Package 'fssg'

June 3, 2026

Title Parametric Survival Modeling in Bulk

Version 1.0.0

Description A simple tool for the bulk creation and testing of parametric survival models. Simply provide 'fssg' with a formula and some data, and let it identify the best distributions for you.

License MIT + file LICENSE

URL https://github.com/jmrothen/fssg

BugReports https://github.com/jmrothen/fssg/issues

Encoding UTF-8

LinkingTo Rcpp

Imports actuar, dplyr, extraDistr, flexsurv, magrittr, methods, Rcpp, rstudioapi, stats, survAUC, survival, SurvMetrics, tictoc, VGAM

Suggests knitr, rmarkdown, testthat (>= 3.0.0), splines2

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author John Rothen [aut, cre, cph] (ORCID: <https://orcid.org/0009-0008-4897-8004>)

Maintainer John Rothen <jmrothen.business@gmail.com>

Repository CRAN

Date/Publication 2026-06-03 13:10:08 UTC

Contents

`check_inits` *Function to check if times can be calculated using the distribution with default inits.*

Description

Function to check if times can be calculated using the distribution with default inits.

Usage

```
check_inits(times, distribution)
```

Arguments

`times`	`Surv` object or numeric vector.
`distribution`	A distribution object from `fssg_dist_lists`.

Details

This should work with all `fssg` custom distribution, but does not work on some of the native `flexsurv` distributions.

Value

Boolean indicator for success. If true, then all values can be calculated, and life is good.

Examples

```
# choose a distribution
dist <- get_fssg_dist('gamma_gompertz')

# identify all of the actual survival times
times <- rpois(1000, 100) # simulated times

# check if the distribution can be calculated at each time with default inits
check_inits(times, dist)
```

`derlang` *Erlang Density Function*

Description

Provides the probability density function at point x for an erlang distribution of parameters k and lambda. X, k, and l can be vectors.

Usage

```
derlang(x, k, l, log = FALSE)
```

Arguments

`x`	vector of quantiles.
`k`	shape parameter, positive integer.
`l`	short for lambda, rate parameter, must be greater than zero.
`log`	logical: if TRUE, log of probability is returned.

Value

The probability of the Erlang probability distribution at x with parameters K and lambda.

References

https://quarto.wessa.net/erlang.html

https://en.wikipedia.org/wiki/Erlang_distribution

Examples

```
derlang(1, 1, 1)
```

dgamgomp *Gamma-Gompertz Distribution Function*

Description

Provides probability density function for Gamma-Gompertz distribution.

Usage

```
dgamgomp(x, b, sigma, beta, log = FALSE)
```

Arguments

x	vector of quantiles.
b	scale paramater, must be greater than 0.
sigma, beta	shape parameters, must be greater than 0.
log	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles x.

References

https://en.wikipedia.org/wiki/Gamma/Gompertz_distribution

Examples

```
dgamgomp(1,1,1,1)
```

dhypertab *Hypertabastic Distribution Function*

Description

Provides probability distribution function for Hypertabastic distribution.

Usage

```
dhypertab(x, a, b, log = FALSE)
```

Arguments

x	vector of quantiles.
a	alpha parameter. Must be greater than 0.
b	beta parameter. Must be greater than 0.
log	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles x.

References

https://en.wikipedia.org/wiki/Hypertabastic_survival_models

Examples

```
dhypertab(1,1,1)
```

---

dinvlind *Inverse Lindley Distribution Function*

Description

Providers probability distribution function for Inverse Lindley distribution.

Usage

```
dinvlind(x, theta, log = FALSE)
```

Arguments

x	vector of quantiles.
theta	paramater, must be greater than 0.
log	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles x.

References

Sharma, V. K., Singh, S. K., Singh, U., & Agiwal, V. (2015). The inverse Lindley distribution: a stress-strength reliability model with application to head and neck cancer data. Journal of Industrial and Production Engineering, 32(3), 162-173. doi:10.1080/21681015.2015.1025901

Asgharzadeh, Akbar & Alizadeh Sangtarashani, Mojtaba. (2023). Inverse Lindley distribution: different methods for estimating their PDF and CDF. Journal of Statistical Computation and Simulation. 94. 1-20. doi:10.1080/21681015.2015.1025901

Examples

```
dinvlind(1,1)
```

`dlogcauchy` *Log Cauchy Distribution Functions*

Description

Provides probability distribution function for Log Cauchy distribution.

