

Package ‘TieFreeCensor’

August 21, 2026

Type Package

Title Algorithm for Generating Tie-Free Progressive Type-II Censored Samples

Version 0.1.0

Description Generates tie-free progressive Type-II censored samples from discrete distributions and user-specified discrete probability mass functions (PMF) or cumulative distribution functions (CDF). Provides maximum likelihood estimation (MLE), Bayesian estimation via Markov chain Monte Carlo (MCMC) Metropolis-within-Gibbs sampling, likelihood-based parametric bootstrap goodness-of-fit (GOF) tests, profile log-likelihood diagnostics, and discrete survival and probability calculations. Methods are based on Ahmad and Mansour (2026) <[doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)>, Balakrishnan and Dembinska (2008) <[doi:10.1016/j.jspi.2007.02.006](https://doi.org/10.1016/j.jspi.2007.02.006)>, Joe and Zhu (2005) <[doi:10.1002/bimj.200410102](https://doi.org/10.1002/bimj.200410102)>, and Balakrishnan and Aggarwala (2000, ISBN:978-1-4612-1334-5).

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports stats, graphics

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

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 CRAN

Date/Publication 2026-08-21 12:41:06 UTC

Contents

coef.gpd_mle	2
covid_deaths	3
fit_gpd_bayes	3
fit_gpd_mle	5
generate_tiefree_p2c	6
gof_p2c_bootstrap	7
gpd_cdf	9
gpd_pmf	9
gpd_random	10
logLik.gpd_mle	11
plot.gpd_bayes	11
plot_profile_loglik	12
print.gof_bootstrap	13
print.gpd_bayes	13
print.gpd_mle	14
print.tiefree_p2c	14
prob_gpd	15
spinal_tumors	15
summary.gpd_bayes	16
summary.gpd_mle	17
vcov.gpd_mle	17
Index	18

coef.gpd_mle	<i>Coef Method for GPD MLE</i>
--------------	--------------------------------

Description

Coef Method for GPD MLE

Usage

```
## S3 method for class 'gpd_mle'  
coef(object, ...)
```

Arguments

- object Object of class "gpd_mle".
- ... Additional arguments.

Value

A named numeric vector of length 2 containing the maximum likelihood estimates for parameters q (mean parameter) and h (dispersion parameter).

covid_deaths

*COVID-19 Daily Death Counts Dataset***Description**

Dataset containing 25 daily COVID-19 death counts recorded in the US Virgin Islands during May 2021, as reported by Worldometer (2021) and analyzed by Ahmad and Mansour (2026).

Usage

```
covid_deaths
```

Format

A numeric vector of length 25 containing daily reported death counts.

Source

Worldometer (2021). COVID-19 Coronavirus Pandemic.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078.

fit_gpd_bayes

*Bayesian Estimation for GPD Under Progressive Type-II Censoring***Description**

Fits the Generalized Poisson Distribution (GPD) to a tie-free progressively Type-II right-censored sample using Bayesian Markov chain Monte Carlo (MCMC) with Metropolis-within-Gibbs sampling on transformed parameters.

Usage

```
fit_gpd_bayes(
  x,
  R,
  hyper_a = 1,
  hyper_b = 1,
  hyper_c = 10,
  iter = 5000,
  burnin = 1000,
  thin = 1,
  start = NULL,
```

```

    prop_sd = c(0.05, 0.05),
    conf_level = 0.95
  )

```

Arguments

x	Numeric vector of observed tie-free failure times ($x_1 < x_2 < \dots < x_m$).
R	Numeric vector of progressive removal counts ($R_1, R_2, \dots, R_m$).
hyper_a	Shape parameter $a > 0$ for Beta prior on h . Default is 1 (Uniform).
hyper_b	Shape parameter $b > 0$ for Beta prior on h . Default is 1 (Uniform).
hyper_c	Scale parameter $c > 0$ for Exponential prior on q . Default is 10.
iter	Total number of MCMC iterations. Default is 5000.
burnin	Number of initial iterations discarded as burn-in. Default is 1000.
thin	Thinning interval. Default is 1.
start	Numeric vector of length 2 giving starting values $c(q, h)$. If NULL, defaults to sample estimates.
prop_sd	Numeric vector of length 2 specifying proposal standard deviations for $(\log q, \logit h)$. Default is $c(0.05, 0.05)$.
conf_level	Credible level for equal-tailed credible intervals (CRI). Default is 0.95.

