Package 'UniCensor'

July 17, 2026

Type Package

Title Reproducible Random Samples Under Univariate Censoring Schemes

Version 0.1.0

Description Generates reproducible random samples from any user-specified univariate distribution under a comprehensive suite of censoring and truncation schemes.
Users supply the probability density function (PDF), cumulative distribution function (CDF), survival function, support bounds, and parameters; the same seed and inputs yield identical samples across sessions. Supported schemes include right and left truncation, random, right, left, interval, and middle censoring, block random censoring, balanced joint progressive Type-II (BJPT-II), progressive first failure, joint Type-I, Type-I, Type-II, progressive Type-II, Type-II progressively hybrid, joint Type-II, hybrid, hybrid Type-I, doubly Type-II, Type-I hybrid, and hybrid Type-II censoring. Diagnostic histogram, dot plot, and autocorrelation plots are provided for each scheme to verify distributional behaviour.
Methods are described in
Nagar, Kumar, and Krishna (2026) <doi:10.59467/IJASS.2026.22.1>,
Goel, Kumar, and Krishna (2026, ``Estimation in power Lindley distributions using balanced joint progressively Type-II censored data"),
Wu and Kus (2009) <doi:10.1016/j.csda.2009.03.010>,
Goel and Krishna (2026) <doi:10.1007/s13198-026-03208-w>,
Balakrishnan and Aggarwala (2000, ISBN:978-1-4612-1334-5),
Mondal and Kundu (2020) <doi:10.1080/03610926.2018.1554128>,
Ding and Gui (2023) <doi:10.3390/math11092003>,
Prajapati, Mitra, and Kundu (2019) <doi:10.1007/s13571-018-0167-0>,
Yadav, Jaiswal, and Yadav (2026) <doi:10.1007/s11135-026-02647-8>,
Iyer, Jammalamadaka, and Kundu (2008) <doi:10.1016/j.jspi.2007.03.062>,
Banerjee and Kundu (2008) <doi:10.1109/TR.2008.916890>,
and Kundu and Joarder (2006) <doi:10.1016/j.csda.2005.05.002>.

License GPL-3

Depends R (>= 4.0.0)

Imports graphics, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1606-0844>)

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-07-17 13:40:08 UTC

Contents

as.data.frame.censored_sample . . . 3
as.vector.censored_sample . . . 3
dist_spec . . . 4
plot_censor_acf . . . 5
plot_censor_diagnostics . . . 6
plot_censor_dot . . . 6
plot_censor_hist . . . 7
print.censored_sample . . . 7
print.dist_spec . . . 8
r_bjpt2_censor . . . 8
r_block_random_censor . . . 9
r_complete . . . 10
r_doubly_type2_censor . . . 11
r_hybrid_censor . . . 11
r_hybrid_type1_censor . . . 12
r_hybrid_type2_censor . . . 13
r_interval_censor . . . 14
r_joint_type1_censor . . . 15
r_joint_type2_censor . . . 15
r_left_censor . . . 16
r_left_truncation . . . 17
r_middle_censor . . . 18
r_progressive_first_failure . . . 18
r_progressive_hybrid_type2_censor . . . 19
r_progressive_type2_censor . . . 20
r_random_censor . . . 21
r_right_censor . . . 22
r_right_truncation . . . 22
r_type1_censor . . . 23
r_type1_hybrid_censor . . . 24
r_type2_censor . . . 25

Index **26**

`as.data.frame.censored_sample`

Coerce censored sample to data frame

Description

Coerce censored sample to data frame

Usage

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

Arguments

`x`	a `censored_sample` object.
`row.names`	ignored.
`optional`	ignored.
`...`	ignored.

Value

A data frame with columns `x` (observed values), `status` (integer status code: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), `t_true` (latent true failure times), and `time2` (upper interval bound for interval- or middle-censored observations, `NA` otherwise).

`as.vector.censored_sample`

Extract observed values from a censored sample

Description

Extract observed values from a censored sample

Usage

```
## S3 method for class 'censored_sample'
as.vector(x, ...)
```

Arguments

`x`	a `censored_sample` object.
`...`	ignored.

Value

Numeric vector of observed values (failures and censoring times).

`dist_spec` *Define a univariate distribution for simulation*

Description

Creates a reusable distribution specification passed to all **UniCensor** sampling functions. At least one of `q`, `surv`, `p`, or `d` must be supplied for random number generation via inverse transform sampling.

