

Package ‘margEVT’

September 23, 2026

Type Package

Title Regularized Point Processes and Stochastic Marginalization for
Extremes

Version 0.3.0

Description Implements a non-stationary extreme value analysis framework by coupling a covariate-driven Non-Homogeneous Poisson Process (NHPP) with Elastic-Net regularization and analytical gradients. Provides methods for estimating conditional return levels and unconditional (marginalized) return levels via parametric stochastic integration over stable Vector Autoregressive VAR(p) or univariate autoregressive covariate trajectories, or non-parametric annual-block resampling. Supports block-specific penalty controls, operational active-set thresholds, conditional parametric bootstrap inference, and walk-forward assessment.

License GPL (>= 3)

Encoding UTF-8

URL <https://github.com/rodrigossqrt3/margEVT>

BugReports <https://github.com/rodrigossqrt3/margEVT/issues>

RoxygenNote 7.3.3

Imports stats, utils, vars

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Rodrigo Fonseca Villa [aut, cre] (ORCID:
<<https://orcid.org/0009-0005-2938-2270>>)

Maintainer Rodrigo Fonseca Villa <rodrigo03.villa@gmail.com>

Repository CRAN

Date/Publication 2026-09-23 17:00:02 UTC

Contents

active_covariates	2
backtest	3
bic_nhpp	4
bootstrap_coef	5
bootstrap_rl	5
build_cov_annual	6
build_design_matrices	7
fit_nhpp	8
fit_var_generator	10
is_nhpp_fit	10
marginalize	11
n_exceedances	12
plot.nhpp_fit	13
pp_grad	13
pp_nllh	14
predict_params	15
rl_table	16
simulate_covariates	16
summary.nhpp_fit	17
Index	18

active_covariates	<i>Extract names of non-seasonal covariates from a fitted model</i>
-------------------	---

Description

Returns the covariate names that appear in the model with a non-negligible coefficient — i.e. those that actually affect the parameters and need to be supplied during marginalization.

Usage

```
active_covariates(fit, free_cols = NULL, tol = NULL)
```

Arguments

fit	An nhpp_fit object.
free_cols	Character vector of column names to always exclude from the result (intercept and seasonality terms are always excluded). Default NULL.
tol	Numeric. Coefficients smaller than this in absolute value are treated as zero. If NULL, inherits fit\$active_tol; legacy objects fall back to 1e-2.

Value

Character vector of active covariate names, or NULL if the model has no active non-seasonal covariates.

backtest

*Walk-forward backtesting for NHPP extreme value models***Description**

Evaluates predictive performance of a fitted `nhpp_fit` model via walk-forward cross-validation. For each window, the model is refit on training data at the original penalty and lambda, return levels are computed, and exceedance rates are compared to nominal rates via binomial tests. A Kolmogorov-Smirnov comparison on tail-conditional PIT values is returned as an exploratory diagnostic; it is not a formal test of global predictive calibration. Validation years with fewer than `min_obs_year` observed values are excluded from binomial denominators.

Usage

```
backtest(
  fit,
  data,
  varname,
  TRs = c(2, 5, 10),
  n_obs = NULL,
  period = NULL,
  window_years = 5L,
  min_train_years = 10L,
  n_boot = 200L,
  interactions = list(),
  year_col = "year",
  min_obs_year = NULL,
  seed = 2024L,
  verbose = TRUE
)
```

Arguments

<code>fit</code>	An <code>nhpp_fit</code> object.
<code>data</code>	The data frame used to fit <code>fit</code> . Must contain columns <code>y</code> , <code>year</code> (or <code>year_col</code>), and all active covariates.
<code>varname</code>	Character. Name of the column in <code>data</code> containing the raw (non-declustered) series, used to compute observed annual maxima for validation.
<code>TRs</code>	Numeric vector of return periods to evaluate. Default <code>c(2, 5, 10)</code> .
<code>n_obs</code>	Integer observations per year. If <code>NULL</code> , uses the rounded <code>fit\$obs_per_year</code> value.
<code>period</code>	Numeric seasonal period in observations. If <code>NULL</code> , inherits <code>fit\$obs_per_year</code> .
<code>window_years</code>	Integer. Width of each validation window in years. Default 5L.
<code>min_train_years</code>	Integer. Minimum number of training years before the first validation window. Default 10L.

