent re-import. Note that the habitats and ecosystem service label trees passed here may be used to subsequently read in the data with read_ncai_template().

Usage

```
create_ncai_template(
  template_out,
  habitats_label_tree,
  es_label_tree,
  ci_names,
  year_list,
  overwrite = FALSE,
  habitat_extent = NULL,
  provision_per_unit_scores = NULL,
  between_importance_scores = NULL,
  within_importance_scores = NULL,
  ci_scores = NULL,
  indicator_directory = NULL,
  ci_relevance_matrices = NULL
)
```

Arguments

`template_out`	A string representing the file path where the .xlsx will be saved.
`habitats_label_tree`	A named list of character vectors representing the habitat hierarchy. Names: broad habitats, typically EUNIS level 1. Character vector items: typically EUNIS level 2 habitats.
`es_label_tree`	A named list of character vectors representing the ecosystem service hierarchy. Names: typically SEEA ecosystem service types. Character vector items: typically CICES-type ecosystem services.
`ci_names`	A character vector of condition indicator names.
`year_list`	A vector of years (numeric or character) to be included in the account.
`overwrite`	Logical. If `TRUE`, an existing file at `template_out` will be overwritten without warning. Default is `FALSE` to prevent accidental loss of manually entered data.
`habitat_extent`	Optional data frame of existing habitat extent data. Row and column order and dimensions MUST matched label trees/year list.
`provision_per_unit_scores`	Optional data frame of existing ES Potential Unit scores. Row and column order MUST match lower levels of `habitats_label_tree` and `es_label_tree`.
`between_importance_scores`	Optional list of ecosystem service type importance weights. Order MUST match names of `es_label_tree`.
`within_importance_scores`	Optional list of lists of weights for specific services. Order MUST match the order and dimensions of the lower level of `es_label_tree`.
`ci_scores`	Optional data frame of existing condition indicator scores. Rows and columns MUST match dimensions and order of `year_list` and `ci_names`.
`indicator_directory`	Optional data frame of existing indicator salience data. Row order and length MUST match `ci_names`. Column order and length MUST match `es_label_tree` names.
`ci_relevance_matrices`	Optional list of existing binary relevance matrices. Matrix list order and length must match `ci_names`. Each matrix row and column order MUST match lower levels of `habitats_label_tree` and `es_label_tree`.

Details

Optional arguments to pre-populate the template with data assume that the passed-in data matches those of the label trees, year list and condition indicator list as relevant. Users are recommended to create an empty template first to verify the required formats before attempting to pass in data.

Value

Generates an Excel file at the specified path and returns a message of success.

Examples

```
# 1. Define the habitat hierarchy
habitat_label_tree <- list(
  'B. Coastal Habitats' = c(
    "Coastal vegetated shingle",
    "Coastal dunes and sandy shores"
  ),
  'E. Grasslands' = c(
    "Dry Grasslands",
    "Mesic Grasslands"
  )
)

# 2. Define the ecosystem service hierarchy
ecosystem_service_label_tree <- list(
  'PROVISIONING' = c(
    "1.1 Cultivated Crops",
    "1.2 Reared Animals And Their Outputs"
  ),
  'CULTURAL' = c(
    "3.5. Existence & bequest"
  )
)

# 3. Define indicators and temporal scope
condition_indicator_list <- c("National Water Quality Index", "AgriSCOR")
list_of_years <- 2020:2025

# 4. Generate the template in a temporary location
# tempfile() ensures a valid path that won't clutter your working directory.
tmp_path <- tempfile(fileext = ".xlsx")

create_ncai_template(
  template_out = tmp_path,
  habitats_label_tree = habitat_label_tree,
  es_label_tree = ecosystem_service_label_tree,
  ci_names = condition_indicator_list,
  year_list = list_of_years
)

message("Template successfully created at: ", tmp_path)
```

---

get_ncai *Main User Interface for the Natural Capital Asset Index (NCAI)*

Description

This is the primary function for calculating the NCAI. It processes habitat extent, condition scores, and importance weights through the full NCAI pipeline, with options to return intermediate matrices or specific breakdowns.