Usage

```
dist_spec(
  d = NULL,
  p = NULL,
  q = NULL,
  surv = NULL,
  lower = 0,
  upper = Inf,
  param = list(),
  name = "custom"
)
```

Arguments

`d`	function; PDF $f(x, \theta)$.
`p`	function; CDF $F(x, \theta)$.
`q`	function; quantile function $F^{-1}(u, \theta)$.
`surv`	function; survival function $S(x, \theta)$.
`lower`	numeric; lower support bound (default `0`).
`upper`	numeric; upper support bound (default `Inf`).
`param`	named list of distribution parameters (default empty).
`name`	character label for plots (default `"custom"`).

Value

An object of class `"dist_spec"`, which is a list with elements `d`, `p`, `q`, `surv` (user-supplied functions or `NULL`), `lower` and `upper` (support bounds), `param` (named list of parameters), and `name` (character label).

Examples

```
# Exponential(rate = 1)
exp_dist <- dist_spec(
  d    = function(x, p) p$rate * exp(-p$rate * x),
  p    = function(x, p) 1 - exp(-p$rate * x),
  q    = function(u, p) -log(1 - u) / p$rate,
  surv = function(x, p) exp(-p$rate * x),
  lower = 0, upper = Inf,
  param = list(rate = 1),
  name  = "Exp(1)"
)
```

`plot_censor_acf` *Autocorrelation function plot*

Description

Plots the ACF of latent failure times to assess independence of the underlying RNG stream (should resemble white noise for valid simulation).

Usage

```
plot_censor_acf(samp, use_latent = TRUE, max_lag = NULL, ...)
```

Arguments

`samp`	a `censored_sample` object.
`use_latent`	logical; compute ACF on `t_true` (default) or observed `x`.
`max_lag`	maximum lag (default `min(20, n/4)`).
`...`	passed to `plot`.

Value

Invisibly, an object of class `"acf"` as returned by `acf`, containing the autocorrelation values at each lag. Called primarily for its side effect of producing an ACF plot with 95\

plot_censor_diagnostics *Combined diagnostic panel: histogram, dot plot, and ACF*

Description

Produces a three-panel plot (histogram with PDF overlay, dot plot, and ACF) for a censored sample. Graphical parameters are saved before modification and restored on exit via `on.exit`.

Usage

```
plot_censor_diagnostics(samp, dist = NULL, use_latent = TRUE, ...)
```

Arguments

`samp`	a `censored_sample` object.
`dist`	optional `dist_spec`.
`use_latent`	passed to individual plot functions.
`...`	ignored.

Value

Invisibly, the input `censored_sample` object `samp`. Called primarily for its side effect of producing a combined diagnostic plot panel.

plot_censor_dot *Dot plot (index plot) of sample values*

Description

Dot plot (index plot) of sample values

Usage

```
plot_censor_dot(samp, use_latent = TRUE, pch = NULL, ...)
```

Arguments

`samp`	a `censored_sample` object.
`use_latent`	logical; plot latent (`t_true`) or observed (`x`) values.
`pch`	plotting character; censored points use `pch = 1`, events use `pch = 16`.
`...`	passed to `plot`.

Value

Invisibly, a list with elements `index` (integer vector of observation indices) and `values` (numeric vector of plotted values). Called primarily for its side effect of producing a dot (index) plot distinguishing events from censored observations.

`plot_censor_hist` *Histogram of observed values with true density overlay*

Description

For censored samples, uses latent `t_true` values to overlay the target density when available; observed `x` values are histogrammed with a note on censoring status.

Usage

```
plot_censor_hist(samp, dist = NULL, use_latent = TRUE, nbreaks = 30L, ...)
```

Arguments

`samp`	a `censored_sample` object.
`dist`	optional `dist_spec`; defaults to `samp$dist`.
`use_latent`	logical; plot histogram of latent failures (`t_true`) instead of observed `x` (default `TRUE` for verifying the underlying distribution).
`nbreaks`	number of histogram breaks (default `30`).
`...`	graphical parameters passed to `hist`.

Value

Invisibly, an object of class `"histogram"` as returned by `hist`, containing break points, counts, and density values. Called primarily for its side effect of producing a histogram plot with an optional PDF overlay.

`print.censored_sample` *Print method for censored_sample*

Description

Print method for censored_sample

Usage

```
## S3 method for class 'censored_sample'
print(x, ...)
```

Arguments

x	a censored_sample object.
...	ignored.

Value

Invisibly, the input censored_sample object x. Called for its side effect of printing the censoring scheme, distribution name, sample size, seed, and a frequency table of status codes.

