Package: precisionenvelope (via r-universe)

November 26, 2026

Type Package

Title Sharp Precision Envelopes for Outcomes Missing at Random

Version 0.1.0

Description Computes the range of precision that can survive when an outcome or an all-or-none block of variables is missing at random. For a stated overall observation rate and a lower bound on the observation probability, the package returns the smallest and largest efficient variance that any admissible missingness mechanism can produce, the corresponding information retention and standard error inflation, and whether a precision requirement is guaranteed, depends on the allocation of missingness, or cannot be met. It also plans minimum-cost targeted recovery of missing outcomes, and estimates the envelope from data on a partially observed continuous outcome by cross-fitted one-step estimation with a refitted bootstrap band over a prespecified grid of observation rates. The method is described in Nakamura (2026) <doi:10.3390/stats9050110>.

License BSD_3_clause + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports graphics, grDevices, stats

Suggests knitr, rmarkdown, survival, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Satoshi Nakamura [aut, cre]

Maintainer Satoshi Nakamura <s-nakamura@med.mie-u.ac.jp>

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

Date/Publication 2026-11-26 01:52:21 UTC

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

RemoteRef HEAD

RemoteSha 678df4bf7b32eff41181e769f7528d3398dee0d3

Contents

precisionenvelope-package 2
envelope_bootstrap 3
envelope_fit 5
envelope_learners 8
precision_envelope 9
recovery_plan 12
residual_law_gamma 13

Index **15**

`precisionenvelope-package`
Sharp Precision Envelopes for Outcomes Missing at Random

Description

When an outcome, or an all-or-none block of variables, is missing at random (MAR) given always observed covariates W, the overall missingness rate does not determine how much precision survives. Two mechanisms with the same rate can leave very different efficient variances, because the loss depends on which units go missing. This package computes the range of efficient variance, and of information retention, that the MAR mechanisms with a given overall observation rate p and a lower bound π_L on the observation probability can produce. The range is sharp: both endpoints are attained or approached by admissible mechanisms.

For a scalar target θ with complete-data efficient influence function $\varphi_\theta(Z)$, write $a_\theta(W) = E\{\varphi_\theta(Z) \mid W\}$ and $b_\theta(W) = \mathrm{Var}\{\varphi_\theta(Z) \mid W\}$, $A_\theta = \mathrm{Var}\{a_\theta(W)\}$ and $B_\theta = E\{b_\theta(W)\}$. Under a mechanism $\pi(W)$ the observed-data efficient variance is $A_\theta + C_\theta(\pi)$ with $C_\theta(\pi) = E\{b_\theta(W)/\pi(W)\}$, and the information retention is $\rho_\theta = (A_\theta + B_\theta)/\{A_\theta + C_\theta(\pi)\}$. The package reports the smallest and largest $C_\theta(\pi)$ over the admissible class.

The package has three layers.

Planning `precision_envelope` computes the envelope from A_θ (or the share R^2 of the influence-function variance explained by W) and a law for $b_\theta(W)$. No data are needed. With a precision requirement it classifies each rate as guaranteed, allocation-sensitive or inadmissible and returns the threshold missingness rates. `residual_law_gamma` supplies a convenient law for a heteroskedastic $b_\theta(W)$.

Recovery `recovery_plan` gives the minimum-cost targeted recovery of missing outcomes that meets a requirement.

Estimation `envelope_fit` estimates the envelope from data on a partially observed continuous outcome, for the mean of that outcome, and `envelope_bootstrap` adds a simultaneous refitted-bootstrap band over the grid of rates, certified phases and confidence intervals for the threshold rates.

The package does not impute missing values, does not improve the estimate of θ, and does not test MAR. It states how much precision can survive under MAR and a stated lower bound on the observation probability.

Author(s)

Satoshi Nakamura <s-nakamura@med.mie-u.ac.jp>

References

Nakamura, S. (2026). Sharp estimand-specific precision envelopes for blockwise missing data under missing at random. *Stats*, **9**(5), 110. doi:10.3390/stats9050110

`envelope_bootstrap` *Simultaneous Refitted-Bootstrap Band and Certified Phases*

Description

Resamples units, refits every nuisance function and recomputes the guaranteed and possible margins at each grid rate, and forms a simultaneous band from the maximum studentized deviation. A rate is certified as guaranteed, allocation-sensitive or inadmissible only when the band excludes the other phases; otherwise it is reported as indeterminate. The band is also inverted into confidence intervals for the threshold missingness rates.