Value

A list of class "gpd_bayes" containing:

- estimates: Vector of posterior means (SEL) for q and h .
- sd: Posterior standard deviations.
- cri: Matrix of equal-tailed credible intervals (2.5% and 97.5% quantiles).
- draws: Matrix of thinned MCMC posterior samples for q and h .
- accept_rates: Acceptance rates for MCMC updates.
- data: List containing original input data.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)

Examples

```

# NF2 Spinal Tumors dataset example from paper
x_tumor <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16)
R_tumor <- c(69, 12, 14, 5, 6, 4, 8, 8, 0, 1, 4, 0, 0, 0, 0, 11)
set.seed(123)
fit_b <- fit_gpd_bayes(x = x_tumor, R = R_tumor, iter = 2000, burnin = 500)
print(fit_b)
summary(fit_b)

```

fit_gpd_mle	<i>Maximum Likelihood Estimation for GPD Under Progressive Type-II Censoring</i>
-------------	--

Description

Fits the Generalized Poisson Distribution (GPD) to a tie-free progressively Type-II right-censored sample using Maximum Likelihood Estimation (MLE).

Usage

```
fit_gpd_mle(x, R, start = NULL, method = "BFGS", conf_level = 0.95, ...)
```

Arguments

x	Numeric vector of observed tie-free failure times $(x_1 < x_2 < \dots < x_m)$.
R	Numeric vector of progressive removal counts $(R_1, R_2, \dots, R_m)$.
start	Numeric vector of length 2 giving initial parameter values $c(q, h)$. If NULL, starting values are calculated using sample moments.
method	Optimization method passed to optim . Options include "BFGS", "Nelder-Mead", "L-BFGS-B", "CG". Default is "BFGS".
conf_level	Confidence level for asymptotic confidence intervals (ACI). Default is 0.95.
...	Additional control parameters passed to optim .

Value

An object of class "gpd_mle" containing:

- par: Vector of estimated parameters $(\hat{q}, \hat{h})$.
- se: Vector of standard errors for $\hat{q}$ and $\hat{h}$.
- vcov: Estimated variance-covariance matrix.
- loglik: Maximum log-likelihood value.
- aci: Matrix of asymptotic confidence intervals.
- convergence: Optimization convergence code.
- data: List containing original data x and removal scheme R.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)

Examples

```
# NF2 Spinal Tumors dataset example from paper
x_tumor <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16)
R_tumor <- c(69, 12, 14, 5, 6, 4, 8, 8, 0, 1, 4, 0, 0, 0, 0, 11)
fit_mle <- fit_gpd_mle(x = x_tumor, R = R_tumor)
print(fit_mle)
summary(fit_mle)
```

generate_tiefree_p2c *Generate Tie-Free Progressive Type-II Censored Samples*

Description

Implements the constructive algorithm described by Ahmad and Mansour (2026) to generate progressively Type-II right-censored samples that are guaranteed to be free of ties, even when the underlying discrete distribution produces repeated values.

Usage

```
generate_tiefree_p2c(
  x0 = NULL,
  n = NULL,
  m = NULL,
  R = NULL,
  dist = c("gpd", "poisson", "geometric", "nbinom", "custom"),
  dist_pmf = NULL,
  dist_cdf = NULL,
  params = list(),
  max_attempts = 100
)
```

Arguments

x0	Numeric vector representing the original complete discrete dataset (with potential ties), or NULL if generating from a distribution.
n	Target total sample size (if generating from distribution). Defaults to length(x0).
m	Target number of observed failures. If NULL or larger than the maximum available distinct values, it is automatically adjusted.
R	Numeric vector specifying the progressive removal scheme $(R_1, R_2, \dots, R_m)$ such that $\sum R_i = n - m$. If NULL, a default uniform or right-heavy scheme is generated.
dist	Character string specifying built-in discrete distribution if x0 is NULL. Options: "gpd" (Generalized Poisson), "poisson", "geometric", "nbinom".
dist_pmf	Custom function for discrete PMF $P(X = x)$ taking quantiles as first argument. Used when x0 is NULL and dist = "custom".

dist_cdf	Custom function for discrete CDF $P(X \leq x)$ taking quantiles as first argument.
params	List of parameter values passed to distribution functions.
max_attempts	Maximum number of simulation attempts to find a tie-free sample before reducing candidate m .

Value

A list of class "tiefree_p2c" containing:

x	Numeric vector of tie-free observed progressive Type-II failure times $(x_{(1)} < x_{(2)} < \dots < x_{(m)})$.
R	Numeric vector of final progressive removal counts.
n	Total original sample size.
m	Final number of observed failure times.
x0	Original discrete dataset (with ties).
m_max	Number of distinct values in the original dataset.