---

print.dist_spec *Print method for dist_spec*

Description

Print method for dist_spec

Usage

```
## S3 method for class 'dist_spec'
print(x, ...)
```

Arguments

x	a dist_spec object.
...	ignored.

Value

Invisibly, the input dist_spec object x. Called for its side effect of printing a human-readable summary of the distribution name, support, parameters, and available functions.

---

r_bjpt2_censor *Balanced joint progressive Type-II (BJPT-II) censoring*

Description

k samples of size n are progressively Type-II censored with a balanced removal scheme scheme applied jointly at each failure time across all groups.

Usage

```
r_bjpt2_censor(n, dist, scheme = NULL, k = 2L, seed = NULL)
```

Arguments

n	sample size per group.
dist	a dist_spec object.
scheme	integer vector of total removals at each failure (split evenly across groups).
k	number of groups (default 2).
seed	optional integer seed.

Value

A list of censored_sample objects (class "censored_sample_list"), one per group. Each element contains scheme, dist, n, seed, t_true, x, status (0 = right-censored, 1 = event), time2, and extra fields group and k.

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_bjpt2_censor(20, d, k = 2, seed = 1)
```

---

r_block_random_censor *Block random censoring*

Description

Subjects are partitioned into blocks of size block_size; entire blocks are randomly selected for right censoring at the block maximum observed failure time or at an exponential censoring time, whichever is smaller.

Usage

```
r_block_random_censor(
  n,
  dist,
  block_size = 5L,
  prop_blocks = 0.4,
  cen_rate = 0.2,
  seed = NULL
)
```

Arguments

n	sample size.
dist	a dist_spec object.
block_size	block length (default 5).
prop_blocks	fraction of blocks censored (default 0.4).
cen_rate	exponential censoring rate within censored blocks.
seed	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_block_random_censor(30, d, seed = 1)
```

`r_complete` *Generate a complete (uncensored) random sample*

Description

Generate a complete (uncensored) random sample

Usage

```
r_complete(n, dist, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`seed`	optional integer seed for reproducibility.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 1 = event for all observations in a complete sample), and `time2` (`NA` for all).

Examples

```
exp_dist <- dist_spec(
  q = function(u, p) -log(1 - u) / p$rate,
  surv = function(x, p) exp(-p$rate * x),
  param = list(rate = 1), name = "Exp(1)"
)
samp <- r_complete(50, exp_dist, seed = 42)
table(samp$status)
```

r_doubly_type2_censor *Doubly Type-II censoring*

Description

Classical doubly Type-II censoring: the `r1` smallest and `r2` largest latent failure times are censored; the middle order statistics are observed exactly. Requires `r1 + r2 < n`.

Usage

```
r_doubly_type2_censor(n, dist, r1, r2, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`r1`	number of smallest order statistics left-censored.
`r2`	number of largest order statistics right-censored.
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_doubly_type2_censor(50, d, r1 = 5, r2 = 5, seed = 1)
```

r_hybrid_censor *Hybrid censoring (Type-I and Type-II constraints)*

Description

Stops at `cen_time` or when `r_failures` events occur, whichever comes first.

Usage

```
r_hybrid_censor(n, dist, cen_time, r_failures, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
cen_time	fixed time limit.
r_failures	failure count limit.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_hybrid_censor(30, d, cen_time = 2, r_failures = 15, seed = 1)
```

---

r_hybrid_type1_censor *Hybrid Type-I censoring with progressive withdrawal*

Description

At each time in prog_times, scheme[j] units are withdrawn; remaining units are Type-I censored at cen_time.

Usage

```
r_hybrid_type1_censor(n, dist, cen_time, prog_times, scheme, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
cen_time	final Type-I censoring time.
prog_times	vector of progressive withdrawal times.
scheme	removals at each progressive time.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_hybrid_type1_censor(30, d, cen_time = 3,
                      prog_times = c(1, 2), scheme = c(2, 2), seed = 1)
```

---

r_hybrid_type2_censor *Hybrid Type-II censoring*

Description

Progressive Type-II removals are applied, then the experiment terminates at the r_failures-th remaining failure.

Usage

```
r_hybrid_type2_censor(n, dist, scheme, r_failures, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
scheme	progressive removal scheme before Type-II stop.
r_failures	failure count for final Type-II termination.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_hybrid_type2_censor(30, d, scheme = c(1, 1), r_failures = 10, seed = 1)
```

`r_interval_censor` *Interval censoring*

Description

A proportion `prop_interval` of subjects are interval-censored into windows of width `width`; optional random right censoring applies to the remainder at rate `cen_rate`.

Usage