Usage

```
get_ncai(
  habitat_extent,
  ci_scores,
  habitats_label_tree,
  es_label_tree,
  year_list,
  year_one = NULL,
  provision_per_unit_scores,
  provision_per_unit_divisor = 5,
  custom_divisor_matrix = NULL,
  between_importance_scores,
  within_importance_scores,
  ci_relevance_matrices,
  indicator_directory,
  total_indicator_relevances_constant = 2,
  smoothing_weights = c(0.2, 0.4, 0.6, 0.8, 1),
  return = c("overall_ncai", "by_ecosystem_service_type", "by_broad_habitat",
    "wellbeing_index", "flow_of_es_index", "yearly_ncai_matrices",
    "yearly_wellbeing_matrices", "flow_of_es_matrices", "es_potential_base",
    "wellbeing_potential_base", "flow_of_es_base", "everything")
)
```

Arguments

`habitat_extent`	A data frame of habitat area/extent per year.
`ci_scores`	A matrix of condition indicator scores.
`habitats_label_tree`	A named list defining the hierarchy of habitats.
`es_label_tree`	A named list defining the hierarchy of ecosystem services.
`year_list`	A vector (character or numeric) of years for the index.
`year_one`	Optional. The baseline year for indexing (where index = 100). Defaults to the first year in `year_list`.
`provision_per_unit_scores`	A data frame of Ecosystem Service Provision Potential per Unit.
`provision_per_unit_divisor`	Numeric. A standard divisor to convert Provision Per Unit scores to weights; the number out of which Provision Per Unit scores have been awarded. Default value is 5. Alternatively, a `custom_divisor_matrix` may be provided.
`custom_divisor_matrix`	Optional. A matrix of divisors specific to habitat/service combinations.
`between_importance_scores`	Scores representing the relative importance between different ecosystem service types.
`within_importance_scores`	Scores representing the relative importance within ecosystem service types.

`ci_relevance_matrices`
: Condition Indicator Relevance Matrices list.

`indicator_directory`
: Directory mapping indicators to services/habitats.

`total_indicator_relevances_constant`
: Numeric, must be >= 0. The constant used in the Total Indicator Relevance calculation. Defaults to 2.

`smoothing_weights`
: Numeric vector of weights for 5-year trailing smoothing. Defaults to `c(0.2, 0.4, 0.6, 0.8, 1.0)`.

`return`
: Character. Specifies the object to return. Options include:

- `"overall_ncai"`: The standard overall NCAI data frame (default).
- `"by_ecosystem_service_type"`: NCAI broken down by Ecosystem Service Type.
- `"by_broad_habitat"`: NCAI broken down by Broad Habitat.
- `"wellbeing_index"`: The potential wellbeing contribution of the habitats (before weighting by likely flow of services) over the years, indexed.
- `"flow_of_es_index"`: The likely flow of ecosystem services (based on information from condition indicators) over the years, indexed.
- `"yearly_ncai_matrices"`: The overall NCAI in its unaggregated form, expressed as yearly matrices of value per habitat/ecosystem service.
- `"yearly_wellbeing_matrices"`: The yearly wellbeing potential in its unaggregated form, expressed as yearly matrices of value per habitat/ecosystem service.
- `"yearly_flow_of_es_matrices"`: The yearly likely flow of ecosystem services in its unaggregated form, expressed as yearly matrices of value per habitat/ecosystem service.
- `"es_potential_base"`: The Ecosystem Service Potential Base matrix, i.e. the habitat extent weighted weighted by exemplary provision-per-unit scores in year one.
- `"wellbeing_potential_base"`: The Wellbeing Potential Base, ie. year one potential wellbeing matrix.
- `"flow_of_es_base"`: The year one likely flow of services matrix.
- `"everything"`: A named list containing all of the above.

Details

Missing Data: `check_missing` is run on the core pipeline inputs before any calculation. If any input has missing data, `get_ncai()` stops with a report naming the affected objects; run `show_missing` on a flagged object for detail.

Mandatory Inputs: Users must provide either an `provision_per_unit_divisor` or a `custom_divisor_matrix` to convert potential scores into weights.

Smoothing and Baseline Years: The smoothed index is calculated using a 5-year trailing window. If `year_one` is set to a year other than the first year of the dataset, the `smoothed_index` value for that baseline year will likely not be exactly 100. This is because the smoothing reflects the trend of the preceding 4 years. The `raw_index` will always remain anchored at 100 for the `year_one`.

Value

An object of the type specified by the `return` argument. Typically a data frame or a named list of NCAI components.

