

Package ‘WSPsignal’

January 8, 2026

Title Weibull Shape Parameter Tests For Signal Detection

Version 1.0.0

Description This is an R package to perform Bayesian or frequentist Weibull Shape Parameter (WSP) tests developed for signal detection in pharmacovigilance to flag associations between drugs and adverse events based on time-to-event data. The WSP test is based on the assumption of constant hazard reflected by a Weibull type distribution with shape parameters equal to one. Based on the shape parameter estimates (posterior distribution or point estimate), the WSP test method performs a hypothesis test on each shape parameter and combines them to a decision on the presence of a signal.

License MIT + file LICENSE

Encoding UTF-8

Roxxygen list(markdown = TRUE)

RoxxygenNote 7.3.3

Biarch true

Depends R (>= 3.5.0)

Imports methods,

Rcpp (>= 0.12.0),
RcppParallel (>= 5.0.1),
rstan (>= 2.18.1),
rstap (>= 2.4.0),
Rdpack,
HDInterval,
ROCR,
stats,
graphics,
dplyr,
ggplot2,
survival,
furrr

Suggests tidyverse

LinkingTo BH (>= 1.66.0),

Rcpp (>= 0.12.0),
RcppEigen (>= 0.3.3.3.0),
RcppParallel (>= 5.0.1),
rstan (>= 2.18.1),
StanHeaders (>= 2.18.0)

SystemRequirements GNU make

RdMacros Rdpack**LazyData** true

Contents

WSPsignal-package	2
bwsp_model	3
bwsp_test	4
eval.calc_perf	8
eval.eff_sample_sizes	10
eval.execution_times	10
eval.non_conv_cases	11
eval.rank_auc	12
eval.roc_curve	13
fwsp_model	14
fwsp_test	16
gamprior_repar	17
logprior_repar	18
muscu	20
pgw	21
plot_pgw	22
sim.datagen_tte	23
sim.merge_results	24
sim.priors_template	24
sim.run	26
sim.run_parallel	26
sim.setup_sim_pars	27
tte	30
tte2priordat	30

Index

33

WSPsignal-package *The 'WSPsignal' package.*

Description

An R package to perform Bayesian Weibull Shape Parameter (BWSP) tests for signal detection.

Author(s)

Maintainer: Julia Dyck <j.dyck@uni-bielefeld.de>

Authors:

- Odile Sauzet

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.6. <https://mc-stan.org>

bwsp_model	<i>Fit Bayesian model to time-to-event data</i>
------------	---

Description

Fits a Bayesian model to time-to-event (tte) data for the purpose of performing Weibull shape parameter signal detection tests with [bwsp_test](#).

Usage

```
bwsp_model(
  datstan,
  tte.dist = c("w", "dw", "pgw"),
  prior.dist = c("fg", "f1", "gg", "ll"),
  chains = 4,
  iter = 11000,
  warmup = 1000
)
```

Arguments

datstan	named list of data for the stanmodel; tte data can be formated with tte2priordat
tte.dist	character indicating the modelling approach; options are "w", "dw", "pgw" (see details)
prior.dist	character indicating the prior distribution for the parameters of the tte distribution; options are "fg", "f1", "gg", "ll" (see details)
chains	number of Markov chains to run
iter	total number of iterations per chain (including warmup)
warmup	number of warmup iterations per chain

Details

The function applies the [sampling](#) command with the No U-Turn sampler to fit a Bayesian model to time-to-event data. The model can be a Weibull ("w"), a double Weibull ("dw", estimating two Weibull models - one to the data as is and one to the data censored at mid of observation period), or a power generalized Weibull ("pgw") model.

The posterior is proportional to the likelihood times the prior. The likelihood is

$$\mathcal{L}(t|\Theta) = \prod_{i=1}^N S(t_i)^{1-d_i} \cdot f(t_i)^{d_i}$$

with $S(t)$ the survival function of the chosen distribution and $f(t)$ the density function (Nikulin et al. 2016). The pair (t_i, d_i) are the tte observations.