Usage

```
dlogcauchy(x, mu, sigma, log = FALSE)
```

Arguments

`x`	vector of quantiles.
`mu`	location parameter, must be real.
`sigma`	scale parameter, must be greater than 0.
`log`	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles x.

References

https://en.wikipedia.org/wiki/Log-Cauchy_distribution

Examples

```
dlogcauchy(1,1,1)
```

`fssg` *fssg: Flexsurv "Shotgun".*

Description

A simple tool for the bulk creation and testing of parametric survival models.

Usage

```
fssg(
  formula,
  data = NA,
  models = NA,
  skip = c("default"),
  opt_method = "BFGS",
  spline = NA,
  max_knots = 1,
  dump_models = TRUE,
  detailed = FALSE,
  ibs = FALSE,
  progress = TRUE,
  warn = FALSE
)
```

Arguments

`formula`	Formula. Should be a survival formula, with a Surv object on the left hand side.
`data`	If your formula needs a dataset, provide that here.
`models`	Vector of strings. If you only want to run specific models, specify them here by their list name in `fssg_dist_list`.
`skip`	Vector. If you want to skip any specific models, you can add their names here. By default, some of the repetitive or incredibly niche models are skipped.
`opt_method`	String. Named of the preferred optimization method. Default for fssg is 'Nelder-Mead', with 'BFGS' being used as a back-up in case of errors. Can be any valid `optim` method, and the back-up method will always be 'BFGS', or 'Nelder-Mead' if 'BFGS' is the primary method provided.
`spline`	String or Vector of Strings. Include 'rp' or 'wy' for Royston-Parmar natural cubic spline, or Wang-Yan alternative natural cubic spline respectively. The Wang-Yan version requires the package `splines2ns`. If set to NA, then the spline step will be skipped. `fssg` runs spline models using all three available scale options in `flexsurvspline`, for 'hazard', 'odds' and 'normal.
`max_knots`	Integer. Specifies the maximum number of knots to be considered in spline models.
`dump_models`	Logical. If TRUE, each successful model will be placed into a list and returned.
`detailed`	Logical. If True, calculates a number of additional fit statistics for each model.

ibs	Logical. If TRUE, calculate integrated brier score for each model. Please note that this *greatly* increases run time, and is not recommended for large data.
progress	Logical. If TRUE, prints progress updates while the function runs.
warn	Logical. If TRUE, also prints any warnings that appear.

Details

Please see `vignette("fssg")` for a more in-depth example of the function.

Value

List containing a summary of the models generated. If dump_models is True, also returns a list of generated models.

Examples

```
library(survival)
fssg(
  Surv(time, status)~1,
  data=aml,
  models=c('genf','exp','dagum','lomax','rayleigh','gamma_gompertz'),
  spline = c('rp'),
  max_knots=2,
  warn = TRUE
)$summary
```

---

fssg_dist *fssg_dist*

Description

`flexsurv` allows for the creating of custom distribution objects, which must follow a specific format to be used in `flexsurv` functions. `fssg` uses a modified version of this formatting, which specifies the relevant distribution functions directly.

Usage

```
fssg_dist(
  name,
  pars,
  location,
  transforms,
  inv.transforms,
  inits,
  d,
  p,
  q = NA,
```

```
  h = NA,
  H = NA,
  fullname = ""
)
```

Arguments

name	Simple short hand name for the relevant distribution. flexsurv will search for the distribution functions using this name by default if the functions aren't explicitly defined. E.g. 'norm' for pnorm, dnorm, etc.
pars	Vector of parameter names which will be provided to the relevant distribution functions.
location	Name of the parameter which should be allowed to vary based on covariates. The name 'location' is an artifact of the original flexsurv methodology, where it was assumed that the distributions were from a location-scale family. flexsurv does provide the ability to allow for more than one parameter to vary through the 'anc' option, however this has not been adapted into fssg at this time.
transforms	Vector of the functions which should be used to scale each parameter to the real number line. If a parameter must be positive, then 'log' would scale the parameter to the real line. This is used to pass parameters through to the optimization function. If no transformation is needed, use 'identity'.
inv.transforms	Vector of the inverse transformation functions for parameters. E.g. If transforms is 'log', inv.transforms would be 'exp'.
inits	Function which will take a vector of times t and provide the initial parameter estimates. Ideally these should be estimated using the times from the relevant dataset, but can also be arbitrary initial values such as 1.
d	Density function of the relevant distribution, such as dnorm. If not supplied, flexsurv will search the global environment for a p-function using the name paramater. If one is not found, flexsurv will attempt to search for a hazard function instead (hnorm). If this fails, flexsurv will not be able to model.
p	Distribution function of the relevant distribution, such as pnorm. If not supplied, flexsurv will search the global environment for a p-function using the name paramater. In the case that one is not found, then flexsurv will estimate one using integration, which will slow the process considerably.
q	Quantile function for the relevant distribution, such as qnorm. fssg provides a helper function (quantilify) to approximate q functions based on a p function. Will be estimated if not provided.
h	Hazard function of the relevant distribution. Not required, but may be supplied instead of a d function if that is preferred. Will be estimated if not provided.
H	Cumulative hazard function. Will be estimated if not provided.
fullname	Alternative name(s) for the distribution. This is used for labeling of outputs generated by fssg.