Usage

```
envelope_bootstrap(fit, R = 999L, alpha = 0.05,
                   failure = c("abort", "discard"),
                   min_success_fraction = 0.75, min_success_count = 40L)

## S3 method for class 'envelope_band'
print(x, digits = 3, ...)

## S3 method for class 'envelope_band'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`fit`	An object returned by `envelope_fit` with a requirement.
`R`	Number of bootstrap refits.
`alpha`	One minus the simultaneous confidence level.

`failure` "abort" (default) stops unless every refit succeeds. "discard" drops failed refits, provided that at least `min_success_count` refits and a fraction `min_success_fraction` of them succeed; the band is then conditional on success.

`min_success_fraction`, `min_success_count`
Thresholds used with `failure` = "discard".

`x` An object returned by `envelope_bootstrap`.

`row.names`, `optional`
Not used; present for compatibility with the generic.

`digits` Number of decimal places to print.

`...` Ignored.

Details

With S successful refits, the critical value is the order statistic of rank $\lceil (S+1)(1-\alpha) \rceil$ of the maximum absolute studentized deviation, a conservative finite-sample choice; at least $\lceil (1-\alpha)/\alpha \rceil$ successful refits are needed. The guaranteed and possible margins are nonincreasing in the missingness rate, and the band is tightened under this constraint before it is inverted into intervals for q_G^* and q_P^* that respect $q_G^* \le q_P^*$.

Each refit repeats the full cross-fitted estimation, so the computation time is about `R` times that of `envelope_fit`. Call `set.seed` beforehand for reproducible output.

Value

An object of class "envelope_band", a list with components

`band` A data frame with one row per grid rate: `p`, `q`, the estimated margins with their simultaneous lower and upper limits, and the certified `phase`.

`critical_value`, `rank`, `alpha`
The critical value, its rank and the level.

`attempted`, `successful`, `failure`
Counts of refits and the failure policy.

`feasibility` For an absolute requirement, whether it is certified to hold with complete data; otherwise "not applicable".

`thresholds` A matrix with rows `guaranteed` and `possible` and columns `lower` and `upper`, or `NULL` when phases cannot be certified.

`fit` The input fit.

References

Nakamura, S. (2026). Sharp estimand-specific precision envelopes for blockwise missing data under missing at random. *Stats*, **9**(5), 110. doi:10.3390/stats9050110

See Also

`envelope_fit`

Examples

```
set.seed(2)
n <- 400
w1 <- rnorm(n)
y <- 10 + 2 * w1 + rnorm(n, sd = exp(0.3 * w1))
seen <- rbinom(n, 1, plogis(1 + 0.8 * w1))
dat <- data.frame(y = ifelse(seen == 1, y, NA), w1 = w1)
fit <- envelope_fit(dat, outcome = "y", covariates = "w1",
                    p = c(0.6, 0.8, 1.0), lower = 0.1, target = 0.8)

# A small number of refits keeps the example fast; use R = 999 in practice.
band <- envelope_bootstrap(fit, R = 39)
band

band_full <- envelope_bootstrap(fit, R = 999)
as.data.frame(band_full)
```

`envelope_fit` *Estimate the Precision Envelope from Data*

Description

Estimates, for the mean of a partially observed continuous outcome, the variance components, the realized information retention and the sharp precision envelope over a prespecified grid of observation rates. The nuisance functions are fitted by cross-fitting and the envelope endpoints by one-step estimators.

Usage

```
envelope_fit(data, outcome, covariates, observed = NULL, p, lower,
             target = NULL, variance_limit = NULL, folds = 2L,
             learners = envelope_learners(), variance_floor = NULL,
             propensity_clip = c(0.03, 0.995), min_observed_training = 20L)

## S3 method for class 'envelope_fit'
print(x, digits = 3, ...)

## S3 method for class 'envelope_fit'
plot(x, y, ...)

## S3 method for class 'envelope_fit'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`data`	A data frame with one row per unit.

`outcome` Name of the numeric outcome column. Missing values mark missing outcomes unless `observed` is given.

`covariates` Names of the completely observed numeric covariate columns W.

`observed` Optional name of a 0/1 (or logical) column marking observed outcomes.

`p` Strictly increasing grid of overall observation rates at which the envelope is estimated, each between `lower` and 1.

`lower` Lower bound π_L on the observation probability, strictly between 0 and 1. It is an assumption stated by the analyst and cannot be checked from the data.

`target`, `variance_limit` Optional precision requirement, as in `precision_envelope`. A requirement is needed for `envelope_bootstrap`.

