Package: NRMstatsML (via r-universe)

June 23, 2026

Type Package

Title Statistical and Machine Learning Engine for Long-Term Natural Resource Management Data

Version 0.1.4

Description A comprehensive toolkit for statistical and machine learning-based analysis of long-term Natural Resource Management (NRM) datasets. Integrates formula-driven approaches, statistical inference, and machine learning (ML) models for advanced analytics. Modules cover trend and structural analysis (Mann-Kendall test, slope estimation, Chow test, structural break detection), multivariate system modelling (Partial Least Squares (PLS), Structural Equation Modelling (SEM)), response curve optimisation, time-series forecasting (Autoregressive Integrated Moving Average (ARIMA), hybrid models), panel data and treatment effects (Difference-in-Differences (DiD), causal machine learning), uncertainty and sensitivity analysis (bootstrap, Monte Carlo, Bayesian), and automated model selection and performance comparison. Designed for long-term datasets covering soil, water, crop, and climate domains. Key references: Mann and Kendall (1945) <doi:10.2307/1907187>; Sen (1968) <doi:10.1080/01621459.1968.10480934>; Bai and Perron (2003) <doi:10.1002/jae.659>; Rosseel (2012) <doi:10.18637/jss.v048.i02>; Croissant and Millo (2008) <doi:10.18637/jss.v027.i02>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Language en-US

Depends R (>= 4.1.0)

Imports Kendall (>= 2.2), trend (>= 1.1.4), strucchange (>= 1.5.3), plm (>= 2.6.0), forecast (>= 8.20), lavaan (>= 0.6.12), pls (>= 2.8.0), caret (>= 6.0.93), boot (>= 1.3.28), ggplot2 (>= 3.4.0), rlang (>= 1.1.0), stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, keras, tensorflow, BayesianTools, sensitivity, mboost, mlr3, covr

VignetteBuilder knitr

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Sadikul Islam [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-2924-7122>)

Maintainer Sadikul Islam <sadikul.islamiasri@gmail.com>

Config/pak/sysreqs libicu-dev

Repository https://sadikulislamiasri-hub.r-universe.dev

Date/Publication 2026-06-07 18:30:08 UTC

RemoteUrl https://github.com/cran/NRMstatsML

RemoteRef HEAD

RemoteSha f2bfeca18b07c45d9f58d1fb43c8aae4231e53a8

Contents

NRMstatsML-package *NRMstatsML: Statistical and Machine Learning Engine for Long-Term NRM Data*

Description

A comprehensive R package for statistical and machine learning-based analysis of long-term Natural Resource Management (NRM) datasets. It integrates formula-driven approaches, statistical inference, and machine learning models for advanced analytics across soil, water, crop, and climate domains.

Core Modules

trendML Trend and structural analysis: Mann-Kendall test, Sen's slope, structural break detection (Chow, Bai-Perron), and ML-based trend prediction. See `nrm_trend`, `nrm_mann_kendall`.

multiSysML Multivariate system modelling: regression, PLS, SEM, and ML feature selection. See `nrm_multivariate`, `nrm_pls`.

responseML Response curve and input optimisation: quadratic and nonlinear regression. See `nrm_response_curve`, `nrm_optimize_input`.

tsML Time series forecasting: ARIMA, SARIMA, LSTM, and hybrid models. See `nrm_forecast`, `nrm_arima`.

panelML Panel data and treatment effects: fixed/random effects, DiD, causal ML. See `nrm_panel`, `nrm_did`.

uncertaintyML Uncertainty and sensitivity analysis: bootstrap, Monte Carlo, Bayesian modelling. See `nrm_uncertainty`, `nrm_bootstrap`.

autoML Automated model selection, tuning, and benchmarking. See `nrm_automl`, `nrm_benchmark`.

Typical Workflow

1. Validate and prepare data with `nrm_data_check`.
2. Run trend analysis with `nrm_trend` or `nrm_mann_kendall`.
3. Fit models with the relevant module function.
4. Summarise results with `nrm_summary`.
5. Visualise with `nrm_plot`.

Author(s)

Maintainer: Sadikul Islam <sadikul.islamiasri@gmail.com> (ORCID) [copyright holder]

References

Mann, H. B. (1945). Nonparametric tests against trend. *Econometrica*, 13, 245–259.

Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association*, 63, 1379–1389.

nrm_arima *ARIMA model for NRM time series*

Description

Fits an ARIMA model with automatic order selection via **forecast**.

Usage

```
nrm_arima(data, value_var, time_var = "year", seasonal = TRUE, frequency = 1L)
```

Arguments

data	A data frame ordered by time.
value_var	Character. Column to forecast.
time_var	Character. Time index column. Default "year".
seasonal	Logical. If TRUE (default), seasonal ARIMA terms are considered (SARIMA).
frequency	Integer. Seasonal frequency (e.g. 12 for monthly, 1 for annual data). Default 1.

Value

A list of class "nrm_arima" with the fitted model and AIC/BIC.

Examples

```
data(nrm_example)
ar <- nrm_arima(nrm_example, value_var = "crop_yield")
print(ar)
```

nrm_automl *Automated model selection and tuning*

Description

Trains multiple model types via **caret**, selects the best model by cross-validated Root Mean Square Error (RMSE), and returns a ranked comparison table.

Usage

```
nrm_automl(
  data,
  formula,
  methods = c("lm", "rf", "gbm", "svmRadial"),
  cv_folds = 5L,
  seed = 42L
)
```

Arguments

`data`	A data frame.
`formula`	A model formula.
`methods`	Character vector of **caret** method names to compare. Default: `c("lm", "rf", "gbm", "svmRadial")`.
`cv_folds`	Integer. Number of cross-validation folds. Default `5`.
`seed`	Integer. Random seed for reproducibility. Default `42`.

Value

A list of class `"nrm_automl"` with:

`best_model` The best-performing trained **caret** model.

`best_method` Name of the best method.

`leaderboard` Data frame ranking all methods by Root Mean Square Error (RMSE).

`models` Named list of all fitted models.

Examples

```
data(nrm_example)
am <- nrm_automl(nrm_example,
                 formula  = crop_yield ~ N + P + K + rainfall,
                 methods  = c("lm", "rf"),
                 cv_folds = 5)
nrm_summary(am)
```

`nrm_benchmark` *Benchmark model metrics on a hold-out test set*

Description

Evaluates a set of model objects against a hold-out test set and computes Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared.

Usage

```
nrm_benchmark(models, test_data, response_var)
```

Arguments

`models`	A named list of fitted model objects that support `predict`.
`test_data`	A data frame (the hold-out set).
`response_var`	Character. Name of the response variable in `test_data`.

Value

A list of class "nrm_benchmark" containing a data frame metrics with Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared, ranked by RMSE.

Examples

```
data(nrm_example)
n     <- nrow(nrm_example)
train <- nrm_example[seq_len(floor(0.8 * n)), ]
test  <- nrm_example[seq(floor(0.8 * n) + 1L, n), ]
m1    <- lm(crop_yield ~ N + P + K, data = train)
bm    <- nrm_benchmark(list(ols = m1), test_data = test,
                       response_var = "crop_yield")
print(bm)
```

---

nrm_bootstrap *Bootstrap uncertainty estimation*

Description

Estimates confidence intervals for a user-defined statistic via non-parametric bootstrapping using **boot**.

Usage

```
nrm_bootstrap(data, stat_fn, n_iter = 1000L, alpha = 0.05, ...)
```

Arguments

data	A data frame.
stat_fn	A function that takes a data frame and returns a single numeric value (or named numeric vector). Must accept a data argument as its first positional argument.
n_iter	Integer. Number of resamples / simulations. Default 1000.
alpha	Numeric. Significance level for confidence intervals. Default 0.05.
...	Additional arguments passed to stat_fn.

Value

A list of class "nrm_bootstrap" with mean, sd, ci, and raw (the **boot** object).

Examples

```
data(nrm_example)
bs <- nrm_bootstrap(nrm_example,
        stat_fn = function(d) mean(d$crop_yield),
        n_iter = 500)
print(bs)
```

nrm_data_check *Validate and summarise an NRM dataset*

Description

Checks a data frame for missing values, type consistency, and expected columns for NRM analysis. Returns a structured report.

Usage

```
nrm_data_check(data, time_var = "year", verbose = TRUE)
```

Arguments

data	A data frame, ideally containing columns such as year, treatment, crop_yield, soil_OC, N, P, K, rainfall, runoff, biomass.
time_var	Character. Name of the time column. Default "year".
verbose	Logical. Print report to console. Default TRUE.