Implemented prior distributions for the scale and shape parameters are products of the following univariate distributional choices:

for scale parameter	for shape parameter(s)	abbreviation
fixed to prior mean	gamma	fg
gamma	gamma	gg

fixed to prior mean	lognormal	fl
lognormal	lognormal	ll

Value

A stanfit object or, in case of `tte.dist = "dw"`, a list of two stanfit objects with `$uncens` containing the stanfit object obtained from the model to the tte data as is, and `$cens` containing the stanfit object obtained from the model to the tte data censored at mid of observation period.

References

Nikulin M, Wu HI, others (2016). *The Cox model and its applications*. Springer.

Examples

```
# prep the data
head(tte)
standat = tte2priordat(dat = tte,
                       tte.dist = "pgw",
                       scale.mean = 1,
                       scale.sd = 10,
                       shape.mean = 1,
                       shape.sd = 10,
                       powershape.mean = 1,
                       powershape.sd = 10)

# fit a pgw model
fit = bwsp_model(datstan = standat,
                   tte.dist = "pgw",
                   prior.dist = "ll",
                   chains = 4,
                   iter = 110,           # (posterior sample
                   warmup = 10)          # is small for demo purpose)

# print the summary
fit
```

Description

Bayesian hypothesis test based on the shape parameter(s) of a distribution of the Weibull family.

Usage

```
bwsp_test(
  credregion,
  nullregion,
  tte.dist = c("w", "dw", "pgw"),
  option = c(1, 2, 3)
)
```

Arguments

credregion	vector of length 2 or 4 with the lower and upper boundaries of the credibility interval (CI) reflecting the posterior distribution of the shape parameter(s); required order: 1. lower CI boundary of first shape parameter, 2. upper CI boundary of first shape parameter; and if existent: 3. lower CI boundary of second shape parameter, 4. upper CI boundary of second shape parameter
nullregion	vector of length 2 or 4 with the lower and upper boundaries of the region of practical equivalence (ROPE) reflecting the null hypothesis region of the shape parameter(s); required order: 1. lower ROPE boundary of first shape parameter, 2. upper ROPE boundary of first shape parameter; and if existent: 3. lower ROPE boundary of second shape parameter, 4. upper ROPE boundary of second shape parameter
tte.dist	character specifying the modelling approach used to obtain the posterior samples; options are "w", "dw", "pgw";
option	numeric value out of 1, 2, 3; rule to be used to deduct a binary outcome (signal/no signal) from the HDI+ROPE test results of each shape parameter (see details)

Value

binary, 0 if H_0 is accepted, 1 if H_1 is rejected; see details for definition of H_0 and H_1

Test concept

The Bayesian Weibull shape parameter (WSP) test is a hypothesis test for signal detection of adverse drug reactions. It is based on the principle of non-constant hazard (Cornelius et al. 2012) which associates a constant hazard function with the absence of a drug-event association and a non-constant hazard function with the presence of a drug-event association.

This can be formalized as the following hypotheses depending on the underlying model:

	H_0	H_1
hypothesis	constant hazard function	non-constant hazard function
under Weibull model	$\nu = 1$	$\nu \neq 1$
under double Weibull model	$\nu_1 = 1$ and $\nu_2 = 1$	$\nu_1 \neq 1$ or $\nu_2 \neq 1$
under PGW model	$\nu = 1$ and $\gamma = 1$	$\nu \neq 1$ or $\gamma \neq 1$

Bayesian test components

Information on the Bayesian variant of the Power Generalized Weibull (PGW) shape parameter test can be found in Dyck and Sauzet (2025). The same concept applies to the construction of the Bayesian Weibull and double Weibull shape parameter test.

The region of practical equivalence (ROPE) specified in the nullregion argument represents the expected parameter value under H_0 . The credibility region(s) specified in the credregion argument represent the posterior distribution of each shape parameter. For the ROPE we recommend to set up an equal-tailed interval (ETI) with

$$[q_{(1-\alpha)/2}, q_{(1+\alpha)/2}]$$

based on the quantiles q of the shape parameters' prior distributions under H_0 at a chosen credibility level $1 - \alpha$. For the posterior CI, we recommend either an ETI at the same credibility level obtained

from the empirical quantiles of the posterior distribution per shape parameter or a highest density interval (HDI, Kruschke (2015))

$$HDI(\nu) = \{\nu \mid p_1(\nu) \geq w\} \text{ with } w \in [0, 1] \text{ such that } \int_{\nu \mid p_1(\nu) \geq w} p_1(\nu|t) d\nu = 1 - \alpha$$

at the same credibility level. See Examples for exemplary ROPE and post CI setup in R. Best choices for ROPE and CI can be obtained from a simulation study to tune the BWSP test specifications.

