

Package: gofPHCS (via r-universe)

October 4, 2026

Type Package

Title Goodness-of-Fit Tests for Complete, Progressively Type-II,
Type-I Hybrid, and Type-II Hybrid Censored Data

Version 0.1.0

Description Provides goodness-of-fit tests for lifetime data collected under complete sampling, progressive Type-II censoring, and Type-I/Type-II hybrid censoring schemes. Users supply the observed (censored) data and the assumed probability density/mass function, cumulative distribution function, or survival function of the target model, and the package returns the corresponding test statistic together with an asymptotic or Monte Carlo p-value. Implements the spacings-based exponentiality test of Balakrishnan, Ng and Kannan (2002, in ``Goodness-of-Fit Tests and Model Validity", Birkhauser, pp. 89-111) and its location-scale generalization Balakrishnan, Ng and Kannan (2004) <[doi:10.1109/TR.2004.833317](https://doi.org/10.1109/TR.2004.833317)>, the power comparison and Kaplan-Meier based tests of Doering and Cramer (2019) <[doi:10.1080/00949655.2019.1648468](https://doi.org/10.1080/00949655.2019.1648468)>, the Kolmogorov-Smirnov type tests for hybrid censored data of Banerjee and Pradhan (2018) <[doi:10.1080/03610926.2016.1205616](https://doi.org/10.1080/03610926.2016.1205616)>, and follows the unified treatment of hybrid censoring schemes reviewed in Balakrishnan and Kundu (2013) <[doi:10.1016/j.csda.2012.03.025](https://doi.org/10.1016/j.csda.2012.03.025)> and in Cramer and Balakrishnan (2023, ``Hybrid Censoring Know-How", Chapter 11) <[doi:10.1016/B978-0-12-398387-9.00019-2](https://doi.org/10.1016/B978-0-12-398387-9.00019-2)>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports stats, withr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spelling, covr

VignetteBuilder knitr

Config/testthat/edition 3
Language en-US
NeedsCompilation no
Author Shikhar Tyagi [aut, cre] (ORCID:
 <<https://orcid.org/0000-0003-1606-0844>>), Arvind Pandey [aut],
 Bhupendra Singh [aut], Vrijesh Tripathi [aut]
Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>
Repository <https://shikhartyagi.r-universe.dev>
Date/Publication 2026-08-04 14:57:13 UTC
RemoteUrl <https://github.com/cran/gofPHCS>
RemoteRef HEAD
RemoteSha 02a2eacd077fbac8207bf871de04ef383bb25577

Contents

aluminum_coupons	2
cens_data	3
gof_test	4
insulating_fluid	6
make_distribution	6
rcens	8
Index	10

aluminum_coupons	<i>Aluminum Coupon Fatigue Lifetimes Data</i>
------------------	---

Description

Fatigue lifetimes (in 10^3 cycles) of 101 aluminum coupons under periodic stress loading.

Usage

aluminum_coupons

Format

A numeric vector of length 101.

Source

Birnbaum, Z. W., & Saunders, S. C. (1958). A statistical model for life-length of materials. *JASA*, 53(281), 151-160.

References

Banerjee, B., & Pradhan, B. (2018). Kolmogorov-Smirnov test for life test data with hybrid censoring. *Communications in Statistics - Theory and Methods*, 47(11), 2590-2604.

cens_data

Create a Censored Data Container

Description

Constructor for creating a structured censored dataset object of class "cens_data".

Usage

```
cens_data(
  x,
  scheme = c("complete", "typeII", "proptypeII", "hybridI", "hybridII", "hybridI_hcs",
    "hybridII_hcs"),
  n = length(x),
  r = NULL,
  R = NULL,
  T0 = NULL,
  verbose = FALSE
)
```

Arguments

x	Numeric vector of observed failure times. Must be finite.
scheme	Character string specifying the censoring scheme. Options are: "complete", "typeII", "proptypeII", "hybridI", "hybridII", "hybridI_hcs", and "hybridII_hcs".
n	Total sample size at the beginning of the experiment. Defaults to length(x).
r	Integer specifying the pre-fixed failure count for Type-II or hybrid censoring schemes.
R	Integer vector of progressive removals for "proptypeII", "hybridI_hcs", or "hybridII_hcs".
T0	Positive numeric value for pre-fixed time threshold in hybrid censoring schemes.
verbose	Logical; if TRUE, emits informative messages (e.g. when sorting unsorted x). Default is FALSE.