Details

fssg distributions should be specified as a list, with the following attributes.

Important note: flexsurv by default only varies one model parameter (what is specified in the distributions as location) We can make more than one parameter vary using the anc parameter in flexsurvreg. Example: anc = list(shape1 = ~ var1 + var2, shape2 = ~ var3). This requires the ancillary parameters to be outright specified, which is distribution specific.

Value

A flexsurv-ready distribution object.

References

flexsurv vignette by Christopher H. Jackson https://CRAN.R-project.org/package=flexsurv

Examples

```
fssg_dist(
  name = 'betapr',
  pars= c('shape1','shape2','scale'),
  location='scale',
  transforms= c(log,log,log),
  inv.transforms= c(exp,exp,exp),
  inits= function(t){c(3, 2, 1)},  # can be improved
  d = extraDistr::dbetapr,
  p = extraDistr::pbetapr,
  fullname='beta_prime'
)
```

---

fssg_dist_list *Compiles list of available distributions*

Description

Compiles list of available distributions

Usage

```
fssg_dist_list()
```

Details

For details on all distributions, please see vignette("Distributions").

Value

a list of all possible distributions

Examples

```
fdl <- fssg_dist_list()
```

`get_fit_stats` *Collect fit statistics for a parametric survival model*

Description

Collect fit statistics for a parametric survival model

Usage

```
get_fit_stats(model, ibs = FALSE)
```

Arguments

`model`	Model object. Currently formatted to work with `flexsurvreg` objects, and may work for other types of models.
`ibs`	Logical. If True, calculates the integrated Brier Score, which is a helpful fit statistic but is *much* slower to calculate than all other statistics.

Details

For a table of fit statistics and their sources, please see the vignette `vignette("Fit_Statistics")` for more details.

Value

List of fit statistics for the model.

Please note that for concordance and AUC statistics, the ranks are arbitrarily sorted in one direction regardless of PH/AFT specification of the model. To account for this, the statistics returned are the max of (statistic, 1-statistic), which should always provide the correct value regardless of rank sort order.

Examples

```
library(survival)
library(flexsurv)

flexsurvreg(Surv(time,status) ~ age +sex, data=cancer, dist= 'weibull') -> model
get_fit_stats(model = model, ibs = FALSE)
```

get_fssg_dist *Function to return a specific distribution object.*

Description

Function to return a specific distribution object.

Usage

```
get_fssg_dist(dist_name)
```

Arguments

dist_name	Name of the distribution.

Value

Distribution object.

Examples

```
get_fssg_dist('weibull')
get_fssg_dist('frechet')
get_fssg_dist('gamma_gompertz')
```

helper fssg *Helper Functions*

Description

Functions that extrapolate the quantile, Survival, hazard, and cumulative hazard functions for a distribution based on the provided density and/or distribution function(s). These functions assume you have a p and d function formatted in the standard format found in native R distribution functions (such as pnorm and dnorm).

Usage

```
survivify(p_function)

hazardify(d_function, p_function)

cumhazardify(p_function)

quantilify(p_function)
```

Arguments

p_function	Distribution function. E.g. pnorm.
d_function	Density function. E.g. dnorm.

Details

survivify creates a function for 1-p(t). hazardify creates a function for d(t)/S(t). cumhazardify creates a function for -log(S(t)). quantilify approximates a quantile function based on p(t) using numeric root (uniroot).

Value

A new function of the desired type with the same parameters as the input functions.

survivify returns the survival function S(t).

hazardify returns the hazard function h(t).

cumhazardify returns the cumulative hazard function H(t)

quantilify returns an estimated quantile function q(t).

