Package 'Gofpt2'

August 2, 2026

Type Package

Title Generalized Goodness-of-Fit Test for Progressive Type-II Censored Data

Version 0.1.0

Date 2026-07-22

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>

Description Implements a generalized goodness-of-fit test based on spacings for general progressive Type-II censored data. The test statistic is based on the work of Qin et al. (2022) <doi:10.1080/02664763.2020.1821613> and extends the methodology of Balakrishnan et al. (2003) <doi:10.1007/978-1-4612-0103-8_8>. Users can test data against any distribution by providing custom pdf, cdf, and survival functions. The package supports both normal approximation and Monte Carlo simulation approaches for computing p-values and critical values.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-08-02 16:40:02 UTC

Contents

`Gofpt2-package` *Gofpt2: Generalized Goodness-of-Fit Test for Progressive Type-II Censored Data*

Description

Implements a generalized goodness-of-fit test based on spacings for general progressive Type-II censored data. The test statistic is based on the work of Qin et al. (2022) and extends the methodology of Balakrishnan et al. (2003). Users can test data against any distribution by providing custom pdf, cdf, and survival functions. The package supports both normal approximation and Monte Carlo simulation approaches for computing p-values and critical values.

Author(s)

Maintainer: Shikhar Tyagi <shikhar1093tyagi@gmail.com> (ORCID)

Authors:

- Arvind Pandey <arvindmzu@gmail.com>
- Bhupendra Singh <bhupendra.rana@gmail.com>
- Vrijesh Tripathi <vrijesh.tripathi@uwi.edu>

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. Journal of Applied Statistics, 49(8), 1821-1636. doi:10.1080/02664763.2020.1821613

Balakrishnan, N., Ng, H.K.T., & Kannan, N. (2003). A test of exponentiality based on spacings for progressively type-II censored data. In Huber-Carol, C., Balakrishnan, N., Nikulin, M.S., & Mesbah, M. (Eds.), Goodness-of-Fit Tests and Model Validity (pp. 89-111). Birkhäuser, Boston, MA. doi:10.1007/978-1-4612-0103-8_8

Balakrishnan, N., & Aggarwala, R. (2000). Progressive Censoring: Theory, Methods, and Applications. Birkhäuser, Boston.

Lawless, J.F. (2003). Statistical Models and Methods for Lifetime Data (2nd ed.). John Wiley & Sons, New York.

compute_spacings *Compute Spacings from Censored Data*

Description

Computes the spacings $S_{r+1}, S_{r+2}, \ldots, S_m$ from progressive Type-II censored data according to equation (5) in Qin et al. (2022).

Usage

```
compute_spacings(
  data,
  censoring_scheme,
  cdf_func = NULL,
  survival_func = NULL,
  parameters = list()
)
```

Arguments

`data`	Numeric vector of observed failure times (length $m - r$).
`censoring_scheme`	A list containing `n` (total sample size), `m` (number of observed failures), and `r` (removals vector or initial unobserved count).
`cdf_func`	Optional cumulative distribution function of hypothesized distribution.
`survival_func`	Optional survival function of hypothesized distribution.
`parameters`	List of distribution parameters.

Value

A numeric vector of computed spacings $S_{r+1}, \ldots, S_m$ of length $m - r$.

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. *Journal of Applied Statistics*, 49(8), 1821-1636. doi:10.1080/02664763.2020.1821613

Examples

```
scheme <- list(n = 19, m = 11, r = 2, R = c(0, 0, 2, 0, 0, 2, 0, 0, 4))
obs_data <- c(3.16, 4.15, 4.67, 7.35, 8.01, 8.27, 32.52, 33.91, 36.71)
compute_spacings(obs_data, scheme)
```

`critical_values` *Compute Critical Values for Goodness-of-Fit Test*

Description

Computes lower and upper critical values of the test statistic T for a specified significance level α using either normal approximation or Monte Carlo simulation.

Usage

```
critical_values(
  censoring_scheme,
  alpha = 0.05,
  method = c("normal", "simulation"),
  n_sim = 10000
)
```

Arguments

`censoring_scheme`	A list containing `n` (total sample size), `m` (number of observed failures), and `r` (removals vector or initial unobserved count).
`alpha`	Numeric, significance level (default: 0.05).
`method`	Character string, either `"normal"` for normal approximation or `"simulation"` for Monte Carlo simulation.
`n_sim`	Integer, number of simulations for Monte Carlo method (default: 10000).

Value

A named numeric vector of length 2 containing `lower` and `upper` critical bounds.