```
r_interval_censor(
  n,
  dist,
  width,
  prop_interval = 0.3,
  cen_rate = 0.2,
  seed = NULL
)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`width`	interval window width.
`prop_interval`	fraction interval-censored (default `0.3`).
`cen_rate`	exponential rate for additional right censoring (default `0.2`).
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_interval_censor(30, d, width = 0.5, seed = 1)
```

r_joint_type1_censor *Joint Type-I censoring across multiple samples*

Description

Generates k independent samples of size n each, all censored at a common fixed time cen_time.

Usage

```
r_joint_type1_censor(n, dist, cen_time, k = 2L, seed = NULL)
```

Arguments

n	sample size per group.
dist	a dist_spec object.
cen_time	fixed censoring time.
k	number of independent samples (default 2).
seed	optional integer seed.

Value

A list of censored_sample objects (class "censored_sample_list"), one per group. Each element contains scheme, dist, n, seed, t_true, x, status (0 = right-censored, 1 = event), time2, and extra fields group and k.

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_joint_type1_censor(20, d, cen_time = 2, seed = 1)
```

r_joint_type2_censor *Joint Type-II censoring across multiple samples*

Description

All k samples share the same termination rule: stop when a total of r_failures events have been observed across all groups combined.

Usage

```
r_joint_type2_censor(n, dist, r_failures, k = 2L, seed = NULL)
```

Arguments

n	sample size per group.
dist	a dist_spec object.
r_failures	total failure count across all groups.
k	number of groups (default 2).
seed	optional integer seed.

Value

A list of censored_sample objects (class "censored_sample_list"), one per group. Each element contains scheme, dist, n, seed, t_true, x, status (0 = right-censored, 1 = event), time2, and extra fields group and k.

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_joint_type2_censor(20, d, r_failures = 15, seed = 1)
```

---

r_left_censor *Left censoring at a fixed threshold*

Description

Left censoring at a fixed threshold

Usage

```
r_left_censor(n, dist, threshold, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
threshold	left-censoring threshold.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_left_censor(30, d, threshold = 0.5, seed = 1)
```

`r_left_truncation` *Left-truncated random sample*

Description

Observations are drawn from $X \mid X \geq$ `left`.

Usage

```
r_left_truncation(n, dist, left, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`left`	lower truncation point.
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(p = function(x, p) 1 - exp(-p$rate * x),
               q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_left_truncation(20, d, left = 0.5, seed = 1)
```

r_middle_censor *Middle censoring*

Description

Observations that fall within the interval [lower, upper] are middle-censored; observations outside this interval are observed exactly.

Usage

```
r_middle_censor(n, dist, lower, upper, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
lower	lower bound of the middle censoring interval.
upper	upper bound of the middle censoring interval.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_middle_censor(30, d, lower = 0.5, upper = 1.5, seed = 1)
```

r_progressive_first_failure *Progressive first-failure censoring*

Description

At each failure one surviving unit is randomly removed until only one unit remains under observation.

Usage

```
r_progressive_first_failure(n, dist, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_progressive_first_failure(15, d, seed = 1)
```

---

r_progressive_hybrid_type2_censor

Type-II progressively hybrid censoring

Description

Combines progressive Type-II censoring with a pre-set time limit cen_time. Progressive removals according to scheme are applied at each failure. The experiment terminates at $\min(X_{m:m:n}, T)$, where $X_{m:m:n}$ is the m-th (i.e., r_failures-th) failure among the remaining units and T is cen_time. If fewer than r_failures events occur before cen_time, all surviving units are censored at cen_time. See Kundu and Joarder (2006) doi:10.1016/j.csda.2005.05.002.

Usage

```
r_progressive_hybrid_type2_censor(
  n,
  dist,
  scheme,
  r_failures = NULL,
  cen_time,
  seed = NULL
)
```

Arguments

n	sample size.
dist	a dist_spec object.
scheme	integer vector of removals at successive failure times.

r_failures	target number of failures (default $\lfloor 0.7n \rfloor$).
cen_time	pre-set maximum experiment time.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character "progressive_hybrid_type2"), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), time2 (upper interval bound where applicable, NA otherwise), and extra (list containing scheme, r_failures, cen_time, and stop_time).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_progressive_hybrid_type2_censor(30, d, scheme = c(1, 1, 1),
                                  r_failures = 10, cen_time = 3, seed = 1)
```

---

r_progressive_type2_censor

Progressive Type-II censoring

Description

At the j-th failure time, scheme[j] surviving units are randomly removed from the risk set.