Value

An S3 object of class "cens_data" containing:

x	Numeric vector of sorted observed failure times.
scheme	Censoring scheme name.
n	Total initial sample size.

n_obs	Number of observed failure times <code>length(x)</code> .
r	Pre-fixed failure count (if applicable).
R	Progressive removal vector (if applicable).
T0	Time threshold (if applicable).

References

Cramer, E., & Balakrishnan, N. (2023). Goodness-of-fit tests. In *Hybrid Censoring Know-How* (Chapter 11, pp. 321-329). Academic Press. doi:[10.1016/B9780123983879.000192](https://doi.org/10.1016/B9780123983879.000192)

Examples

```
# Complete data
cd1 <- cens_data(x = c(0.5, 1.2, 2.1, 3.4), scheme = "complete")
print(cd1)

# Progressive Type-II censored data
cd2 <- cens_data(x = c(0.19, 0.78, 0.96, 1.31), scheme = "progtypesII", R = c(0, 1, 0, 2))
summary(cd2)
```

gof_test

Goodness-of-Fit Test Dispatcher for Censored Lifetime Data

Description

Main function for conducting goodness-of-fit tests on complete, progressively Type-II, and hybrid censored lifetime data.

Usage

```
gof_test(
  data,
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  pmf = NULL,
  distribution = NULL,
  params = NULL,
  statistic = "auto",
  p.method = c("auto", "asymptotic", "montecarlo"),
  nsim = 9999,
  seed = NULL,
  conf.level = 0.95,
  ...
)
```

Arguments

<code>data</code>	A "cens_data" object created by cens_data .
<code>pdf</code>	Optional probability density function $f(x, \dots)$.
<code>cdf</code>	Optional cumulative distribution function $F(x, \dots)$.
<code>sf</code>	Optional survival function $S(x, \dots)$.
<code>pmf</code>	Optional probability mass function $P(x, \dots)$.
<code>distribution</code>	Optional pre-constructed "gof_distribution" object created by make_distribution .
<code>params</code>	Parameter vector or named list for the null distribution.
<code>statistic</code>	Character string specifying test statistic name. Default "auto" selects default test for data\$scheme. Use "all" to compute all applicable test statistics.
<code>p.method</code>	Character string specifying p-value calculation method ("auto", "asymptotic", or "montecarlo").
<code>nsim</code>	Integer specifying number of Monte Carlo replicates. Default is 9999.
<code>seed</code>	Optional integer seed for reproducible Monte Carlo simulation.
<code>conf.level</code>	Confidence level (default 0.95).
<code>...</code>	Additional arguments passed to specific statistic functions (e.g. <code>mc_moments</code> for "genT").

Value

An object of class `c("gof_htest", "htest")` containing:

<code>statistic</code>	Observed test statistic value.
<code>p.value</code>	Calculated p-value.
<code>method</code>	Character string describing the test.
<code>data.name</code>	Character string describing the dataset.
<code>estimate</code>	Parameter estimates / fitted values.

References

- Balakrishnan, N., Ng, H. K. T., & Kannan, N. (2002). A test of exponentiality based on spacings for progressively type-II censored data. In *Goodness-of-Fit Tests and Model Validity* (pp. 89-111). Birkhauser.
- Banerjee, B., & Pradhan, B. (2018). Kolmogorov-Smirnov test for life test data with hybrid censoring. *Communications in Statistics - Theory and Methods*, 47(11), 2590-2604. doi:10.1080/03610926.2016.1205616
- Cramer, E., & Balakrishnan, N. (2023). Goodness-of-fit tests. In *Hybrid Censoring Know-How* (Chapter 11). Academic Press. doi:10.1016/B9780123983879.000192

Examples

```
set.seed(42)
x_data <- rexp(15, rate = 0.5)
cd <- cens_data(x = x_data, scheme = "complete")
dist <- make_distribution(cdf = function(x, rate) pexp(x, rate), params = c(rate = 0.5))
res <- gof_test(cd, distribution = dist, statistic = "KS", p.method = "montecarlo", nsim = 99)
print(res)
```

insulating_fluid

Insulating Fluid Breakdown Times Data

Description

Times to breakdown (in minutes) of an insulating fluid subjected to high voltage stress.