n_boot	Integer. Bootstrap years for covariate marginalization within each window. Default 200L.
interactions	Named list of interactions passed to <code>build_cov_annual</code> . Default <code>list()</code> .
year_col	Character. Name of the year column. Default "year".
min_obs_year	Integer. Minimum observations in a validation year for it to be included. Default <code>as.integer(n_obs * 0.8)</code> .
seed	Integer. Seed used for empirical annual-block sampling. Default 2024L. Each validation window uses a distinct derived seed.
verbose	Logical. Print progress. Default TRUE.

Value

A list with three elements:

results	Data frame with one row per validation year, containing the observed annual maximum, PIT value, and exceedance indicators for each return period.
binom_tests	Data frame with nominal binomial comparisons for each TR. The <code>not_rejected</code> field is the recommended interpretation; <code>calibrated</code> is retained as a legacy alias.
ks_test	Data frame with the exploratory tail-conditional PIT KS comparison, including <code>not_rejected</code> and the legacy <code>calibrated</code> alias, or NULL if fewer than 5 validation years had events.

bic_nhpp	<i>Compute BIC for a fitted nhpp_fit model</i>
----------	--

Description

Compute BIC for a fitted `nhpp_fit` model

Usage

```
bic_nhpp(fit, tol = NULL)
```

Arguments

fit	An <code>nhpp_fit</code> object.
tol	Numeric. Coefficients smaller than this are counted as zero for the active parameter count. If NULL, inherits <code>fit\$active_tol</code> ; legacy objects fall back to <code>1e-2</code> .

Value

Numeric scalar. BIC value.

bootstrap_coef	<i>Parametric bootstrap confidence intervals for model coefficients</i>
----------------	---

Description

Simulates R bootstrap samples, refits at the original penalty and lambda, and returns empirical quantile CIs for each coefficient.

Usage

```
bootstrap_coef(fit, data, R = 200L, level = 0.95, seed = 42L, verbose = TRUE)
```

Arguments

fit	An nhpp_fit object.
data	The data frame used to fit fit. Must contain column y and all covariates.
R	Integer. Number of bootstrap replicates. Default 200L.
level	Numeric. Confidence level. Default 0.95.
seed	Integer. Random seed. Default 42L.
verbose	Logical. Print progress. Default TRUE.

Value

A data frame with columns parameter, estimate, CI_low, CI_high, n_ok, with one row for every coefficient in the fitted model.

bootstrap_rl	<i>Parametric bootstrap confidence intervals for return levels</i>
--------------	--

Description

Simulates R bootstrap samples from the fitted model, refits at the original model design, penalty, lambda, and activity tolerance, recomputes return levels via [marginalize](#), and returns empirical quantile intervals. Exceedance count and times are held fixed, so the result is a nominal conditional summary rather than unconditional or post-selection inference.

Usage

```
bootstrap_rl(  
  fit,  
  data,  
  TRs = c(2, 5, 10, 20, 50, 100),  
  R = 200L,  
  approach = "C",  
  marginalize_args = list(),  
  level = 0.95,  
  seed = 42L,  
  verbose = TRUE  
)
```

Arguments

fit	An nhpp_fit object.
data	The data frame used to fit fit. Must contain column y and all covariates.
TRs	Numeric vector of return periods. Default c(2, 5, 10, 20, 50, 100).
R	Integer. Number of bootstrap replicates. Default 200L.
approach	Character. Which marginalization approach to use for each replicate: "A", "B", or "C". Default "C".
marginalize_args	Named list of additional arguments passed to marginalize (e.g. n_boot, seed, scenarios, interactions).
level	Numeric. Confidence level. Default 0.95.
seed	Integer. Random seed. Default 42L.
verbose	Logical. Print progress. Default TRUE.