`folds` Number of cross-fitting folds, at least 2.

`learners` A list of nuisance learners, see `envelope_learners`. Elements left out take the defaults.

`variance_floor` Positive lower bound applied to the fitted residual variance. The default is one thousandth of the variance of the observed outcomes.

`propensity_clip` Lower and upper bounds applied to the fitted observation probabilities.

`min_observed_training` Minimum number of observed outcomes required in each training fold.

`x` An object returned by `envelope_fit`.

`y`, `row.names`, `optional` Not used; present for compatibility with the generic.

`digits` Number of decimal places to print.

`...` Further arguments passed to `plot`, or ignored.

Details

The target is the outcome mean, whose complete-data influence function is $Y - \mu$. Then $a(W) = m(W) - \mu$ and $b(W) = v(W)$, with $m(W) = E(Y \mid W)$ and $v(W) = \mathrm{Var}(Y \mid W)$. Within each cross-fitting fold the outcome mean is fitted on the observed outcomes of the training folds, the residual variance is fitted to squared residuals taken from an inner two-way split of the same observed outcomes, and the observation probability $e(W)$ is fitted on all training rows. The one-step estimators of μ, A, B and the realized $C = E\{v(W)/e(W)\}$ use the augmented inverse-probability weighted influence functions; the envelope endpoints at each grid rate use the fitted favorable allocation and the fitted upper quantile of $v(W)$ for the adverse endpoint.

The reported Wald interval for the realized retention is marginal. For simultaneous statements over the grid, and for phases certified at a stated level, use `envelope_bootstrap`. The fitted share of observation probabilities below `lower` is reported as a diagnostic; it does not validate the lower bound or MAR.

Results depend on the random split into folds; call `set.seed` beforehand for reproducible output.

Value

An object of class "envelope_fit", a list including

`mu` The estimated outcome mean.

`A`, `B`, `C_realized`, `V_full`, `V_observed`
Estimated variance components, complete-data efficient variance and observed-data efficient variance under the realized mechanism.

`rho_realized`, `rho_se`, `rho_ci`
Realized information retention, its standard error and a marginal 95% Wald interval.

`table` A data frame with one row per grid rate: `p`, `q`, `C_favorable`, `C_adverse`, `rho_favorable`, `rho_adverse` and, with a requirement, `margin_guaranteed`, `margin_possible` and the point classification `phase`.

`requirement`, `theta`
The requirement and the vector of margins used by `envelope_bootstrap`.

`n`, `observed_count`, `observation_rate`, `share_below_lower`
Sample description and the overlap diagnostic.

`nuisance` The cross-fitted predictions `m`, `v` and `e`.

The object also keeps the specification, the learners and the data needed for refitting.

References

Nakamura, S. (2026). Sharp estimand-specific precision envelopes for blockwise missing data under missing at random. *Stats*, **9**(5), 110. doi:10.3390/stats9050110

See Also

`envelope_bootstrap`, `envelope_learners`, `precision_envelope`

Examples

```
set.seed(1)
n <- 500
w1 <- rnorm(n)
w2 <- rbinom(n, 1, 0.4)
y <- 10 + 2 * w1 + 1.5 * w2 + rnorm(n, sd = exp(0.3 * w1))
seen <- rbinom(n, 1, plogis(1 + 0.8 * w1))
dat <- data.frame(y = ifelse(seen == 1, y, NA), w1 = w1, w2 = w2)

fit <- envelope_fit(dat, outcome = "y", covariates = c("w1", "w2"),
                    p = c(0.6, 0.7, 0.8, 0.9), lower = 0.1, target = 0.8)
fit
plot(fit)
```

`envelope_learners` *Nuisance Learners for Envelope Estimation*

Description

`envelope_learners` collects the three learners used by `envelope_fit`: one for the outcome mean, one for the residual variance and one for the observation probability. `learner_linear` and `learner_logistic` construct the default least-squares and logistic regression learners.

Usage

```
envelope_learners(mean = learner_linear(), variance = learner_linear(),
                  propensity = learner_logistic())

learner_linear()

learner_logistic()
```

Arguments

`mean`	Learner for the outcome mean $E(Y \mid W)$, fitted on observed outcomes.
`variance`	Learner for the residual variance, fitted to squared residuals.
`propensity`	Learner for the observation probability, fitted to the 0/1 observation indicator. It must return probabilities.