Usage

```
insulating_fluid
```

Format

A numeric vector of length 18.

Source

Nelson, W. (1982). *Applied Life Data Analysis*. John Wiley & Sons.

References

Cramer, E., & Balakrishnan, N. (2023). Goodness-of-fit tests. In *Hybrid Censoring Know-How* (Chapter 11). Academic Press.

make_distribution

Create a Unified Distribution Adapter

Description

Constructs a distribution object encapsulating probability density function (pdf), cumulative distribution function (cdf), survival function (sf), and quantile function (qdist). Missing functions are derived automatically using numerical calculus and root-finding.

Usage

```
make_distribution(
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  pmf = NULL,
  discrete = FALSE,
  support = c(-Inf, Inf),
  params = NULL
)
```

Arguments

pdf	Probability density function $f(x, \dots)$.
cdf	Cumulative distribution function $F(x, \dots)$.
sf	Survival function $S(x, \dots)$.
pmf	Probability mass function $P(x, \dots)$ for discrete models.
discrete	Logical indicating if the distribution is discrete. Default is FALSE.
support	Numeric vector of length 2 defining distribution support bounds $c(\text{lower}, \text{upper})$. Default is $c(-\text{Inf}, \text{Inf})$.
params	Named list or numeric vector of parameters passed to pdf, cdf, sf, or pmf.

Value

An S3 object of class "gof_distribution" containing:

pdf	Vectorized density function $\text{pdf}(x)$.
cdf	Vectorized CDF function $\text{cdf}(x)$.
sf	Vectorized survival function $\text{sf}(x)$.
qdist	Vectorized quantile function $\text{qdist}(p)$.
discrete	Logical flag.
support	Support interval.
params	User-supplied parameter vector or list.

Examples

```
# Create an Exponential distribution adapter
exp_dist <- make_distribution(
  cdf = function(x, rate) pexp(x, rate = rate),
  params = c(rate = 0.5),
  support = c(0, Inf)
)
exp_dist$cdf(1)
exp_dist$pdf(1)
```

rcens

*Random Generation Engine for Censored Data***Description**

Generates simulated datasets under specified censoring schemes from a quantile function `qdist`.

Usage

```
rcens(
  scheme = c("complete", "progtypesII", "hybridI", "hybridII", "hybridI_hcs",
    "hybridII_hcs"),
  n,
  R = NULL,
  r = NULL,
  T0 = NULL,
  qdist,
  nsim = 1
)
```

Arguments

<code>scheme</code>	Character string specifying censoring scheme ("complete", "progtypesII", "hybridI", "hybridII", "hybridI_hcs", "hybridII_hcs").
<code>n</code>	Integer specifying initial sample size.
<code>R</code>	Integer vector of progressive removals.
<code>r</code>	Integer specifying target failure count for Type-II or hybrid schemes.
<code>T0</code>	Numeric value specifying time threshold for hybrid schemes.
<code>qdist</code>	Quantile function <code>qdist(p)</code> of the target distribution.
<code>nsim</code>	Integer specifying the number of datasets to simulate. Default is 1.

Value

If `nsim = 1`, returns a single "cens_data" object. If `nsim > 1`, returns a list of "cens_data" objects.

References

Balakrishnan, N., & Sandhu, R. A. (1995). A simple simulational algorithm for generating progressive Type-II censored samples. *The American Statistician*, 49(2), 229-230.

Banerjee, B., & Pradhan, B. (2018). Kolmogorov-Smirnov test for life test data with hybrid censoring. *Communications in Statistics - Theory and Methods*, 47(11), 2590-2604. doi:10.1080/03610926.2016.1205616

Examples

```
# Simulate a progressive Type-II censored sample from Standard Exponential
set.seed(123)
sim_data <- rcens(
  scheme = "proptypeII", n = 10, R = c(0, 1, 0, 1, 3),
  qdist = function(p) qexp(p, rate = 1)
)
print(sim_data)
```

Index

* **datasets**

aluminum_coupons, [2](#)

insulating_fluid, [6](#)

aluminum_coupons, [2](#)

cens_data, [3](#), [5](#)

gof_test, [4](#)

insulating_fluid, [6](#)

make_distribution, [5](#), [6](#)

rcens, [8](#)