References

- Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)
- Balakrishnan, N., & Dembińska, A. (2008). Progressively Type-II right censored order statistics from discrete distributions. *Journal of Statistical Planning and Inference*, 138(4), 845-856.

Examples

```
set.seed(123)
# Generate tie-free progressive Type-II sample from GPD
res <- generate_tiefree_p2c(n = 30, m = 15, dist = "gpd", params = list(q = 3.0, h = 0.3))
print(res)

# Generate tie-free sample from a raw dataset with ties
raw_data <- c(0,0,0,1,1,2,2,3,4,4,5,6,6,7,8,9,10)
res_raw <- generate_tiefree_p2c(x0 = raw_data, m = 8)
print(res_raw)
```

Description

Evaluates model adequacy for progressively Type-II censored discrete data using a likelihood-based parametric bootstrap test, as formulated in Section 6.1 of Ahmad and Mansour (2026).

Usage

```
gof_p2c_bootstrap(x, R, B = 100, dist = "gpd", alpha = 0.05)
```

Arguments

x	Numeric vector of observed tie-free failure times $(x_1 < x_2 < \dots < x_m)$.
R	Numeric vector of progressive removal counts $(R_1, R_2, \dots, R_m)$.
B	Number of bootstrap replications. Default is 100.
dist	Model family under test. Default is "gpd".
alpha	Significance level for hypothesis testing. Default is 0.05.

Value

An object of class "gof_bootstrap" containing:

- statistic: Observed test statistic $T^{(obs)} = -2 \log L(\hat{q}, \hat{h})$.
- p_value: Monte Carlo bootstrap p -value.
- mle: MLE fit object for the observed dataset.
- bootstrap_stats: Vector of bootstrap test statistics $T^{(b)} = -2 \log L(\hat{q}^{(b)}, \hat{h}^{(b)})$.
- B: Number of successful bootstrap replicates.
- decision: Hypothesis test conclusion ("Fail to reject H0" or "Reject H0").

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)

Besag, J., & Clifford, P. (1991). Sequential Monte Carlo p-values. *Biometrika*, 78(2), 301-304.

Examples

```
# NF2 Spinal Tumors dataset test
x_tumor <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16)
R_tumor <- c(69, 12, 14, 5, 6, 4, 8, 8, 0, 1, 4, 0, 0, 0, 0, 11)
set.seed(123)
test_res <- gof_p2c_bootstrap(x = x_tumor, R = R_tumor, B = 20)
print(test_res)
```

gpd_cdf

*Generalized Poisson Distribution Cumulative Distribution Function***Description**

Evaluates the cumulative distribution function (CDF) of the Generalized Poisson Distribution (GPD).

Usage

```
gpd_cdf(x, q, h, lower_tail = TRUE, log_p = FALSE)
```

Arguments

x	Numeric vector of quantiles.
q	Parameter $q > 0$.
h	Parameter $0 \leq h < 1$.
lower_tail	Logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise $P(X > x)$.
log_p	Logical; if TRUE, probabilities are given as $\log(p)$. Default is FALSE.

Value

Numeric vector of CDF values.

Examples

```
gpd_cdf(x = 5, q = 3.0, h = 0.2)
```

gpd_pmf

*Generalized Poisson Distribution (GPD) Probability Mass Function***Description**

Evaluates the probability mass function (PMF) of the Generalized Poisson Distribution (GPD).

Usage

```
gpd_pmf(x, q, h, log_p = FALSE)
```

Arguments

x	Numeric vector of non-negative integer quantiles.
q	Parameter $q > 0$ (mean parameter when $h = 0$).
h	Parameter $0 \leq h < 1$ (dispersion parameter).
log_p	Logical; if TRUE, probabilities are given as $\log(p)$. Default is FALSE.

Value

Numeric vector of PMF values.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)

Joe, H., & Zhu, R. (2005). Generalized Poisson distribution: the property of mixture of Poisson and comparison with negative binomial distribution. *Biometrical Journal*, 47(2), 219-229.

Examples

```
gpd_pmf(x = 0:10, q = 3.0, h = 0.2)
```

gpd_random

Random Generation for Generalized Poisson Distribution

Description

Generates random variates from the Generalized Poisson Distribution (GPD).

Usage

```
gpd_random(n, q, h)
```

Arguments

n	Number of observations to sample.
q	Parameter $q > 0$.
h	Parameter $0 \leq h < 1$.

Value

Numeric vector of random counts.