Value

A list (invisibly) with components n_rows, n_cols, missing_summary, numeric_cols, and warnings.

Examples

```
data(nrm_example)
nrm_data_check(nrm_example)
```

nrm_did *Difference-in-Differences (DiD) estimator*

Description

Estimates the average treatment effect on the treated (ATT) using a two-period, two-group Difference-in-Differences (DiD) design appropriate for policy or intervention evaluation in Natural Resource Management (NRM) contexts.

Usage

```
nrm_did(data, outcome, treat_var, time_var, covariates = NULL)
```

Arguments

data	A data frame.
outcome	Character. Outcome variable name.
treat_var	Character. Binary treatment indicator (0/1).
time_var	Character. Binary post-period indicator (0/1).
covariates	Character vector of additional control variable names. Default NULL.

Value

A list of class "nrm_did" with:

att Average treatment effect on the treated (ATT).

model The underlying lm object.

p_value P-value of the DiD interaction term.

Examples

```
set.seed(42)
d <- data.frame(
  crop_yield = c(rnorm(20, 3.5, 0.4), rnorm(20, 4.2, 0.4)),
  treatment  = rep(c(0L, 1L), each = 20),
  post       = rep(c(0L, 1L, 0L, 1L), each = 10)
)
did <- nrm_did(d,
               outcome   = "crop_yield",
               treat_var = "treatment",
               time_var  = "post")
print(did)
```

---

nrm_example *Example long-term NRM dataset*

Description

A synthetic dataset mimicking 20 years of a long-term fertiliser experiment covering crop yield, soil properties, nutrient inputs, and climate variables. Generated for illustration and testing purposes only.

Usage

```
nrm_example
```

Format

A data frame with 20 rows and 10 variables:

year Integer. Year of observation (2000–2019).

treatment Character. Treatment label (`"control"` or `"NPK"`).

crop_yield Numeric. Grain yield (t/ha).

soil_OC Numeric. Soil organic carbon (%).

N Numeric. Nitrogen applied (kg/ha).

P Numeric. Phosphorus applied (kg/ha).

K Numeric. Potassium applied (kg/ha).

rainfall Numeric. Annual rainfall (mm).

runoff Numeric. Annual runoff (mm).

biomass Numeric. Total biomass (t/ha).

Source

Synthetically generated; not derived from any real experiment.

Examples

```
data(nrm_example)
head(nrm_example)
nrm_data_check(nrm_example)
```

---

nrm_forecast *Forecast NRM time series*

Description

High-level wrapper that fits an ARIMA or hybrid ARIMA+ML model and returns a forecast with confidence intervals.

Usage

```
nrm_forecast(
  data,
  value_var,
  time_var = "year",
  horizon = 5L,
  method = "auto_arima",
  frequency = 1L
)
```

Arguments

data	A data frame ordered by time.
value_var	Character. Column to forecast.
time_var	Character. Time index column. Default "year".
horizon	Integer. Forecast horizon (number of time steps). Default 5.
method	Character. "auto_arima" (default) or "hybrid".
frequency	Integer. Seasonal frequency (e.g. 12 for monthly, 1 for annual data). Default 1.

Value

A list of class "nrm_forecast" with:

forecast The **forecast** object.

method Method used.

horizon Forecast horizon.

accuracy In-sample accuracy metrics.

Examples

```
data(nrm_example)
fc <- nrm_forecast(nrm_example, value_var = "crop_yield", horizon = 5)
nrm_plot(fc)
```

---

nrm_mann_kendall *Mann-Kendall trend test*

Description

Applies the Mann-Kendall nonparametric test for monotonic trend in a time series. Suitable for non-normally distributed and autocorrelated data typical in long-term NRM records.

Usage

```
nrm_mann_kendall(data, time_var = "year", value_var, alpha = 0.05)
```

Arguments

data	A data frame containing at least the columns specified by time_var and value_var.
time_var	Character. Name of the column holding the time index (e.g. "year").
value_var	Character. Name of the column holding the response variable (e.g. "crop_yield").
alpha	Numeric. Significance level for the Mann-Kendall test. Default 0.05.

Value

A list of class "nrm_mann_kendall" with components:

tau Kendall's tau statistic.

p_value Two-sided p-value.

significant Logical; TRUE when p < alpha.

trend_direction Character: "increasing", "decreasing", or "no trend".

raw Full output from MannKendall.