The HDI+ROPE test checks the relationship between ROPE and credibility region(s) leading to either acceptance, rejection or no decision regarding the null hypothesis for a single shape parameter. Options to generate a binary outcome, i.e. a signal or not, from HDI+ROPE test results based on one (in case of "w") or two (in case of "dw", "pgw") shape parameters are:

HDI+ROPE outcome for shape_1	HDI+ROPE outcome for shape_2	combination rule (option = 1)	combination rule (option = 2)	combination rule (option = 3)
rejection	(none)	signal	signal	signal
acceptance	(none)	-	-	-
no decision	(none)	signal	-	-
rejection	rejection	signal	signal	signal
acceptance	rejection	signal	-	-
rejection	acceptance	signal	-	-
acceptance	acceptance	-	-	-
no decision	rejection	signal	signal	-
no decision	acceptance	-	-	-
rejection	no decision	signal	signal	-
acceptance	no decision	-	-	-
no decision	no decision	signal	-	-

The hypotheses as stated above (see test concept) are implemented in option = 1 whereas option = 2 and option = 3 lead to a signal in fewer cases.

More details on the HDI+ROPE test, recommendations for interval specifications and the combination rules can be found in (Kruschke 2018) and (Dyck and Sauzet 2025).

References

- Cornelius VR, Sauzet O, Evans SJ (2012). “A signal detection method to detect adverse drug reactions using a parametric time-to-event model in simulated cohort data.” *Drug safety*, **35**, 599–610.
- Dyck J, Sauzet O (2025). “The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions.” preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.
- Kruschke J (2015). *Doing Bayesian Data Analysis (Second Edition)*. Academic Press, Boston.
- Kruschke JK (2018). “Rejecting or Accepting Parameter Values in Bayesian Estimation.” *Advances in Methods and Practices in Psychological Science*, **1**(2), 270-280. doi:[10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304), <https://doi.org/10.1177/2515245918771304>.

Examples

```
#### Exemplary conduction of a test from data and prior to test result:
```

```

# under weibull model:

# 1. specify ROPE reflecting the null hypothesis:
# we choose an 80% confidence interval around the
# null value (1 for both shape parameters)
logpars = logprior_repar(1, 10) # get parameters of a
                                # lognormal distribution with
                                # mean 1 and sd 10

rope = qlnorm(p = c(0.1,0.9), meanlog = logpars[1], sdlog = logpars[2])

# 2. Prior specification and model fitting:
# we formalize a prior belief (here "no association
# between drug and event", therefore prior mean = 1 for shape parameter)
# and reformat our tte data to fit the model in the following
standat = tte2priordat(dat = tte, # reformat the data
                       tte.dist = "w",
                       scale.mean = 1,
                       scale.sd = 10,
                       shape.mean = 1,
                       shape.sd = 10)

fit = bwsp_model(datstan = standat,      # fit the model
                  tte.dist = "w",
                  prior.dist = "ll",
                  chains = 4,
                  iter = 110,           # (posterior sample
                  warmup = 10)          # is small for demo purpose)

# 3. HDI specification and extraction:
# extract 80% HDIs representing posterior samples of the shape parameters
post.sample = rstan::extract(fit, pars = c("nu"))
nu.hdi = HDInterval::hdi(object = post.sample$nu, credMass = 0.8)

# 4. conduct the BWSPTtest
bwsp_test(credregion = nu.hdi,
           nullregion = rope,
           tte.dist = "w",
           option = 1)
# returns a signal

# under pgw model:

# 1. specify ROPE reflecting the null hypothesis:
# we choose an 80% confidence interval around the
# null value (1 for both shape parameters)

logpars = logprior_repar(1, 10) # get parameters of a
                                # lognormal distribution with
                                # mean 1 and sd 10

rope = qlnorm(p = c(0.1,0.9), meanlog = logpars[1], sdlog = logpars[2])

# 2. Prior specification and model fitting:
# we formalize a prior belief (here "no association
# between drug and event", therefore prior mean = 1 for both shape parameters)