Value

A data frame with columns TR, RL_est (the return level from the original fitted model under the requested approach), CI_low, CI_high, n_ok (number of successful replicates).

build_cov_annual	<i>Build a covariate data frame for marginalization</i>
------------------	---

Description

Constructs a data frame with n_obs rows containing all columns required by a fitted nhpp_fit model. Seasonality columns (cos1, sen1, cos2, sen2) are computed automatically using period. Any other required column is filled from cov_vals, or set to zero with a warning if absent.

Usage

```
build_cov_annual(
  fit,
  cov_vals = list(),
  n_obs = NULL,
  period = NULL,
  interactions = list()
)
```

Arguments

<code>fit</code>	An <code>nhpp_fit</code> object.
<code>cov_vals</code>	Named list of covariate values. Each element is either a scalar (replicated to <code>n_obs</code> rows) or a vector of length <code>n_obs</code> .
<code>n_obs</code>	Integer number of rows (time points) to generate. If <code>NULL</code> , uses the rounded <code>fit\$obs_per_year</code> value.
<code>period</code>	Numeric seasonal period in observations. If <code>NULL</code> , inherits <code>fit\$obs_per_year</code> ; legacy objects fall back to 365.25.
<code>interactions</code>	Named list of length-2 character vectors declaring interaction columns to compute. Each element is <code>c("col_A", "col_B")</code> and the interaction column is their product. The name of the list element becomes the column name. Example: <code>list(AB = c("A", "B"))</code> adds column $AB = A * B$.

Value

A data frame with `n_obs` rows and all columns required by `fit`.

`build_design_matrices` *Build design matrices for the NHPP point process model*

Description

Build design matrices for the NHPP point process model

Usage

```
build_design_matrices(
  df,
  loc_vars = NULL,
  scale_vars = NULL,
  shape_vars = NULL,
  free_vars = NULL
)
```

Arguments

df	A data frame containing all required columns.
loc_vars	Character vector of covariate names for the location (μ) parameter.
scale_vars	Character vector of covariate names for the scale (σ) parameter.
shape_vars	Character vector of covariate names for the shape (ξ) parameter.
free_vars	Character vector of column names that should never be penalized (e.g. seasonality terms like <code>c("cos1", "sen1")</code>). Defaults to NULL.

Value

A list with elements:

X_mu	Design matrix for location.
X_sigma	Design matrix for scale.
X_xi	Design matrix for shape.
idx_pen_mu	Integer indices of penalized columns in X_mu.
idx_pen_sigma	Integer indices of penalized columns in X_sigma.
idx_pen_xi	Integer indices of penalized columns in X_xi.

fit_nhpp

Fit a non-homogeneous point process model for extremes

Description

Fits a peaks-over-threshold point process model via penalized maximum likelihood with BFGS optimization and exact analytical gradients.

Usage

```
fit_nhpp(
  df,
  threshold,
  loc_vars = NULL,
  scale_vars = NULL,
  shape_vars = NULL,
  free_vars = NULL,
  penalty = c("none", "lasso", "ridge", "elnet"),
  alpha = 0.5,
  lambda = "bic",
  penalize_shape = TRUE,
  obs_per_year = 365.25,
  active_tol = 0.01,
  lambda_scaling = c("gradient", "common"),
  maxit = 10000L,
  calc_hessian = FALSE,
  verbose = TRUE
)
```