Usage

```
r_progressive_type2_censor(n, dist, scheme = NULL, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
scheme	integer vector of removals at successive failure times.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_progressive_type2_censor(30, d, seed = 1)
```

`r_random_censor` *Random censoring*

Description

Each subject is observed until $\min(X, C)$ where C follows an exponential distribution with rate `cen_rate`.

Usage

```
r_random_censor(n, dist, cen_rate = 0.2, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`cen_rate`	exponential censoring rate (default `0.2`).
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_random_censor(30, d, seed = 1)
```

r_right_censor *Right censoring at a fixed time*

Description

Right censoring at a fixed time

Usage

```
r_right_censor(n, dist, cen_time, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
cen_time	fixed right-censoring time.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_right_censor(30, d, cen_time = 2, seed = 1)
```

r_right_truncation *Right-truncated random sample*

Description

Observations are drawn from $X \mid X \leq$ right.

Usage

```
r_right_truncation(n, dist, right, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
right	upper truncation point.
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(p = function(x, p) 1 - exp(-p$rate * x),
               q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_right_truncation(20, d, right = 3, seed = 1)
```

---

r_type1_censor *Type-I (time-terminated) censoring*

Description

All units are observed until fixed time cen_time; failures after that time are right-censored.

Usage

```
r_type1_censor(n, dist, cen_time = NULL, seed = NULL)
```

Arguments

n	sample size.
dist	a dist_spec object.
cen_time	fixed censoring time (default: 70th percentile of latent failure times from a pilot draw).
seed	optional integer seed.

Value

A censored_sample object (S3 class) containing: scheme (character), dist (the dist_spec), n (integer sample size), seed, t_true (numeric vector of latent failure times), x (observed/censored values), status (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and time2 (upper interval bound where applicable, NA otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_type1_censor(30, d, cen_time = 2, seed = 1)
```

`r_type1_hybrid_censor` *Type-I hybrid censoring*

Description

Experiment terminates at `cen_time` or when `r_failures` events occur; if `r_failures` is reached first, a final Type-I censoring time `cen_time` still applies to remaining units.

Usage

```
r_type1_hybrid_censor(n, dist, cen_time, r_failures, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`cen_time`	fixed censoring time.
`r_failures`	failure count target.
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_type1_hybrid_censor(30, d, cen_time = 2, r_failures = 15, seed = 1)
```

`r_type2_censor` *Type-II (failure-terminated) censoring*

Description

Experiment stops at the `r_failures`-th observed failure; remaining units are right-censored at that time.

Usage

```
r_type2_censor(n, dist, r_failures = NULL, seed = NULL)
```

Arguments

`n`	sample size.
`dist`	a `dist_spec` object.
`r_failures`	number of failures before termination (default $\lfloor 0.7n \rfloor$).
`seed`	optional integer seed.

Value

A `censored_sample` object (S3 class) containing: `scheme` (character), `dist` (the `dist_spec`), `n` (integer sample size), `seed`, `t_true` (numeric vector of latent failure times), `x` (observed/censored values), `status` (integer: 0 = right-censored, 1 = event, 2 = left-censored, 3 = interval-censored, 4 = middle-censored), and `time2` (upper interval bound where applicable, `NA` otherwise).

Examples

```
d <- dist_spec(q = function(u, p) -log(1 - u) / p$rate,
               param = list(rate = 1), name = "Exp(1)")
r_type2_censor(30, d, r_failures = 20, seed = 1)
```

Index

acf, *5*
as.data.frame.censored_sample, 3
as.vector.censored_sample, 3

dist_spec, 4, *9–25*

hist, *7*

on.exit, *6*

plot, *5*, *6*
plot_censor_acf, 5
plot_censor_diagnostics, 6
plot_censor_dot, 6
plot_censor_hist, 7
print.censored_sample, 7
print.dist_spec, 8

r_bjpt2_censor, 8
r_block_random_censor, 9
r_complete, 10
r_doubly_type2_censor, 11
r_hybrid_censor, 11
r_hybrid_type1_censor, 12
r_hybrid_type2_censor, 13
r_interval_censor, 14
r_joint_type1_censor, 15
r_joint_type2_censor, 15
r_left_censor, 16
r_left_truncation, 17
r_middle_censor, 18
r_progressive_first_failure, 18
r_progressive_hybrid_type2_censor, 19
r_progressive_type2_censor, 20
r_random_censor, 21
r_right_censor, 22
r_right_truncation, 22
r_type1_censor, 23
r_type1_hybrid_censor, 24
r_type2_censor, 25