```

```

# and reformat our tte data to fit the model in the following
standat = tte2priordat(dat = tte,           # reformat the data
                       tte.dist = "pgw",
                       scale.mean = 1,
                       scale.sd = 10,
                       shape.mean = 1,
                       shape.sd = 10,
                       powershape.mean = 1,
                       powershape.sd = 10)

fit = bwsp_model(datstan = standat,       # fit the model
                  tte.dist = "pgw",
                  prior.dist = "ll",
                  chains = 4,
                  iter = 110,          # (posterior sample
                  warmup = 10)         # is small for demo purpose)

# 3. HDI specification and extraction:
# extract 80% HDIs representing posterior samples of the shape parameters
post.samples = rstan::extract(fit, pars = c("nu", "gamma"))
nu.hdi = HDInterval::hdi(object = post.samples$nu, credMass = 0.8)
ga.hdi = HDInterval::hdi(object = post.samples$gamma, credMass = 0.8)

# 4. conduct the BWSP test
bwsp_test(credregion = c(nu.hdi, ga.hdi),
           nullregion = rope,
           tte.dist = "pgw",
           option = 1)

# returns a signal

```

eval.calc_perf

Calculate performance metrics for WSP test configurations in simulation

Description

Computes performance metrics for all the specified Bayesian and frequentist Weibull Shape Parameter (BWSP, BWSP) test configurations across simulated scenarios. The output provides the base for a ranking of tests (see [eval.rank_auc](#)).

Usage

```
eval.calc_perf(pc_list)
```

Arguments

pc_list	list of simulation parameters generated with sim.setup_sim_pars
---------	---

Details

Based the merged simulation results (obtained with `sim.merge_results`), the function performs WSP tests for all specified test setups - for Bayesian tests depending on the combination of posterior credibility interval (CI) type, credibility level and the sensitivity option (see `bwsp_test` and `pc_list$test`); for frequentist tests depending on the credibility level (see `fwsp_test` and `pc_list$input$cred.level`).

Given binary test results the function calculates for each subset representing one scenario and test combination the following performance measures:

- False positive rate:

$$fpr = \frac{FP}{FP + TN}$$

- True positive rate (sensitivity, recall):

$$tpr = \frac{TP}{TP + FN}$$

- False negative rate:

$$fnr = \frac{FN}{TP + FN}$$

- True negative rate (specificity):

$$tnr = \frac{TN}{FP + TN}$$

with FP being the number of false positive cases, TN the number of true negative cases, TP the number of true positive cases and FN the number of false negative cases among simulation repetitions, as well, as

- Area under the ROC curve:

The AUC is the area under the receiver operating characteristic (ROC) graph (Fawcett 2004). Here, we use the ROC curve with one threshold based on equal numbers of ADR-positive and control scenarios. It is computed using the `performance` function.

Value

A data frame containing one row per ADR-positive scenario, WSP model and test configuration, and corresponding performance measurements in additional columns, namely the `auc`, `fpr`, `tpr`, `fnr` and `tnr`.

Scenarios with incomplete amount of simulations return NA for performance metrics. FWSP tests return NA for scenario/model characteristics only relevant for BWSP test specification.

References

- Fawcett T (2004). “ROC graphs: Notes and practical considerations for researchers.” *Machine learning*, **31**(1), 1–38.

`eval.eff_sample_sizes` *Evaluate effective sample sizes by tte and prior distribution*

Description

Summarizes and visualizes effective sample sizes of the stan models fitted during the simulation study, grouped by time-to-event (tte) and prior distribution types. This helps assess which of the tte and prior distribution choices is suitable for HDI+ROPE testing (along with other diagnostics such as `eval.non_conv_cases` and `eval.execution_times`).