Details

A learner is a function `function(x, y, newx)` that fits a model to the numeric covariate matrix `x` and response `y` and returns a numeric vector of predictions for the rows of the matrix `newx`. Any regression method can be wrapped in this form, for example a spline or a random forest from another package. The defaults standardize the covariates and fit, with an intercept, ordinary least squares (`learner_linear`) or logistic regression (`learner_logistic`). Fitted residual variances are bounded below and fitted probabilities are clipped by `envelope_fit`.

Value

`envelope_learners` returns a named list of three functions; `learner_linear` and `learner_logistic` return a learner.

See Also

`envelope_fit`

Examples

```
# A learner that adds squared terms of every covariate.
quadratic <- function(x, y, newx) {
  learner_linear()(cbind(x, x^2), y, cbind(newx, newx^2))
}
learners <- envelope_learners(mean = quadratic, variance = quadratic)
names(learners)
```

`precision_envelope` *Sharp Precision Envelope for a Planned Study*

Description

Computes, for each overall observation rate p, the smallest and largest efficient variance that a missing-at-random mechanism with observation probability bounded below by `lower` can produce, together with the value under missing completely at random (MCAR). With a precision requirement it classifies each rate and returns the threshold missingness rates. No data are needed: the inputs describe the target and the covariates at the planning stage.

Usage

```
precision_envelope(p, lower, A = NULL, r2 = NULL, b = 1, weights = NULL,
                   covariate = c("continuous", "strata"),
                   target = NULL, variance_limit = NULL)

## S3 method for class 'precision_envelope'
print(x, digits = 3, ...)

## S3 method for class 'precision_envelope'
plot(x, y, n = 201L, ...)

## S3 method for class 'precision_envelope'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`p`	Numeric vector of overall observation rates, each between `lower` and 1. The missingness rate is $q = 1 - p$.
`lower`	Lower bound π_L on the observation probability $\pi(W)$, strictly between 0 and 1. Missingness rates up to $1 - \pi_L$ are admissible.
`A`	The between-covariate variance component $A_\theta = \mathrm{Var}\{a_\theta(W)\}$, on the same scale as `b`. Supply exactly one of `A` and `r2`.
`r2`	The share $R^2 = A_\theta/(A_\theta + B_\theta)$ of the complete-data influence-function variance explained by W, in $[0, 1)$. For the mean of an outcome Y this is the usual R^2 of Y on W.

b	Values of the residual variance $b_\theta(W)$, describing its law. A single value (the default) means that $b_\theta(W)$ is constant. A vector is read as a sample of units or a discretized law, with masses `weights`. Only the shape matters for retention, so `b` can be on any scale when `r2` is used.
`weights`	Optional non-negative masses for the values in `b`; equal masses by default.
`covariate`	`"continuous"` (default) treats the covariate space as having no atoms, so that a mechanism can separate units with the same value of $b_\theta(W)$; the adverse endpoint then has the closed upper-tail form. `"strata"` treats each entry of `b` as an indivisible stratum of a discrete W, within which every unit shares one observation probability; the adverse endpoint is then found by exact enumeration, for at most 15 strata.
`target`	Optional retention requirement ρ_0 in $(0, 1]$: the requirement holds when $\rho_\theta \geq \rho_0$.
`variance_limit`	Optional absolute requirement: the observed-data efficient variance must not exceed this value. Supply at most one of `target` and `variance_limit`.
`x`	An object returned by `precision_envelope`.
`y`	Not used; present for compatibility with the generic.
`row.names, optional`	Not used; present for compatibility with the generic.
`digits`	Number of decimal places to print.
`n`	Number of missingness rates at which the curves are drawn.
`...`	Further arguments passed to `plot`, or ignored.

Details

Write $B_\theta = E\{b_\theta(W)\}$. Over observation laws with $\pi_L \leq \pi(W) \leq 1$ and $E\{\pi(W)\} = p$, the favorable endpoint $C^-(p)$ minimizes $E\{b_\theta(W)/\pi(W)\}$ and is attained by the clipped square-root rule $\pi(W) = \min[1, \max\{\pi_L, \{b_\theta(W)/\lambda\}^{1/2}\}]$. On a covariate space without atoms the adverse endpoint is

$$C^+(p) = B_\theta + (1/\pi_L - 1) \int_{1-a}^{1} Q_b(u)\, du, \qquad a = (1-p)/(1-\pi_L),$$

where Q_b is the quantile function of $b_\theta(W)$. The MCAR value is B_θ/p, which always lies between the endpoints.

Information retention is $\rho = (A_\theta + B_\theta)/(A_\theta + C)$, and the standard error is inflated by the factor $\rho^{-1/2}$. The MCAR-equivalent missingness rate of a value C is $1 - B_\theta/C$, the MCAR rate that would give the same retention.

A requirement translates into a residual budget C_{req}, equal to $(A_\theta + B_\theta)/\rho_0 - A_\theta$ for a retention requirement and to `variance_limit` $-A_\theta$ for an absolute one. A rate is guaranteed when $C^+(p) \leq C_{req}$, so that every admissible mechanism meets the requirement; allocation-sensitive when only $C^-(p) \leq C_{req}$, so that some mechanisms meet it and others do not; and inadmissible when $C^-(p) > C_{req}$. The thresholds q^*_G, q^*_{MCAR} and q^*_P are the missingness rates at which the adverse, MCAR and favorable margins reach zero. An absolute requirement that fails even with complete data is reported as infeasible. A requirement on the half-width h of a 95% interval from n units corresponds to `variance_limit = n * (h / qnorm(0.975))^2`.

Value

An object of class "precision_envelope", a list with components

table	A data frame with one row per rate: p, q, the residual components C_favorable, C_mcar and C_adverse, the retentions rho_favorable, rho_mcar and rho_adverse, the relative standard error increases se_inflation_*, the MCAR-equivalent missingness rates q_equivalent_favorable and q_equivalent_adverse, and, when a requirement is given, margin_guaranteed ($C_{req} - C^{+}$), margin_possible ($C_{req} - C^{-}$) and phase.
A, B, V_full	The variance components and the complete-data efficient variance $A_\theta + B_\theta$.
lower, q_max	The lower bound and the largest admissible missingness rate.
requirement	NULL, or a list with the requirement type, its value, C_required, the retention it implies and whether it can hold with complete data.
thresholds	NULL, or a named vector with the threshold missingness rates guaranteed, mcar and possible. A value equal to q_max means the margin stays non-negative over the whole admissible range; NA means the requirement fails with complete data.
covariate, b, weights	The inputs that describe the law of $b_\theta(W)$.

The plot method draws the three retention curves against the missingness rate and returns their values invisibly.

References

Nakamura, S. (2026). Sharp estimand-specific precision envelopes for blockwise missing data under missing at random. *Stats*, **9**(5), 110. doi:10.3390/stats9050110

See Also

residual_law_gamma, recovery_plan, envelope_fit

Examples