Examples

```
survivify(pnorm)
hazardify(dnorm, pnorm)
cumhazardify(pnorm)
quantilify(pnorm)
```

---

perlang *Erlang Cumulative Density Function*

Description

Provides the cumulative density function at point q for an erlang distribution of parameters k and lambda. X, k, and l can be vectors.

Usage

```
perlang(q, k, l, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	vector of quantiles.
k	shape parameter, positive integer.
l	short for lambda, rate parameter, must be greater than zero.
lower.tail	logical: if TRUE, returns densities from 0 to q, otherwise q to 1.
log.p	logical: if TRUE, log of probability is returned.

Value

The cumulative probability of the Erlang probability distribution at based on quantile q with parameters K and lambda.

References

https://quarto.wessa.net/erlang.html

https://en.wikipedia.org/wiki/Erlang_distribution

Examples

```
perlang(1, 1, 1)
```

---

pgamgomp *Gamma-Gompertz Cumulative Distribution Function*

Description

Provides cumulative density function for Gamma-Gompertz distribution.

Usage

```
pgamgomp(q, b, sigma, beta, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	vector of quantiles.
b	scale paramater, must be greater than 0.
sigma, beta	shape parameters, must be greater than 0.
lower.tail	logical: if TRUE, returns densities from 0 to q, otherwise q to 1.
log.p	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles q.

References

https://en.wikipedia.org/wiki/Gamma/Gompertz_distribution

Examples

```
pgamgomp(1,1,1,1)
```

phypertab *Hypertabastic Cumulative Distribution Function*

Description

Provides cumulative distribution function for Hypertabastic distribution.

Usage

```
phypertab(q, a, b, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	vector of quantiles.
a	alpha parameter. Must be greater than 0.
b	beta parameter. Must be greater than 0.
lower.tail	logical: if TRUE, returns densities from 0 to q, otherwise q to 1.
log.p	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles q.

References

https://en.wikipedia.org/wiki/Hypertabastic_survival_models

Examples

```
phypertab(1,1,1)
```

pinvlind *Inverse Lindley Distribution Function*

Description

Providers probability distribution function for Inverse Lindley distribution.

Usage

```
pinvlind(q, theta, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	vector of quantiles.
theta	paramater, must be greater than 0.
lower.tail	logical: if TRUE, returns densities from 0 to q, otherwise q to 1.
log.p	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles q.

References

Sharma, V. K., Singh, S. K., Singh, U., & Agiwal, V. (2015). The inverse Lindley distribution: a stress-strength reliability model with application to head and neck cancer data. Journal of Industrial and Production Engineering, 32(3), 162-173. doi:10.1080/21681015.2015.1025901

Asgharzadeh, Akbar & Alizadeh Sangtarashani, Mojtaba. (2023). Inverse Lindley distribution: different methods for estimating their PDF and CDF. Journal of Statistical Computation and Simulation. 94. 1-20. doi:10.1080/21681015.2015.1025901

Examples

```
pinvlind(1,1)
```

---

plogcauchy *Log Cauchy Cumulative Distribution Functions*

Description

Provides cumulative distribution function for Log Cauchy distribution.

Usage

```
plogcauchy(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	vector of quantiles.
mu	location parameter, must be real.
sigma	scale parameter, must be greater than 0.
lower.tail	logical: if TRUE, returns densities from 0 to q, otherwise q to 1.
log.p	logical: if TRUE, log of probability is returned.

Value

Probabilities for quantiles q.

References

https://en.wikipedia.org/wiki/Log-Cauchy_distribution

Examples

```
plogcauchy(1,1,1)
```

---

pseudo *Pseudo: Simulated data for experimentation*

Description

This data is entirely fabricated. The source code for creating this data set can be found in the data-raw folder of the github repository.

Usage

```
pseudo
```

Format

time Time until event.

death Indicator for death. If TRUE, the patient died at corresponding time.

gender, age, region, surgery, drug, comorb, comorb_cat Arbitrary covariates.

Source

fssg

Examples

```
head(pseudo)
```

---

quick_pipe *Simple add-in which lets you keyboard map the writing of pipe + newline. Intended to make functional programming pipelines a little easier on the hands.*

Description

Simple add-in which lets you keyboard map the writing of pipe + newline. Intended to make functional programming pipelines a little easier on the hands.

Usage

```
quick_pipe()
```

Index