Usage

```
eval.eff_sample_sizes(pc_list, threshold = 10000)
```

Arguments

<code>pc_list</code>	a list containing simulation parameters (see <code>sim.setup_sim_pars</code>)
<code>threshold</code>	numeric threshold for effective sample size acceptable for HDI+ROPE testing (10000 by default as recommended by Kruschke (2015))

Value

A list with summary statistics (`$summary`), a `ggplot2` object (`$plot`), and the data (`$df`) on which summary and plot are based.

References

Kruschke J (2015). *Doing Bayesian Data Analysis (Second Edition)*. Academic Press, Boston.

See Also

`eval.non_conv_cases`, `eval.execution_times`

`eval.execution_times` *Evaluate execution times by tte and prior distribution*

Description

Summarizes and visualizes execution times of the models fitted during the simulation study grouped by time-to-event (tte) and prior distribution types to guide the tte and prior distributional choices (along with other diagnostics such as `eval.non_conv_cases` and `eval.eff_sample_sizes`).

Usage

```
eval.execution_times(pc_list)
```

Arguments

<code>pc_list</code>	list of simulation parameters generated with <code>sim.setup_sim_pars</code>
----------------------	--

Value

A list with summary statistics (\$summary), a ggplot2 object (\$plot), and the data (\$df) on which summary and plot are based.

See Also

[eval.non_conv_cases](#), [eval.eff_sample_sizes](#)

`eval.non_conv_cases` *Evaluate number of non-convergence cases by tte and prior distribution*

Description

Summarizes the number of planned vs. not successfully run simulations grouped by `tte.dist` and `prior.dist`. The purpose of the table is to guide the time-to-event (tte) and prior distributional choices (along with other diagnostics such as [eval.execution_times](#) and [eval.eff_sample_sizes](#)).

Usage

`eval.non_conv_cases(pc_list)`

Arguments

`pc_list` list of simulation parameters generated with [sim.setup_sim_pars](#)

Details

Calculations are based on the stored result batch files of the simulation study. Unsuccessfully run simulations are missing (supposedly due to non-convergence during model estimation).

Value

A data frame with the following columns:

- `tte.dist`: The tte distribution as grouping factor.
- `prior.dist`: The prior distribution as grouping factor.
- `total.planned`: The total number of planned repetitions in this group.
- `total.notrun`: The total number of repetitions that were not run in this group.
- `prop.notrun`: The proportion of repetitions that were not run in this group.

See Also

[eval.execution_times](#), [eval.eff_sample_sizes](#)

`eval.rank_auc`*Classification of WSP test configurations by AUC*

Description

Classifies all model and test specifications grouped by simulation scenarios in terms of the corresponding area under the curve (AUC) value for Weibull Shape Parameter (WSP) tests.

Usage

```
eval.rank_auc(
  perf,
  test.type.subset = c("bwsp", "fwsp"),
  tte.dist.subset = c("w", "dw", "pgw"),
  prior.dist.subset = c("fg", "f1", "gg", "ll")
)
```

Arguments

<code>perf</code>	data frame containing performance results for WSP tests returned by eval.calc_perf
<code>test.type.subset</code>	character to filter for Bayesian and frequentist WSP (BWSP, FWSP) test types to be considered in the ranking; must be a subset of <code>c("bwsp", "fwsp")</code>
<code>tte.dist.subset</code>	character to filter for the time-to-event (tte) distributions considered in the ranking, must be a subset of <code>c("w", "dw", "pgw")</code>
<code>prior.dist.subset</code>	character to filter for the prior distribution (relevant only for BWSP tests), must be a subset of <code>c("fg", "f1", "gg", "ll")</code>

Details

For definitions of the performance metrics AUC, FPR, TPR, FNR and TNR returned in output, see the details section of [eval.calc_perf](#).

The filter mechanism enables filtering for a subset of test specifications. This is helpful for example when tte distributions, prior distributions or an estimation approach are no longer under consideration after inspecting the model diagnostics with [eval.execution_times](#), [eval.non_conv_cases](#) and [eval.eff_sample_sizes](#), or when inspection of only one tte distribution was the objective from the beginning.