References

Mann, H. B. (1945). Nonparametric tests against trend. *Econometrica*, 13, 245–259.

Kendall, M. G. (1975). *Rank Correlation Methods*. Griffin.

Examples

```
data(nrm_example)
mk <- nrm_mann_kendall(nrm_example, time_var = "year",
                       value_var = "crop_yield")
print(mk)
```

---

nrm_monte_carlo *Monte Carlo uncertainty simulation*

Description

Generates Monte Carlo samples by adding Gaussian noise (scaled to the column-wise standard deviation) to each numeric column and evaluates the statistic of interest across simulations.

Usage

```
nrm_monte_carlo(
  data,
  stat_fn,
  n_iter = 1000L,
  alpha = 0.05,
  noise_sd = 0.1,
  ...
)
```

Arguments

data	A data frame.
stat_fn	A function that takes a data frame and returns a single numeric value (or named numeric vector). Must accept a data argument as its first positional argument.
n_iter	Integer. Number of resamples / simulations. Default 1000.
alpha	Numeric. Significance level for confidence intervals. Default 0.05.
noise_sd	Numeric. Scale factor applied to column standard deviations when generating noise. Default 0.1 (10 % CV).
...	Additional arguments passed to stat_fn.

Value

A list of class "nrm_monte_carlo" with mean, sd, ci, and samples.

Examples

```
data(nrm_example)
mc <- nrm_monte_carlo(nrm_example,
        stat_fn = function(d) mean(d$crop_yield),
        n_iter = 500)
print(mc)
```

---

nrm_multivariate *Multivariate regression for Natural Resource Management systems*

Description

Fits a multivariate Ordinary Least Squares (OLS) regression model and returns standardised diagnostics suitable for Natural Resource Management (NRM) data.

Usage

```
nrm_multivariate(data, formula, scale = TRUE)
```

Arguments

data	A data frame.
formula	A model formula, e.g. crop_yield ~ N + P + K + rainfall.
scale	Logical. If TRUE (default), predictors are mean-centred and scaled to unit variance before fitting.

Value

A list of class "nrm_multivariate" with components:

model The fitted lm object.

coefficients Tidy coefficient table.

r_squared Adjusted R-squared.

vif Variance inflation factors (requires **car**).

Examples

```
data(nrm_example)
mv <- nrm_multivariate(nrm_example,
        formula = crop_yield ~ N + P + K + rainfall)
nrm_summary(mv)
```

---

nrm_optimize_input *Optimise input level for maximum response*

Description

Given a fitted response curve, computes the economically optimal input level based on input cost and output price ratios.

Usage

```
nrm_optimize_input(curve_result, price_ratio = 1)
```

Arguments

curve_result	An object of class "nrm_response_curve".
price_ratio	Numeric. Ratio of input cost to output price (e.g. cost per kg N / price per kg grain). Default 1.

Value

A list of class "nrm_optimize_input" with:

biophysical_optimum Input giving maximum biological yield.

economic_optimum Input giving maximum economic return.

price_ratio The price ratio used.

Examples

```
data(nrm_example)
rc  <- nrm_response_curve(nrm_example, input_var = "N",
                          response_var = "crop_yield")
opt <- nrm_optimize_input(rc, price_ratio = 0.5)
print(opt)
```

`nrm_panel` *Panel data regression for Natural Resource Management experiments*

Description

Fits fixed-effects or random-effects panel models using **plm**, with a Hausman test to guide model selection between fixed and random effects.

Usage

```
nrm_panel(data, formula, index, model = "within")
```

Arguments

`data`	A data frame in long format.
`formula`	A model formula, e.g. `crop_yield ~ N + P + rainfall`.
`index`	Character vector of length 2 giving the panel (`"unit"`) and time index columns, e.g. `c("site", "year")`.
`model`	Character. Panel model type: `"within"` (fixed effects, default), `"random"`, or `"pooling"`.

Value

A list of class `"nrm_panel"` with:

`model` The fitted `plm` object.

`hausman` Hausman test result (fixed effects vs random effects).

`summary` Model summary.

