

Package: margEVT (via r-universe)

July 23, 2026

Type Package

Title Regularized Point Processes and Stochastic Marginalization for Extremes

Version 0.1.0

Description Implements a non-stationary extreme value analysis framework by coupling a covariate-driven Non-Homogeneous Poisson Process (NHPP) with Elastic-Net regularization and exact analytical gradients. Provides methodologies for estimating conditional return levels and unconditional (marginalized) return levels via parametric stochastic integration over Vector Autoregressive VAR(p) covariate trajectories, or non-parametric block bootstrapping. Methodologies are based on Villa (2026)
<<https://sabi.ufrgs.br/>> ``A Novel Regularized Point Process and Stochastic Marginalization Framework for Return Level Inference under Covariate-Driven Extremes" (Master's dissertation, Universidade Federal do Rio Grande do Sul).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, utils, vars

Suggests dplyr, tidyr, extRemes, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Rodrigo Fonseca Villa [aut, cre] (ORCID: <<https://orcid.org/0009-0005-2938-2270>>), Flavio Ziegelmann [ths] (ORCID: <<https://orcid.org/0000-0001-8899-350X>>)

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

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-23 14:39:17 UTC

RemoteUrl <https://github.com/cran/margEVT>

RemoteRef HEAD
RemoteSha 2e72ae5f8031a2a9ce1befc0f3ecbf2b6912612a

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	9
is_nhpp_fit	10
marginalize	10
n_exceedances	12
pp_grad	12
pp_nllh	13
predict_params	14
rl_table	15
simulate_covariates	15
summary.nhpp_fit	16
Index	17

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 = 1e-04)
```

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 (LASSO shrinkage). Default 1e-4.

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. Probabilistic calibration is assessed via a Kolmogorov-Smirnov test on the PIT values.

Usage

```
backtest(
  fit,
  data,
  varname,
  TRs = c(2, 5, 10),
  n_obs = 365L,
  window_years = 5L,
  min_train_years = 10L,
  n_boot = 200L,
  interactions = list(),
  year_col = "year",
  min_obs_year = NULL,
  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. Default 365L.
<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> .
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 binomial test results for each TR.
ks_test	Data frame with KS test result for PIT calibration, 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 = 0.01)
```

Arguments

fit	An <code>nhpp_fit</code> object.
tol	Numeric. Coefficients smaller than this are counted as zero for the active parameter count. Default <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. Only parameters with non-negligible estimates ($|\text{estimate}| > 1e-4$) are returned.

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

Description

Simulates R bootstrap samples from the fitted model, refits at the original penalty and lambda, recomputes return levels via [marginalize](#), and returns empirical quantile CIs.

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, 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 from n_obs. 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 = 365L,
  period = 365.25,
  interactions = list()
)
```

Arguments

fit	An nhpp_fit object.
cov_vals	Named list of covariate values. Each element is either a scalar (replicated to n_obs rows) or a vector of length n_obs.
n_obs	Integer. Number of rows (time points) to generate. Typically 365L for one year of daily data.

period	Numeric. Period for the seasonal harmonics in days. Default 365.25.
interactions	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

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

Value

A list with elements:

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

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,
  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.
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 = 365.25, 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. Default 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, or NULL (with a warning) if the model has no active covariates.

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 = 365L,
  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

<code>fit</code>	An <code>nhpp_fit</code> object.
<code>data</code>	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.
<code>TRs</code>	Numeric vector of return periods in years. Default <code>c(2, 5, 10, 20, 50, 100)</code> .
<code>n_obs</code>	Integer. Observations per year. Default 365L.
<code>approaches</code>	Character vector. Which approaches to run. Any subset of <code>c("A", "B", "C")</code> . Default <code>c("A", "C")</code> .
<code>scenarios</code>	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 <code>NULL</code> , a single scenario with all covariates set to zero is used.
<code>mc_sample</code>	List of data frames for approach B. Each element is one simulated year of covariates (<code>n_obs</code> rows). Required if "B" %in% <code>approaches</code> .
<code>interactions</code>	Named list of interactions passed to <code>build_cov_annual</code> . Default <code>list()</code> .
<code>year_col</code>	Name of the year column in <code>data</code> . Default <code>"year"</code> .
<code>n_boot</code>	Integer. Number of bootstrap years for approach C. Default 500L.
<code>seed</code>	Integer. Random seed for approach C. Default 2024L.
<code>tol_root</code>	Numeric. Root-finding tolerance. Default <code>1e-4</code> .
<code>z_hi</code>	Numeric. Upper search bound for return levels. If <code>NULL</code> , set to <code>3 * max(data\$y, na.rm = TRUE)</code> .

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

fit	An nhpp_fit object.
y	Numeric vector of observations used to fit the model.

Value

Integer. Number of observations exceeding the threshold.

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	<i>Penalized negative log-likelihood for the NHPP model</i>
---------	---

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)
```

Arguments

fit	An object of class <code>nhpp_fit</code> .
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 <code>NULL</code> , returns the fitted values stored in <code>fit\$fitted</code> .

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 = 365L, burn_in = 300L, seed = NULL)
```

Arguments

generator	Object from fit_var_generator . If NULL, returns NULL (stationary/no-covariate model).
n_mc	Integer. Number of simulated years.
n_obs	Integer. Observations per simulated year. Default 365L.
burn_in	Integer. Burn-in to discard transient VAR dynamics. Default 300L.
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 = 1e-04, ...)
```

Arguments

object	An nhpp_fit object.
tol	Numeric. Coefficients smaller than this in absolute value are treated as zero. Default 1e-4.
...	Ignored.

Value

Invisibly returns the input nhpp_fit object.

Index

`active_covariates`, [2](#), [9](#)

`backtest`, [3](#)

`bic_nhpp`, [4](#)

`bootstrap_coef`, [5](#)

`bootstrap_rl`, [5](#)

`build_cov_annual`, [4](#), [6](#), [11](#)

`build_design_matrices`, [7](#), [13](#)

`fit_nhpp`, [8](#)

`fit_var_generator`, [9](#), [15](#)

`is_nhpp_fit`, [10](#)

`marginalize`, [5](#), [6](#), [10](#), [15](#)

`n_exceedances`, [12](#)

`pp_grad`, [12](#)

`pp_nllh`, [13](#)

`predict_params`, [14](#)

`rl_table`, [15](#)

`simulate_covariates`, [15](#)

`summary.nhpp_fit`, [16](#)