Value

A list containing

- `$rank.tab`: Ranking of fit and WSP test specifications according to AUC averaged over all sample scenarios (- for BWSP tests given a correct specification of prior belief)
- `$effect.of.N`: Effect of sample size on AUC for the optimal fit and WSP test (- for BWSP tests given a correct specification of prior belief)
- `$effect.of.br`: Effect of background rate on AUC for the optimal fit and WSP test (- for BWSP tests given a correct specification of prior belief)

- \$effect.of.adr.rate: Effect of ADR rate on AUC for the optimal fit and WSP test (- for BWSP tests given a correct specification of prior belief)
- \$effect.of.adr.when: Effect of true expected event times on AUC for the optimal fit and WSP test (- for BWSP tests given a correct specification of prior belief)
- \$effect.of.adr.relsd: Effect of relative standard deviation of event time on AUC for the optimal fit and WSP test (- for BWSP tests given a correct specification of prior belief)
- \$effect.of.dist.prior.to.truth: Effect of distance of prior belief to true adr.when on AUC for the optimal fit and WSP test (only BWSP)

Examples

```
## Not run:
# loading of performance metrics returned by eval.calc_perf function,
# called perf here
load(paste0(pc_list$add$resultpath, "/perf.RData"))

# ranking of all WSP tests considered in simulation setup
rank = eval.rank_auc(perf)

# ranking of subset of all Bayesian WSP tests considered in the simulation setup
rank_b = eval.rank_auc(perf, test.type.subset = "bwsp")

# ranking of subset of all frequentist Weibull tests considered
# in the simulation setup
rank_w = eval.rank_auc(perf, tte.dist.subset = "w")

# ranking of subset of all Bayesian WSP tests with "ll"-prior considered in
# the simulation setup
rank_b_ll = eval.rank_auc(perf, test.type.subset = "bwsp",
                           tte.dist.subset = "pgw",
                           prior.dist.subset = "ll")

# ranking of subset of all frequentist WSP tests with prior distribution
# specification -> leads to warning as prior dist specification has no effect
# on subset
rank_f = eval.rank_auc(perf, test.type.subset = "fwsp",
                           prior.dist.subset = "fg")

## End(Not run)
```

eval.roc_curve

Plot ROC curves for top WSP test specifications

Description

Plots receiver operating characteristic (ROC) curves for the top-ranked test specifications.

Usage

```
eval.roc_curve(rank.tab, n = 10)
```

Arguments

rank.tab	data frame of ranked test specifications obtained from <code>eval.rank_auc</code> (<code>output\$rank.tab</code>)
n	number of top-ranked test specifications to plot (10 by default)

Details

The function returns the receiver ROC curves for the top n WSP test configurations based on the ranking returned by `eval.rank_auc` by plotting the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis (Fawcett 2004). Here, we use the ROC curve with one threshold based on equal numbers of ADR-positive and control scenarios.

For definitions of the performance metrics AUC, FPR, TPR, FNR and TNR returned in printed output, see the details section of `eval.calc_perf`.

The ggplot output can be adjusted to individual needs by adding `ggplot2` layers to the output.

Value

A ggplot object displaying ROC curves with shaded AUC regions.

References

Fawcett T (2004). “ROC graphs: Notes and practical considerations for researchers.” *Machine learning*, **31**(1), 1–38.

Examples

```
## Not run:
# Given object rank as output of eval.rank_auc:
eval.roc_curve(rank$rank.tab, n = 1) # top 1 test
eval.roc_curve(rank$rank.tab, n = 10) # top 10 tests

# exemplary further processing of the plot
library(patchwork) # again, formatted next to each other
# Left plot: n=1, no legend, no title
p1 <- eval.roc_curve(rank$rank.tab, n = 1) +
  ggplot2::guides(color = "none", fill = "none") +
  ggplot2::labs(title = NULL)
Right plot: n=10, legend, no title
p2 <- eval.roc_curve(rank$rank.tab, n = 10) +
  ggplot2::labs(title = NULL)
# Combine with patchwork and add global title
(p1 | p2) + patchwork::plot_annotation(title = "ROC curves")

## End(Not run)
```

Description

Fits a frequentist model to time-to-event (tte) data via maximum likelihood (ML) estimation.