Examples

```
set.seed(123)
gpd_random(n = 20, q = 3.0, h = 0.2)
```

logLik.gpd_mle	<i>LogLik Method for GPD MLE</i>
----------------	----------------------------------

Description

LogLik Method for GPD MLE

Usage

```
## S3 method for class 'gpd_mle'
logLik(object, ...)
```

Arguments

object	Object of class "gpd_mle".
...	Additional arguments.

Value

An object of class "logLik" containing the maximum log-likelihood value of the fitted GPD model, with attributes "df" (degrees of freedom, equal to 2) and "nobs" (number of observed failures m).

plot.gpd_bayes	<i>Plot Method for GPD Bayesian MCMC Diagnostics</i>
----------------	--

Description

Plot Method for GPD Bayesian MCMC Diagnostics

Usage

```
## S3 method for class 'gpd_bayes'
plot(x, ...)
```

Arguments

x	Object of class "gpd_bayes".
...	Additional arguments.

Value

No return value, called for side effects. Generates MCMC diagnostic trace plots and posterior distribution histograms for parameters q and h .

plot_profile_loglik *Plot Profile Log-Likelihood Curves for GPD Parameters*

Description

Computes and plots the profile log-likelihood curves for the parameters q and h of the Generalized Poisson Distribution (GPD) under progressive Type-II censoring (Figures 3 and 4 in Ahmad and Mansour, 2026).

Usage

```
plot_profile_loglik(
  x,
  R,
  param = c("both", "q", "h"),
  grid_len = 50,
  q_range = NULL,
  h_range = NULL,
  main = NULL
)
```

Arguments

x	Numeric vector of observed tie-free failure times $(x_1 < x_2 < \dots < x_m)$.
R	Numeric vector of progressive removal counts $(R_1, R_2, \dots, R_m)$.
param	Character string specifying parameter to profile: "q" or "h". If "both" (default), produces side-by-side plots for both parameters.
grid_len	Integer specifying the number of evaluation points along the profile grid. Default is 50.
q_range	Numeric vector of length 2 specifying range for q . If NULL, automatically determined around MLE.
h_range	Numeric vector of length 2 specifying range for h . If NULL, automatically determined around MLE.
main	Optional title for plot.

Value

Invisible list containing grid points and profile log-likelihood values.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078. [doi:10.1155/jom/3657078](https://doi.org/10.1155/jom/3657078)

Examples

```
# NF2 Spinal Tumors dataset profile log-likelihood
x_tumor <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16)
R_tumor <- c(69, 12, 14, 5, 6, 4, 8, 8, 0, 1, 4, 0, 0, 0, 0, 11)
plot_profile_loglik(x = x_tumor, R = R_tumor, param = "both")
```

```
print.gof_bootstrap      Print Method for GOF Bootstrap Test Result
```

Description

Print Method for GOF Bootstrap Test Result

Usage

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

Arguments

`x` Object of class "gof_bootstrap".
`...` Additional arguments.

Value

Invisibly returns the input object `x` of class "gof_bootstrap". Called for its side effect of printing the goodness-of-fit test statistic, bootstrap replicates, Monte Carlo p-value, and test decision to the console.

```
print.gpd_bayes          Print Method for GPD Bayesian MCMC Estimation
```

Description

Print Method for GPD Bayesian MCMC Estimation

Usage

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

Arguments

`x` Object of class "gpd_bayes".
`...` Additional arguments.

Value

Invisibly returns the input object `x` of class `"gpd_bayes"`. Called for its side effect of printing Bayesian posterior point estimates and MCMC acceptance rates to the console.

<code>print.gpd_mle</code>	<i>Print Method for GPD MLE</i>
----------------------------	---------------------------------

Description

Print Method for GPD MLE

Usage

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

Arguments

<code>x</code>	Object of class <code>"gpd_mle"</code> .
<code>...</code>	Additional arguments.

Value

Invisibly returns the input object `x` of class `"gpd_mle"`. Called for its side effect of printing the GPD maximum likelihood parameter estimates and log-likelihood value to the console.

<code>print.tiefree_p2c</code>	<i>Print Method for Tie-Free Progressive Type-II Sample</i>
--------------------------------	---

Description

Print Method for Tie-Free Progressive Type-II Sample

Usage

```
## S3 method for class 'tiefree_p2c'
print(x, ...)
```

Arguments

<code>x</code>	An object of class <code>"tiefree_p2c"</code> .
<code>...</code>	Additional arguments.