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. *Journal of Applied Statistics*, 49(8), 1821-1636. doi:10.1080/02664763.2020.1821613

Examples

```
scheme <- list(n = 19, m = 11, r = 2, R = c(0, 0, 2, 0, 0, 2, 0, 0, 4))
critical_values(scheme, alpha = 0.05, method = "normal")
```

`expected_spacings` *Compute Expected Spacings under Null Hypothesis*

Description

Calculates the expected values of spacings $E(S_i)$ under the standard exponential null distribution for a given censoring scheme.

Usage

```
expected_spacings(censoring_scheme)
```

Arguments

`censoring_scheme`
: A list containing `n` (total sample size), `m` (number of observed failures), and `r` (removals vector or initial unobserved count).

Value

A numeric vector of expected spacings $E(S_{r+1}), \ldots, E(S_m)$.

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. *Journal of Applied Statistics*, 49(8), 1821-1636.

Examples

```
scheme <- list(n = 19, m = 11, r = 2, R = c(0, 0, 2, 0, 0, 2, 0, 0, 4))
expected_spacings(scheme)
```

`generate_progressive_censored`
Generate General Progressive Type-II Censored Data

Description

Generates progressive Type-II censored samples from a specified probability distribution.

Usage

```
generate_progressive_censored(
  censoring_scheme,
  pdf_func = NULL,
  cdf_func = NULL,
  survival_func = NULL,
  quantile_func = NULL,
  parameters = list()
)
```

Arguments

censoring_scheme	A list containing n (total sample size), m (number of observed failures), and r (removals vector or initial unobserved count).
pdf_func	Optional probability density function of target distribution.
cdf_func	Optional cumulative distribution function of target distribution.
survival_func	Optional survival function of target distribution.
quantile_func	Optional quantile function (inverse CDF) of target distribution.
parameters	List of distribution parameters.

Value

A numeric vector of length m - r containing ordered failure times.

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. *Journal of Applied Statistics*, 49(8), 1821-1636.

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

Examples

```
scheme <- list(n = 19, m = 11, r = 2, R = c(0, 0, 2, 0, 0, 2, 0, 0, 1))
generate_progressive_censored(scheme)
```

---

gof_test_censored *Generalized Goodness-of-Fit Test for Progressive Type-II Censored Data*

Description

Performs a generalized goodness-of-fit test based on spacings for general progressive Type-II censored data as proposed by Qin et al. (2022).

Usage

```
gof_test_censored(
  data,
  censoring_scheme,
  pdf_func = NULL,
  cdf_func = NULL,
  survival_func = NULL,
  method = c("normal", "simulation"),
  n_sim = 10000,
  alpha = 0.05,
  parameters = list()
)
```

Arguments

`data`	Numeric vector of observed failure times (ordered).
`censoring_scheme`	A list containing `n` (total sample size), `m` (number of observed failures), and `r` (removals vector or initial unobserved count).
`pdf_func`	Optional probability density function of hypothesized distribution.
`cdf_func`	Optional cumulative distribution function of hypothesized distribution.
`survival_func`	Optional survival function of hypothesized distribution.
`method`	Character string, either `"normal"` for normal approximation or `"simulation"` for Monte Carlo simulation.
`n_sim`	Integer, number of simulations for Monte Carlo method (default: 10000).
`alpha`	Numeric, significance level for critical values (default: 0.05).
`parameters`	List of distribution parameters.

Value

An object of class `"gof_censored"` containing:

`test_statistic`	Value of the test statistic T.
`p_value`	Computed p-value.
`critical_value`	Named vector of lower and upper critical values at significance level α.
`method`	Method used for computation (`"normal"` or `"simulation"`).
`mu`	Expected mean of test statistic T under null distribution.
`sigma`	Standard deviation of test statistic T under null distribution.
`censoring_scheme`	Parsed input censoring scheme.
`data`	Validated input failure data.
`spacings`	Computed spacings vector.
`alpha`	Significance level.
`parameters`	Distribution parameters.
`distribution`	Name of hypothesized distribution.

References

Qin, X., Gui, W., & Balakrishnan, N. (2022). A goodness-of-fit test for exponential distribution based on spacings for general progressive Type-II censored data. *Journal of Applied Statistics*, 49(8), 1821-1636. doi:10.1080/02664763.2020.1821613

Balakrishnan, N., Ng, H.K.T., & Kannan, N. (2003). A test of exponentiality based on spacings for progressively type-II censored data. In *Goodness-of-Fit Tests and Model Validity* (pp. 89-111). Birkhäuser, Boston, MA.

Examples

```
scheme <- list(n = 19, m = 11, r = 2, R = c(0, 0, 2, 0, 0, 2, 0, 0, 4))
obs_data <- c(3.16, 4.15, 4.67, 7.35, 8.01, 8.27, 32.52, 33.91, 36.71)

# Test for Exponential distribution using Normal approximation
fit <- gof_test_censored(obs_data, scheme, method = "normal")
print(fit)
summary(fit)
```

Index