Usage

```
fwsp_model(dat, tte.dist = c("w", "dw", "pgw"))
```

Arguments

<code>dat</code>	data frame or matrix with time information in first column and event information (binary status) in second column
<code>tte.dist</code>	character specifying the distribution for the model; options are "w", "dw", "pgw" (see details)

Details

The model can be a Weibull ("w"), a double Weibull ("dw", estimating two Weibull models - one to the data as is and one to the data censored at mid of observation period), or a power generalized Weibull ("pgw") model.

The likelihood used in ML estimation is

$$\mathcal{L}(t) = \prod_{i=1}^N S(t_i)^{1-d_i} \cdot f(t_i)^{d_i}$$

with $S(t)$ the survival function of the chosen distribution and $f(t)$ the density function (Nikulin et al. 2016). The pair (t_i, d_i) are the tte observations.

Since the `survreg` function from the `survival` package uses a different parametrization, the parameters transformed to the parametrization used in `rweibull` and `rgpw` are printed after function call.

Value

Output of the fitted model. For "w", a `summary.survreg` object; for "dw", a list of two `summary.survreg` objects with `$uncens` containing the ML estimate obtained from the model to the tte data as is, and `$cens` containing the ML estimate obtained from the model fitted to the tte data censored at mid of observation period; for pgw, a list containing the maximum likelihood estimate (log of all parameters) obtained from `nls`.

References

Nikulin M, Wu HI, others (2016). *The Cox model and its applications*. Springer.

Examples

```
head(tte)
fwsp_model(tte, tte.dist = "w") # Weibull model
fwsp_model(tte, tte.dist = "dw") # double Weibull model
fwsp_model(tte, tte.dist = "pgw") # power generalized Weibull model
```

fwsp_test*Frequentist Shape Parameter Test*

Description

Frequentist hypothesis test based on the shape parameter(s) of a distribution of the Weibull family.

Usage

```
fwsp_test(
  mod.output,
  tte.dist = c("w", "dw", "pgw"),
  credlevel = 1 - c(1:10/1000, 2:10/100)
)
```

Arguments

mod.output	estimation output resulting from fwsp_model
tte.dist	character indicating the modelling approach; options are "w", "dw", "pgw"
credlevel	numeric or vector of credibility levels (i.e. 1 - significance level) for the test(s) to be performed

Details

This function tests the null hypothesis that the shape parameter(s) of the Weibull family distribution are equal to one. The distribution specific definitions of the null and alternative hypotheses can be seen in Sauzet and Cornelius (2022).

For the "w" and "dw" case, the model output is a summary of a `survival::Survreg` outcome which provides $\ln(1/\nu)$ as transform of the shape parameter estimate ν . The transform $\ln(1/\nu) = 0$ under the null hypothesis $\nu = 1$. The shape parameter test is performed on the transform equivalent to performing the test based on the shape parameter itself.

For the "pgw" case, the shape parameter test is performed on the logarithmized parameter estimates, i.e. `fwsp_test` tests the null hypothesis that the logarithm of the shape parameters of the power generalized Weibull distribution are equal to zero based on the shape estimates and their estimated standard errors extracted from the estimated Hessian matrix. Issues with standard error calculation from the estimated Hessian matrix may lead to NA test result which are then transformed to no signal (0) following Sauzet and Cornelius (2022).

Value

binary vector, 0 if H_0 is accepted, 1 if H_1 is rejected; see details for definition of H_0 and H_1

References

Sauzet O, Cornelius V (2022). “Generalised weibull model-based approaches to detect non-constant hazard to signal adverse drug reactions in longitudinal data.” *Frontiers in Pharmacology*. doi:[10.3389/fphar.2022.889088](https://doi.org/10.3389/fphar.2022.889088), <https://pubmed.ncbi.nlm.nih.gov/36081935/>.

Examples

```
# fit a model
mod = fwsp_model(dat = tte, tte.dist = "pgw")
mod
# perform the shape parameter test at credibility level 0.95
# or significance level 0.05
fwsp_test(mod.output = mod, tte.dist = "pgw", credlevel = 0.95)
```

gamprior_repar

Reparametrization of gamma mean and sd

Description

Can be used after specifying a gamma prior in terms of mean and standard deviation (sd) to calculate the shape and rate as defined in [qgamma](#).