```
# Planning with a constant residual variance: 25% of outcomes expected to be
# missing, covariates explaining 30% of the outcome variance, and an
# observation probability of at least 0.10 for every covariate profile.
env <- precision_envelope(p = c(0.95, 0.85, 0.75, 0.65), lower = 0.10,
                          r2 = 0.30, target = 0.80)
env

# The analytic illustration of Nakamura (2026, Section 5.1): residual
# variance 0.1 or 1.9 with equal probability, A = 1 and lower bound 0.2.
ill <- precision_envelope(p = c(0.9, 0.75, 0.6), lower = 0.2, A = 1,
                          b = c(0.1, 1.9), target = 0.80)
ill$thresholds
plot(ill)

# A heteroskedastic law with coefficient of variation 0.8.
```

```
het <- precision_envelope(p = 0.75, lower = 0.10, r2 = 0.30,
                          b = residual_law_gamma(cv = 0.8))
as.data.frame(het)[, c("q", "rho_favorable", "rho_mcar", "rho_adverse")]
```

`recovery_plan` *Minimum-Cost Targeted Recovery of Missing Outcomes*

Description

Finds the least costly increase of the observation probabilities, from a current observation law, that brings the residual component $E\{b_\theta(W)/\pi(W)\}$ down to a required value. The solution is the cost-adjusted square-root rule $\pi(W) = \min[1, \max\{e_0(W), \{\eta\, b_\theta(W)/c(W)\}^{1/2}\}]$, where e_0 is the current observation probability and c the cost of raising it by one unit.

Usage

```
recovery_plan(b, baseline, A = NULL, target = NULL, variance_limit = NULL,
              residual_target = NULL, cost = 1, weights = NULL)