Arguments

df	A data frame containing the response and all covariates. Must have a column named y (the observations).
threshold	Numeric scalar. The extreme value threshold u.
loc_vars	Character vector of covariate names for the location parameter. NULL for stationary location.
scale_vars	Character vector of covariate names for the scale parameter. NULL for stationary scale.
shape_vars	Character vector of covariate names for the shape parameter. NULL for stationary shape.
free_vars	Character vector of covariate names that are never penalized (e.g. seasonality terms). NULL by default.
penalty	Character. Penalty type: "none" (pure MLE), "lasso" (L1), "ridge" (L2), or "elnet" (elastic net, requires alpha).
alpha	Numeric in [0, 1]. Elastic-net mixing parameter. Only used when penalty = "elnet". 1 = LASSO, 0 = ridge. Scalar or named vector c(mu = , sigma = , xi =).
lambda	Numeric scalar or named vector c(mu=, sigma=, xi=), or "bic" to select automatically via a two-phase BIC grid search. Ignored when penalty = "none".
penalize_shape	Logical. Penalize shape parameter covariates? Default TRUE.
obs_per_year	Numeric. Observations per year. E.g. 365.25 for daily data, 52 for weekly. Default 365.25.
active_tol	Positive numeric tolerance used for BIC complexity counting, active-covariate extraction, and default coefficient reporting. Default 1e-2.
lambda_scaling	Character. Scaling used when lambda = "bic": "gradient" calibrates the scale and shape penalties relative to the location-block score magnitude; "common" applies the same scalar lambda to all parameter blocks. Ignored for numeric lambda.
maxit	Integer. Maximum optimizer iterations. Default 10000L.
calc_hessian	Logical. Compute Hessian at solution? Needed for delta-method standard errors. Default FALSE.
verbose	Logical. Print progress during BIC grid search and convergence warnings. Default TRUE.

Value

An object of class nhpp_fit.

fit_var_generator	<i>Fit a generic VAR generator over a model's active covariates</i>
-------------------	---

Description

Deseasonalizes each active covariate column in data (regressing on cos1, sen1, cos2, sen2), then fits a multivariate VAR to the standardized residuals. No covariate names are hardcoded: whatever `active_covariates` returns (or whatever you pass in vars) is treated as a final, ready-to-model column.

Usage

```
fit_var_generator(fit, data, vars = NULL, period = NULL, lag_max = NULL)
```

Arguments

fit	An nhpp_fit object.
data	Data frame used to fit fit (or any data frame containing the same covariate columns, plus seasonal harmonics if you already have them).
vars	Character vector of covariate names to model. Default NULL: uses <code>active_covariates(fit)</code> , intersected with <code>names(data)</code> .
period	Numeric seasonal period in observations. If NULL, inherits <code>fit\$obs_per_year</code> ; legacy objects fall back to 365.25.
lag_max	Integer. Max VAR lag considered by VARselect. Default NULL: <code>min(5, floor(n / (3*K)))</code> .

Value

An object of class `nhpp_var_generator`, including the fitted root moduli and spectral radius, or NULL (with a warning) if the model has no active covariates. An unstable selected VAR causes an error.

is_nhpp_fit	<i>Check if an object is an nhpp_fit</i>
-------------	--

Description

Check if an object is an nhpp_fit

Usage

```
is_nhpp_fit(x)
```

Arguments

x Any R object.

Value

Logical.

marginalize	<i>Compute unconditional return levels by marginalizing over covariates</i>
-------------	---

Description

Given a fitted `nhpp_fit` model, computes return levels by integrating out covariate uncertainty under one or more approaches:

Usage

```
marginalize(
  fit,
  data,
  TRs = c(2, 5, 10, 20, 50, 100),
  n_obs = NULL,
  period = NULL,
  approaches = c("A", "C"),
  scenarios = NULL,
  mc_sample = NULL,
  interactions = list(),
  year_col = "year",
  n_boot = 500L,
  seed = 2024L,
  tol_root = 1e-04,
  z_hi = NULL
)
```

Arguments

fit	An <code>nhpp_fit</code> object.
data	The data frame used to fit <code>fit</code> . Required for approach C and for computing <code>z_hi</code> . Must contain a column <code>year</code> (or the name given in <code>year_col</code>) and all active covariates.
TRs	Numeric vector of return periods in years. Default <code>c(2, 5, 10, 20, 50, 100)</code> .
n_obs	Integer observations per year. If <code>NULL</code> , uses the rounded <code>fit\$obs_per_year</code> value.
period	Numeric seasonal period in observations. If <code>NULL</code> , inherits <code>fit\$obs_per_year</code> .
approaches	Character vector. Which approaches to run. Any subset of <code>c("A", "B", "C")</code> . Default <code>c("A", "C")</code> .

scenarios	Named list of covariate scenarios for approach A. Each element is itself a named list of scalar covariate values, passed to <code>build_cov_annual</code> . If NULL, a single scenario with all covariates set to zero is used.
mc_sample	List of data frames for approach B. Each element is one simulated year of covariates (n_obs rows). Required if "B" %in% approaches.
interactions	Named list of interactions passed to <code>build_cov_annual</code> . Default <code>list()</code> .
year_col	Name of the year column in data. Default "year".
n_boot	Integer. Number of bootstrap years for approach C. Default 500L.
seed	Integer. Random seed for approach C. Default 2024L.
tol_root	Numeric. Root-finding tolerance. Default 1e-4.
z_hi	Numeric. Upper search bound for return levels. If NULL, set to $3 * \max(\text{data}\$y, \text{na.rm} = \text{TRUE})$.