Value

Invisibly returns the input object `x` of class `"tiefree_p2c"`. Called for its side effect of printing details of the tie-free progressive Type-II sample to the console.

prob_gpd	<i>Compute Discrete Probabilities and Event Metrics for GPD</i>
----------	---

Description

Computes specific discrete probabilities such as $P(X = x)$, $P(X > k)$, $P(X \geq k)$, $P(X < k)$, or $P(X \leq k)$.

Usage

```
prob_gpd(q, h, type = c("eq", "gt", "gte", "lt", "lte"), x = 0, k = 0)
```

Arguments

- q Parameter $q > 0$.
- h Parameter $0 \leq h < 1$.
- type Character string indicating the probability type: "eq" for $P(X = x)$, "gt" for $P(X > k)$, "gte" for $P(X \geq k)$, "lt" for $P(X < k)$, "lte" for $P(X \leq k)$.
- x Non-negative integer for type = "eq".
- k Non-negative integer bound for inequalities.

Value

Numeric probability value.

Examples

```
prob_gpd(q = 5.62, h = 0.75, type = "eq", x = 0)
prob_gpd(q = 5.62, h = 0.75, type = "gt", k = 10)
```

spinal_tumors	<i>Spinal Tumors Count Dataset in Neurofibromatosis Type 2 (NF2) Patients</i>
---------------	---

Description

Dataset containing spinal tumor counts from 158 patients with Neurofibromatosis Type 2 (NF2), originally presented by Joe and Zhu (2005) and analyzed under progressive censoring by Ahmad and Mansour (2026).

Usage

```
spinal_tumors
```

Format

A data frame with 22 rows and 2 variables:

- tumor_count** Number of spinal tumors observed in a patient.
- frequency** Number of patients exhibiting the given tumor count.

Source

Joe, H., & Zhu, R. (2005). Generalized Poisson distribution: the property of mixture of Poisson and comparison with negative binomial distribution. *Biometrical Journal*, 47(2), 219-229.

References

Ahmad, H. H., & Mansour, M. M. M. (2026). An Algorithm for Generating Tie-Free Progressive Type-II Censored Samples From Discrete Distributions. *Journal of Mathematics*, 2026, 3657078.

summary.gpd_bayes	Summary Method for GPD Bayesian MCMC Estimation
-------------------	---

Description

Summary Method for GPD Bayesian MCMC Estimation

Usage

```
## S3 method for class 'gpd_bayes'  
summary(object, ...)
```

Arguments

- object Object of class "gpd_bayes".
- ... Additional arguments.

Value

Invisibly returns the input object object of class "gpd_bayes". Called for its side effect of displaying posterior point estimates, posterior standard errors, and credible intervals in tabular format.

summary.gpd_mle	<i>Summary Method for GPD MLE</i>
-----------------	-----------------------------------

Description

Summary Method for GPD MLE

Usage

```
## S3 method for class 'gpd_mle'
summary(object, ...)
```

Arguments

object	Object of class "gpd_mle".
...	Additional arguments.

Value

Invisibly returns the input object object of class "gpd_mle". Called for its side effect of printing a detailed summary matrix containing MLE estimates, standard errors, and asymptotic confidence intervals.

vcov.gpd_mle	<i>Vcov Method for GPD MLE</i>
--------------	--------------------------------

Description

Vcov Method for GPD MLE

Usage

```
## S3 method for class 'gpd_mle'
vcov(object, ...)
```

Arguments

object	Object of class "gpd_mle".
...	Additional arguments.

Value

A 2x2 numeric variance-covariance matrix of class "matrix" with row and column names c("q", "h"), containing the estimated asymptotic covariances of the maximum likelihood estimates.

Index

* datasets

covid_deaths, [3](#)
spinal_tumors, [15](#)

coef.gpd_mle, [2](#)
covid_deaths, [3](#)

fit_gpd_bayes, [3](#)
fit_gpd_mle, [5](#)

generate_tiefree_p2c, [6](#)
gof_p2c_bootstrap, [7](#)
gpd_cdf, [9](#)
gpd_pmf, [9](#)
gpd_random, [10](#)

logLik.gpd_mle, [11](#)

optim, [5](#)

plot.gpd_bayes, [11](#)
plot_profile_loglik, [12](#)
print.gof_bootstrap, [13](#)
print.gpd_bayes, [13](#)
print.gpd_mle, [14](#)
print.tiefree_p2c, [14](#)
prob_gpd, [15](#)

spinal_tumors, [15](#)
summary.gpd_bayes, [16](#)
summary.gpd_mle, [17](#)

vcov.gpd_mle, [17](#)