Examples

```
# Calculate the NCAI using the bundled NatureScot sample data
ncai_objects <- get_ncai(
  habitat_extent = ns_habitat_extent,
  ci_scores = ns_ci_scores,
  habitats_label_tree = ns_habitats_label_tree,
  es_label_tree = ns_es_label_tree,
  year_list = ns_year_list,
  provision_per_unit_scores = ns_provision_per_unit_scores,
  custom_divisor_matrix = ns_custom_divisor_matrix,
  between_importance_scores = ns_between_importance_scores,
  within_importance_scores = ns_within_importance_scores,
  ci_relevance_matrices = ns_ci_relevance_matrices,
  indicator_directory = ns_indicator_directory,
  return = "everything"
)

# View the final indexed results
names(ncai_objects)
head(ncai_objects$overall_ncai)
```

---

index_and_smooth *Index matrices and apply smoothing*

Description

This function calculates the raw and smoothed Natural Capital Asset Index from a list of yearly asset matrices, e.g. the NCAI matrices. It anchors the index to a specific baseline year and applies a weighted trailing moving average.

Usage

```
index_and_smooth(
  matrix_list,
  smoothing_weights = c(0.2, 0.4, 0.6, 0.8, 1),
  year_one = names(matrix_list)[[1]]
)
```

Arguments

`matrix_list`	A named list of matrices, where each matrix represents the values for a specific year, with rows = habitats and columns = ecosystem service type. The names of the list must be the years.

`smoothing_weights`
: A numeric vector of weights used for the trailing 5-year weighted smoothing. Defaults to `c(0.2, 0.4, 0.6, 0.8, 1.0)`.

`year_one`
: Character or Numeric. The year used as the baseline (where index = 100). Defaults to the first name in `matrix_list`. The total in this year must be non-zero.

Details

Smoothing and Baseline Years: The smoothed index is calculated using a 5-year trailing window via a weighted moving average (using the `slider` package). If `year_one` is set to a year other than the first year of the dataset, the `smoothed_index` value for that baseline year will likely not be exactly 100. This is because the smoothing reflects the trend of the preceding 4 years. The `raw_index` will always remain anchored at 100 for the `year_one`.

Value

A data frame with one row per year (years as row names) containing:

- `raw_total`: The absolute sum of assets for each year.
- `raw_index`: The index value relative to the baseline year.
- `smoothed_index`: The weighted 5-year trailing smoothed trend.

See Also

`calc_ncai_subset`, which checks habitat and service labels before subsetting the yearly NCAI matrices and indexing them with this function.

Examples

```
yearly_matrices <- list(
  "2000" = matrix(c(10, 20, 30, 40), nrow = 2),
  "2001" = matrix(c(12, 20, 33, 41), nrow = 2),
  "2002" = matrix(c(15, 19, 35, 44), nrow = 2)
)
index_and_smooth(yearly_matrices)
```

---

ns_between_importance_scores

Ecosystem Service Importance Scores (between-service-type)

Description

A set of scores denoting the importance of each type group of ecosystem services to Scotland.

Usage

```
ns_between_importance_scores
```

Format

A named list of scores where values range from 0 to 20, representing the relative importance of each ecosystem service type. Item names are ecosystem service types.

---

ns_ci_relevance_matrices

Condition Indicator Relevance Matrix List

Description

A set of matrices recording whether or not each condition indicator is relevant to each habitat/ecosystem service combination.

Usage

```
ns_ci_relevance_matrices
```

Format

A named list of data frames:

- Names are condition indicator IDs matching `ns_indicator_directory`.
- Each data frame has habitats as rows and ecosystem services as columns.
- Values are binary (0 or 1) indicating relevance.

---

ns_ci_scores

Condition Indicator Score Matrix

Description

A matrix containing the raw condition indicator scores over the years of the time series.

Usage

```
ns_ci_scores
```

Format

A data frame:

- Rows represent years in the time series.
- Columns represent condition indicators (e.g., "Adult red grouse density").
- Values are numeric condition scores.

ns_custom_divisor_matrix *Nature Scot Custom Divisor Matrix*

Description

A matrix containing numbers by which the Ecosystem Service Potential per Unit Scores will be divided to calculate a weight.

Usage

```
ns_custom_divisor_matrix
```

Format

A data frame:

- Rows represent habitats matching the habitats label tree.
- Columns represent ecosystem services matching the ecosystem services label tree.
- Values are numeric divisors used for weight normalization.

ns_display_ci_names *Display names of Condition Indicators*

Description

Display name (no string cleaning) list of condition indicator names which is used internally.