References

Croissant, Y., & Millo, G. (2008). Panel data econometrics in R: The plm package. *Journal of Statistical Software*, 27(2), 1–43. doi:10.18637/jss.v027.i02

Examples

```
set.seed(1)
d <- data.frame(
  site       = rep(c("A", "B", "C", "D"), each = 10),
  year       = rep(2010:2019, times = 4),
  crop_yield = rnorm(40, 4, 0.5),
  N          = rnorm(40, 90, 10),
  rainfall   = rnorm(40, 650, 50)
)
pm <- nrm_panel(d,
        formula = crop_yield ~ N + rainfall,
        index   = c("site", "year"),
        model   = "within")
nrm_summary(pm)
```

---

nrm_plot *Generic plot for NRMstatsML objects*

Description

Produces a **ggplot2** visualisation appropriate to the result type.

Usage

```
nrm_plot(x, ...)
```

Arguments

x	An NRMstatsML result object.
...	Currently unused.

Value

A ggplot2 object (invisibly).

`nrm_pls` *Partial Least Squares (PLS) regression*

Description

Fits a Partial Least Squares (PLS) regression model using `plsr`, suitable when predictors are collinear or when the number of predictors approaches the number of observations.

Usage

```
nrm_pls(data, formula, ncomp = NULL, validation = "CV")
```

Arguments

`data`	A data frame.
`formula`	A model formula.
`ncomp`	Integer. Number of PLS components to extract. If `NULL` (default), selected by cross-validation.
`validation`	Character. Cross-validation method passed to `plsr`: `"CV"` (default) or `"LOO"`.

Value

A list of class `"nrm_pls"` with components:

`model` The fitted `plsr` object.

`ncomp` Number of components used.

`rmsep` Root Mean Square Error of Prediction (RMSEP) by component.

References

Wold, S., Sjostrom, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58, 109–130.

Examples

```
data(nrm_example)
pl <- nrm_pls(nrm_example,
              formula = crop_yield ~ N + P + K + rainfall + soil_OC)
nrm_summary(pl)
```

`nrm_response_curve` *Fit a response curve to NRM data*

Description

Fits quadratic or nonlinear response curves relating an input (e.g. fertiliser rate) to a response variable (e.g. crop yield), commonly used in long-term fertiliser trial analysis.

Usage

```
nrm_response_curve(
  data,
  input_var,
  response_var,
  type = c("quadratic", "linear", "mitscherlich")
)
```

Arguments

`data`	A data frame.
`input_var`	Character. Predictor variable name (e.g. `"N"`).
`response_var`	Character. Response variable name (e.g. `"crop_yield"`).
`type`	Character. Curve type: `"quadratic"` (default), `"linear"`, or `"mitscherlich"`.

Value

A list of class `"nrm_response_curve"` with:

`model` The fitted model object.

`type` The curve type used.

`optimum` Estimated optimum input level (for quadratic).

`r_squared` Model R-squared.

Examples

```
data(nrm_example)
rc <- nrm_response_curve(nrm_example,
        input_var = "N", response_var = "crop_yield", type = "quadratic")
nrm_plot(rc)
```

`nrm_sem` *Structural Equation Modelling (SEM)*

Description

Fits a Structural Equation Model (SEM) using `sem`. Intended for path analysis of interlinked Natural Resource Management (NRM) processes (e.g. soil organic carbon mediating the effect of fertiliser on yield).

Usage

```
nrm_sem(data, model, ...)
```

Arguments

`data`	A data frame.
`model`	Character string in **lavaan** model syntax.
`...`	Additional arguments passed to `sem`.

Value

A list of class `"nrm_sem"` containing:

`model` The fitted `lavaan` object.

`fit_indices` Comparative Fit Index (CFI), Tucker-Lewis Index (TLI), Root Mean Square Error of Approximation (RMSEA), and Standardised Root Mean Square Residual (SRMR).

`parameter_estimates` Standardised path coefficients.

References

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1–36. doi:10.18637/jss.v048.i02

Examples

```
if (requireNamespace("lavaan", quietly = TRUE)) {
  model_syn <- "
    crop_yield ~ N + soil_OC
    soil_OC    ~ biomass + N
  "
  data(nrm_example)
  sem_res <- nrm_sem(nrm_example, model = model_syn)
  nrm_summary(sem_res)
}
```

`nrm_sens_slope` *Sen's slope estimator*

Description

Computes Sen's slope as a robust, nonparametric estimate of the linear trend magnitude in a time series.

Usage

```
nrm_sens_slope(data, time_var = "year", value_var)
```

Arguments

`data`	A data frame containing at least the columns specified by `time_var` and `value_var`.
`time_var`	Character. Name of the column holding the time index (e.g. `"year"`).
`value_var`	Character. Name of the column holding the response variable (e.g. `"crop_yield"`).