Details

- A — conditional** Covariates fixed at user-specified scenarios (e.g. quantiles of observed distributions). Fast; useful for sensitivity analysis.
- B — parametric** Covariates drawn from a user-supplied Monte Carlo sample (e.g. from a fitted VAR or any stochastic generator). The sample is a list of n_obs-row data frames, one per simulated year.
- C — empirical** Block bootstrap over observed covariate years. Resamples whole years from data to preserve temporal structure.

Value

A data frame with columns `approach`, `scenario`, `TR`, `RL`.

n_exceedances	<i>Extract the number of exceedances from a fitted model</i>
---------------	--

Description

Extract the number of exceedances from a fitted model

Usage

```
n_exceedances(fit, y)
```

Arguments

<code>fit</code>	An <code>nhpp_fit</code> object.
<code>y</code>	Numeric vector of observations used to fit the model.

Value

Integer. Number of observations exceeding the threshold.

plot.nhpp_fit	<i>Plot diagnostics for a fitted nhpp_fit object</i>
---------------	--

Description

Generates primary diagnostic plots for a fitted nhpp_fit model, including the cumulative intensity measure over time and parameter profiles.

Usage

```
## S3 method for class 'nhpp_fit'
plot(x, type = c("intensity", "fitted"), ...)
```

Arguments

x	An nhpp_fit object.
type	Character. Type of plot to generate: "intensity" (default) for the cumulative intensity measure $\hat{\Lambda}(0, t)$, or "fitted" for time-varying parameter profiles $(\mu(t), \sigma(t), \text{ and } \xi(t))$.
...	Additional graphical parameters passed to plot .

Value

Invisibly returns NULL, called for side effects.

pp_grad	<i>Analytical gradient of the penalized negative log-likelihood</i>
---------	---

Description

Analytical gradient of the penalized negative log-likelihood

Usage

```
pp_grad(
  par,
  dm,
  y,
  threshold,
  lambda = 0.1,
  alpha = 0.5,
  pen_xi = TRUE,
  eps_l1 = 1e-10,
  obs_per_year = 365.25
)
```

Arguments

par	Numeric vector of parameters: c(beta_mu, beta_sigma, beta_xi).
dm	Design matrix list from build_design_matrices .
y	Numeric vector of observations.
threshold	Numeric scalar. The extreme value threshold u.
lambda	Penalty strength. Either a scalar or named vector c(mu=, sigma=, xi=).
alpha	Elastic-net mixing parameter in [0,1]. 1 = LASSO, 0 = ridge. Either a scalar or named vector c(mu=, sigma=, xi=).
pen_xi	Logical. Should the shape parameter covariates be penalized?
eps_l1	Small positive constant for smooth L1 approximation.
obs_per_year	Number of observations per year (e.g. 365.25 for daily).

Value

Numeric vector of length length(par).

pp_nllh

Penalized negative log-likelihood for the NHPP model

Description

Penalized negative log-likelihood for the NHPP model

Usage

```
pp_nllh(
  par,
  dm,
  y,
  threshold,
  lambda = 0.1,
  alpha = 0.5,
  pen_xi = TRUE,
  eps_l1 = 1e-10,
  obs_per_year = 365.25
)
```

Arguments

par	Numeric vector of parameters: c(beta_mu, beta_sigma, beta_xi).
dm	Design matrix list from build_design_matrices .
y	Numeric vector of observations.
threshold	Numeric scalar. The extreme value threshold u.

lambda	Penalty strength. Either a scalar or named vector c(mu=, sigma=, xi=).
alpha	Elastic-net mixing parameter in [0,1]. 1 = LASSO, 0 = ridge. Either a scalar or named vector c(mu=, sigma=, xi=).
pen_xi	Logical. Should the shape parameter covariates be penalized?
eps_l1	Small positive constant for smooth L1 approximation.
obs_per_year	Number of observations per year (e.g. 365.25 for daily).