Usage

```
ns_display_ci_names
```

Format

A character vector where items are the original number and name of NatureScot's 38 condition indicators:

ns_display_es_label_tree

Display name version of Ecosystem Services Label Tree

Description

Display name version (no string cleaning) version of `ns_es_label_tree` which is used internally.

Usage

```
ns_display_es_label_tree
```

Format

A named list of character vectors:

- Names represent ecosystem service types (e.g., "Provisioning").
- Values are character vectors of specific ecosystem services (e.g., "Cultivated crops").

ns_display_habitats_label_tree

Display name version of Habitats Label Tree

Description

Display name version (no string cleaning) version of `ns_habitats_label_tree` which is used internally.

Usage

```
ns_display_habitats_label_tree
```

Format

A named list of character vectors:

- Names represent broad habitat groups (e.g., "Coastal habitats").
- Values are character vectors of specific habitat types (e.g., "Coastal shingle").

ns_es_label_tree *Ecosystem Services Label Tree*

Description

A hierarchical tree mapping ecosystem service types to specific ecosystem services.

Usage

```
ns_es_label_tree
```

Format

A named list of character vectors:

- Names represent ecosystem service types (e.g., "Provisioning").
- Values are character vectors of specific ecosystem services (e.g., "Cultivated crops").

ns_habitat_extent *Habitat Extent for Scotland*

Description

A table containing the annual extent (area) of various habitats.

Usage

```
ns_habitat_extent
```

Format

A data frame:

- Rows represent broad habitats matching the habitats label tree.
- Columns represent years (e.g., 2000 to 2022).
- Values are numeric representing area in hectares.

`ns_habitats_label_tree`
Habitats Label Tree

Description

A hierarchical tree mapping broad habitat groups to specific habitat types.

Usage

```
ns_habitats_label_tree
```

Format

A named list of character vectors:

- Names represent broad habitat groups (e.g., "Coastal habitats").
- Values are character vectors of specific habitat types (e.g., "Coastal shingle").

`ns_indicator_directory`
Indicator Directory

Description

A table recording the salience of habitat condition scores as indicators of the likely flow of services under each ecosystem service type.

Usage

```
ns_indicator_directory
```

Format

A data frame with the following columns:

- `ci_id`: Unique condition indicator identifier.
- `provisioning`: Salience for Provisioning services.
- `regulation_and_maintenance`: Salience for Regulation and Maintenance services.
- `cultural`: Salience for Cultural services.

ns_provision_per_unit_scores

Provision Per Unit Scores

Description

A matrix of scores denoting the exemplary capacity of each habitat to provide each ecosystem service.

Usage

```
ns_provision_per_unit_scores
```

Format

A data frame:

- Rows represent habitats matching the habitats label tree.
- Columns represent ecosystem services matching the ecosystem services label tree.
- Values are numeric scores from 0 (No potential) to 5 (Maximum potential).

ns_within_importance_scores

Ecosystem Service Importance Scores (within-service-type)

Description

Sets of scores denoting the importance of each ecosystem service to Scotland, within its type group.

Usage

```
ns_within_importance_scores
```

Format

A named list of named lists of scores:

- Names represent ecosystem service types.
- Names of scores represent individual ecosystem services
- Values are numeric scores (0-20) for specific services within that group.

ns_year_list *Year List*

Description

A list of years for which data is available and over which the index will be calculated.

Usage

```
ns_year_list
```

Format

A character vector of year labels (e.g., "2000", "2001").

read_ncai_template *Read and Process a Populated NCAI Template*

Description

Imports data from an Excel workbook created by `create_ncai_template`. The function uses the provided label trees to re-align the data and applies automated cleaning to headers to ensure compatibility with internal openNCAI calculation engines. Note that the habitat and ecosystem service label trees should match the labels used in the spreadsheet; typically the same label trees should be used to generate and read the template.

Usage

```
read_ncai_template(path, habitats_label_tree, es_label_tree, ci_names)
```

Arguments

`path`	String. The file path to the populated .xlsx file.
`habitats_label_tree`	A named list of character vectors representing the habitat hierarchy. Names: broad habitats, typically EUNIS level 1. Character vector items: typically EUNIS level 2 habitats.
`es_label_tree`	A named list of character vectors representing the ecosystem service hierarchy. Names: typically SEEA ecosystem service types. Character vector items: typically CICES-type ecosystem services.
`ci_names`	The original character vector of indicator names used to create the template.