## S3 method for class 'recovery_plan'
print(x, digits = 3, ...)
```

Arguments

`b`	Residual variances $b_\theta(W)$ of the units or strata.
`baseline`	Current observation probabilities $e_0(W)$ in $(0, 1]$, one per entry of `b`.
`A`	Between-covariate variance component, needed with `target` or `variance_limit`.
`target`	Retention requirement in $(0, 1]$.
`variance_limit`	Absolute limit on the observed-data efficient variance.
`residual_target`	Required value of the residual component itself. Supply exactly one of `target`, `variance_limit` and `residual_target`.
`cost`	Positive marginal cost of raising the observation probability by one unit, a single value or one value per entry of `b`.
`weights`	Optional masses of the entries of `b`.
`x`	An object returned by `recovery_plan`.
`digits`	Number of decimal places to print.
`...`	Ignored.

Details

For a unit that is currently missing, the planned recovery probability is $\{\pi(W) - e_0(W)\}/\{1 - e_0(W)\}$. The plan treats b_θ, e_0 and the costs as known; with estimated inputs it is a plug-in planning calculation and carries no inferential guarantee for an adaptive recovery design.

Value

An object of class "recovery_plan", a list with components

- status — "planned", "already meets the requirement", "requires every informative outcome" or "infeasible" (the requirement cannot be met even after recovering every outcome).
- units — A data frame with b, baseline, cost, the planned probability final, the recovery probability among missing units recovery_among_missing and the initial priority $b/(c\,e_0^2)$.
- baseline_rate, final_rate, added_rate, added_cost — Mean observation probabilities before and after recovery, their difference and the expected added cost per unit.
- C_complete, C_baseline, C_target, C_achieved — The residual component with complete data, at baseline, as required and as achieved.
- eta — The multiplier of the square-root rule.

References

Nakamura, S. (2026). Sharp estimand-specific precision envelopes for blockwise missing data under missing at random. *Stats*, **9**(5), 110. doi:10.3390/stats9050110

See Also

precision_envelope

Examples

```
# Two equally sized strata; the high-variance stratum is observed half the
# time. Recover enough of it to reach 80% retention.
plan <- recovery_plan(b = c(0.1, 1.9), baseline = c(1, 0.5), A = 1,
                      target = 0.80)
plan
plan$units
```

---

residual_law_gamma *A Heteroskedastic Law for the Residual Variance*

Description

Returns equally weighted quantiles of a gamma distribution with a given mean and coefficient of variation, for use as the argument b of precision_envelope when the residual variance $b_\theta(W)$ is expected to vary across covariate profiles.

Usage

```
residual_law_gamma(cv, n = 1000L, mean = 1)
```

Arguments

cv	Coefficient of variation of $b_\theta(W)$, non-negative. Zero gives a constant residual variance.
n	Number of quantiles.
mean	Mean of the returned values.

Details

The values are the gamma quantiles at $(i - 0.5)/n$, $i = 1, \ldots, n$, rescaled to have exactly the requested mean. The gamma family is a convenient planning assumption, not a model the method requires; any vector of plausible residual variances, for example fitted values from a pilot study, can be passed to precision_envelope instead.

Value

A numeric vector of length n.

See Also

precision_envelope

Examples

```
b <- residual_law_gamma(cv = 0.5)
c(mean = mean(b), cv = sd(b) / mean(b))
```

Index

∗ **distribution**
    residual_law_gamma, 13
∗ **models**
    envelope_bootstrap, 3
    envelope_fit, 5
    envelope_learners, 8
    precision_envelope, 9
∗ **optimize**
    recovery_plan, 12
∗ **package**
    precisionenvelope-package, 2

as.data.frame.envelope_band
        *(*envelope_bootstrap*)*, 3
as.data.frame.envelope_fit
        *(*envelope_fit*)*, 5
as.data.frame.precision_envelope
        *(*precision_envelope*)*, 9

envelope_bootstrap, 3, *3*, *6, 7*
envelope_fit, *3, 4*, 5, *8*, *11*
envelope_learners, *6, 7*, 8

learner_linear *(*envelope_learners*)*, 8
learner_logistic *(*envelope_learners*)*, 8

plot, *6*, *10*
plot.envelope_fit *(*envelope_fit*)*, 5
plot.precision_envelope
        *(*precision_envelope*)*, 9
precision_envelope, *2*, *6, 7*, 9, *13, 14*
precisionenvelope
        *(*precisionenvelope-package*)*, 2
precisionenvelope-package, 2
print.envelope_band
        *(*envelope_bootstrap*)*, 3
print.envelope_fit *(*envelope_fit*)*, 5
print.precision_envelope
        *(*precision_envelope*)*, 9
print.recovery_plan *(*recovery_plan*)*, 12

recovery_plan, *2*, *11*, 12
residual_law_gamma, *2*, *11*, 13

set.seed, *4*, *6*