Value

Scalar. The penalized negative log-likelihood value.

predict_params	<i>Compute time-varying GEV parameters from a fitted model</i>
----------------	--

Description

Given an nhpp_fit object and a data frame of covariate values, returns the implied location, scale, and shape parameters at each row.

Usage

```
predict_params(fit, newdata = NULL, operational = TRUE)
```

Arguments

fit	An object of class nhpp_fit.
newdata	A data frame with one row per time point. Must contain all columns that appear in the model's design matrices (except the intercept). If NULL, returns the fitted values stored in fit\$fitted.
operational	Logical. If TRUE (default), coefficients below the fitted activity tolerance are set to zero before prediction. Set to FALSE to use the unthresholded smooth optimizer solution.

Value

A list with three numeric vectors of length nrow(newdata):

mu	Location parameter.
sigma	Scale parameter (always positive).
xi	Shape parameter.

rl_table	<i>Wide-format return level table from marginalize() output</i>
----------	---

Description

Converts the long-format output of `marginalize` into a wide table with one row per scenario and one column per return period.

Usage

```
rl_table(marg_result)
```

Arguments

`marg_result` Data frame returned by `marginalize`.

Value

A wide data frame.

simulate_covariates	<i>Simulate stationary covariate trajectories from a VAR generator</i>
---------------------	--

Description

Simulate stationary covariate trajectories from a VAR generator

Usage

```
simulate_covariates(
  generator,
  n_mc,
  n_obs = NULL,
  burn_in = 300L,
  initialization = c("stationary", "burnin"),
  seed = NULL
)
```

Arguments

<code>generator</code>	Object from <code>fit_var_generator</code> . If NULL, returns NULL (stationary/no-covariate model).
<code>n_mc</code>	Integer. Number of independently simulated annual trajectories.
<code>n_obs</code>	Integer observations per simulated year. If NULL, uses the rounded seasonal period stored in generator.

burn_in	Integer. Burn-in applied separately when initialization = "burnin". Default 300L.
initialization	Character. "stationary" (default) draws the initial companion state from the fitted stationary Gaussian law; "burnin" starts at zero and applies burn_in transitions.
seed	Integer. Random seed. Default NULL (not set).

Value

A list of length n_mc, each element an n_obs-row data frame with one column per active covariate - directly usable as mc_sample in [marginalize](#) (approach B).

summary.nhpp_fit	<i>Summarise a fitted nhpp_fit model</i>
------------------	--

Description

Prints a structured summary of a fitted nhpp_fit object, including the penalty used, active coefficients, and basic model diagnostics.

Usage

```
## S3 method for class 'nhpp_fit'
summary(object, tol = NULL, ...)
```

Arguments

object	An nhpp_fit object.
tol	Numeric. Coefficients smaller than this in absolute value are treated as zero. If NULL, inherits object\$active_tol; legacy objects fall back to 1e-2.
...	Ignored.

Value

Invisibly returns the input nhpp_fit object.

Index

active_covariates, [2](#), [10](#)

backtest, [3](#)

bic_nhpp, [4](#)

bootstrap_coef, [5](#)

bootstrap_rl, [5](#)

build_cov_annual, [4](#), [6](#), [12](#)

build_design_matrices, [7](#), [14](#)

fit_nhpp, [8](#)

fit_var_generator, [10](#), [16](#)

is_nhpp_fit, [10](#)

marginalize, [5](#), [6](#), [11](#), [16](#), [17](#)

n_exceedances, [12](#)

plot, [13](#)

plot_nhpp_fit, [13](#)

pp_grad, [13](#)

pp_nllh, [14](#)

predict_params, [15](#)

rl_table, [16](#)

simulate_covariates, [16](#)

summary_nhpp_fit, [17](#)