Value

A named list of data structures ready for use in `get_ncai`:

- `clean_habitats_label_tree`: Cleaned version of habitat hierarchy.
- `clean_es_label_tree`: Cleaned version of ES hierarchy.
- `habitat_extent`: Data frame of habitat areas over time.
- `ci_scores`: Data frame of indicator scores over time.
- `provision_per_unit_scores`: Data frame of ES potential per unit area.
- `between_importance`: List of broad ES type weights.
- `within_importance`: List of specific ES service weights.
- `indicator_directory`: Data frame mapping indicators to service types.
- `ci_relevance_matrices`: List of binary matrices for every indicator.

Examples

```
# 1. Define the trees (same as used in create_ncai_template)
habitat_label_tree <- list(
  'B. Coastal Habitats' = c("Coastal vegetated shingle", "Coastal dunes"),
  'E. Grasslands' = c("Dry Grasslands", "Mesic Grasslands")
)

ecosystem_service_label_tree <- list(
  'PROVISIONING' = c("1.1 Cultivated Crops"),
  'CULTURAL' = c("3.5. Existence & bequest")
)

condition_indicator_list <- c("National Water Quality Index")

# 2. Create a temporary template to read back in
# This ensures the example is self-contained and runnable.
tmp_path <- tempfile(fileext = ".xlsx")

create_ncai_template(
  template_out = tmp_path,
  habitats_label_tree = habitat_label_tree,
  es_label_tree = ecosystem_service_label_tree,
  ci_names = condition_indicator_list,
  year_list = 2024:2025
)

# 3. Read the template
# Even though it's empty, the function will process the headers and structure.
ncai_data <- read_ncai_template(
  path = tmp_path,
  habitats_label_tree = habitat_label_tree,
  es_label_tree = ecosystem_service_label_tree,
  ci_names = condition_indicator_list
)
```

```
# 4. Access the cleaned data structures
# Show the first few rows of the imported habitat extent matrix
head(ncai_data$habitat_extent)
```

---

show_missing *Show where a single object has missing data*

Description

Shows missing data for one openNCAI pipeline object, chosen by `check_missing` as the follow-up step once it's flagged something. Data frames (including individual elements of `ci_relevance_matrices`, e.g. `ci_relevance_matrices$some_ci`) are shown as a bar chart of % missing per row, using ggplot2. Named lists (label trees and weight trees) and bare vectors are shown as a plain-text listing of which specific entries are blank or missing, via `message()`, since a bar chart doesn't apply to those shapes.

Usage

```
show_missing(x, label_col = NULL)
```

Arguments

`x`	The object to check: a data frame, named list, or vector.
`label_col`	Optional column name to use as the row label instead of `rownames(x)`, for data frames whose row names aren't meaningful (e.g. `indicator_directory`, which should be shown with `label_col = "ci_id"`).

Value

For data frames with missing data, a ggplot2 plot object. For complete data frames, and for lists and vectors, invisibly `NULL` (the listing, if any, is printed via `message()`).

Examples

```
show_missing(ns_habitats_label_tree)
```

Index

∗ **datasets**
ns_between_importance_scores, 11
ns_ci_relevance_matrices, 12
ns_ci_scores, 12
ns_custom_divisor_matrix, 13
ns_display_ci_names, 13
ns_display_es_label_tree, 14
ns_display_habitats_label_tree, 14
ns_es_label_tree, 15
ns_habitat_extent, 15
ns_habitats_label_tree, 16
ns_indicator_directory, 16
ns_provision_per_unit_scores, 17
ns_within_importance_scores, 17
ns_year_list, 18

calc_ncai_subset, 2, *11*
check_missing, 4, *9*, *20*
create_ncai_template, 5

get_ncai, *3, 4*, 7

index_and_smooth, *3*, 10

ns_between_importance_scores, 11
ns_ci_relevance_matrices, 12
ns_ci_scores, 12
ns_custom_divisor_matrix, 13
ns_display_ci_names, 13
ns_display_es_label_tree, 14
ns_display_habitats_label_tree, 14
ns_es_label_tree, 15
ns_habitat_extent, 15
ns_habitats_label_tree, 16
ns_indicator_directory, 16
ns_provision_per_unit_scores, 17
ns_within_importance_scores, 17
ns_year_list, 18

read_ncai_template, 18

show_missing, *9*, 20