Value

A list of class `"nrm_sens_slope"` with components:

`slope` Sen's slope (units of `value_var` per time step).

`intercept` Estimated intercept.

`conf_int` 95 % confidence interval for the slope.

`raw` Full output from `sens.slope`.

References

Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association*, 63, 1379–1389.

Examples

```
data(nrm_example)
ss <- nrm_sens_slope(nrm_example, time_var = "year",
                     value_var = "crop_yield")
print(ss)
```

`nrm_structural_break` *Structural break detection*

Description

Detects structural breakpoints in a time series using the Bai-Perron framework implemented in **strucchange**.

Usage

```
nrm_structural_break(data, time_var = "year", value_var, max_breaks = 5)
```

Arguments

`data`	A data frame containing at least the columns specified by `time_var` and `value_var`.
`time_var`	Character. Name of the column holding the time index (e.g. `"year"`).
`value_var`	Character. Name of the column holding the response variable (e.g. `"crop_yield"`).
`max_breaks`	Integer. Maximum number of breaks to test. Default `5`.

Value

A list of class `"nrm_structural_break"` with components:

`n_breaks` Number of detected breaks.

`break_dates` Time values at which breaks occur (extracted from `time_var`).

`raw` Full `breakpoints` object.

References

Bai, J., & Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18, 1–22.

Examples

```
data(nrm_example)
sb <- nrm_structural_break(nrm_example, time_var = "year",
                           value_var = "crop_yield")
print(sb)
```

nrm_summary *Generic summary for NRMstatsML objects*

Description

Prints a concise summary for any NRMstatsML result object.

Usage

```
nrm_summary(object, ...)
```

Arguments

object	An NRMstatsML result object (e.g. "nrm_trend", "nrm_automl", etc.).
...	Currently unused.

Value

The object invisibly.

nrm_trend *Comprehensive trend analysis for NRM time series*

Description

Runs Mann-Kendall test, Sen's slope estimator, and optional structural-break detection on a univariate time series extracted from a data frame.

Usage

```
nrm_trend(data, time_var = "year", value_var, breaks = TRUE, alpha = 0.05)
```

Arguments

data	A data frame containing at least the columns specified by time_var and value_var.
time_var	Character. Name of the column holding the time index (e.g. "year").
value_var	Character. Name of the column holding the response variable (e.g. "crop_yield").
breaks	Logical. If TRUE (default), structural breaks are detected with breakpoints.
alpha	Numeric. Significance level for the Mann-Kendall test. Default 0.05.

Value

A list of class "nrm_trend" with components:

mann_kendall Output of nrm_mann_kendall.

sens_slope Output of nrm_sens_slope.

structural_breaks Output of nrm_structural_break or NULL when breaks = FALSE.

data The data argument, for downstream use.

call The matched call.

See Also

nrm_mann_kendall, nrm_sens_slope, nrm_structural_break

Examples

```
data(nrm_example)
result <- nrm_trend(nrm_example, time_var = "year", value_var = "crop_yield")
print(result)
nrm_plot(result)
```

---

nrm_uncertainty *Uncertainty analysis for NRM model outputs*

Description

Runs bootstrap resampling or Monte Carlo simulation to quantify uncertainty around a statistic of interest computed on NRM data.

Usage

```
nrm_uncertainty(
  data,
  stat_fn,
  method = "bootstrap",
  n_iter = 1000L,
  alpha = 0.05,
  ...
)
```

Arguments

data	A data frame.
stat_fn	A function that takes a data frame and returns a single numeric value (or named numeric vector). Must accept a data argument as its first positional argument.
method	Character. "bootstrap" (default) or "monte_carlo".

n_iter	Integer. Number of resamples / simulations. Default 1000.
alpha	Numeric. Significance level for confidence intervals. Default 0.05.
...	Additional arguments passed to stat_fn.

Value

A list of class "nrm_uncertainty" with:

mean Mean of the resampled statistic.

ci Confidence interval at 1 - alpha level.

sd Standard deviation of the resampled statistic.

method Method used.

Examples

```
data(nrm_example)
unc <- nrm_uncertainty(nrm_example,
         stat_fn = function(d) mean(d$crop_yield),
         method = "bootstrap", n_iter = 500)
print(unc)
```

Index