Usage

```
gamprior_repar(mean, sd)
```

Arguments

mean	mean of the gamma distribution
sd	standard deviation of the gamma distribution

Details

Mean $E(X)$ and sd $sd(X)$ of a gamma distributed random variable X parametrized in terms of the shape parameter α and rate parameter β are given by

$$E(X) = \frac{\alpha}{\beta}$$

$$sd(X) = \sqrt{\frac{\alpha}{\beta^2}}$$

For $E(X), sd(X) > 0$ rearranging above equations to

$$\alpha = \frac{\mu^2}{\sigma^2}$$

$$\beta = \frac{\mu}{\sigma^2}$$

is possible such that the shape parameter α and rate parameter β of the gamma distribution are obtained.

The application purpose is to reparametrize gamma prior mean and sd to the parameters used in [qgamma](#), for instance to calculate a ROPE (Kruschke 2018) based on the prior belief representing the null hypothesis in BWSP testing (Dyck and Sauzet 2025).

Value

a vector with the shape and rate of the gamma distribution

References

Dyck J, Sauzet O (2025). “The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions.” preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.

Kruschke JK (2018). “Rejecting or Accepting Parameter Values in Bayesian Estimation.” *Advances in Methods and Practices in Psychological Science*, 1(2), 270-280. doi:[10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304), <https://doi.org/10.1177/2515245918771304>.

See Also

[logprior_repar](#)

Examples

```
# obtain shape and rate for gamma distribution with mean = 1 and sd = 10
m = 1; s = 10
gampars = gamprior_repar(mean = m, sd = s)
shape = gampars[1] # shape parameter
rate = gampars[2] # rate parameter

# test: sample from the gamma distribution with the obtained parameters
gamma_sample = rgamma(10000000, shape = shape, rate = rate)
m_emp = mean(gamma_sample); m_emp # estimated mean
s_emp = sd(gamma_sample); s_emp # estimated sd

# suppose, upper mean and sd reflect prior belief about the Weibull shape parameter and
# calculate an 80% ROPE based on the parameters
rope = qgamma(p = c(0.1, 0.9), shape = gampars[1], rate = gampars[2])
rope
```

[logprior_repar](#)

Reparametrization of lognormal mean and standard deviation

Description

Can be used after specifying a lognormal prior in terms of mean and standard deviation (sd) to calculate the location `meanlog` and scale `sdlog` as defined in [qlnorm](#).

Usage

`logprior_repar(mean, sd)`

Arguments

<code>mean</code>	mean of the lognormal distribution
<code>sd</code>	sd of the lognormal distribution

Details

Mean $E(X)$ and sd $sd(X)$ of a lognormal random variable X parametrized in terms of the location `meanlog` μ and scale `sdlog` σ are given by

$$E(X) = \exp\left(\mu + \frac{1}{2\sigma^2}\right)$$

$$sd(X) = \sqrt{(\exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1))}.$$

If $|sd(X)| > |E(X)| > 0$, rearranging above equations to

$$\mu = \log(E(X)) - \frac{1}{2} \log\left(\frac{sd(X)^2}{E(X)^2} - 1\right)$$

$$\sigma = \sqrt{\log\left(\frac{sd(X)^2}{E(X)^2} + 1\right)}.$$

is possible such that the location parameter `meanlog` and the scale parameter `logmean` of the log-normal distribution are obtained.

The application purpose is to reparametrize lognormal prior mean and sd to the parameters used in `qlnorm`, for instance to calculate a ROPE (Kruschke 2018) based on the prior belief representing the null hypothesis in BWSP testing (Dyck and Sauzet 2025).

Value

a vector with the location `meanlog` and scale `sdlog` of the lognormal distribution

References

Dyck J, Sauzet O (2025). “The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions.” preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.

Kruschke JK (2018). “Rejecting or Accepting Parameter Values in Bayesian Estimation.” *Advances in Methods and Practices in Psychological Science*, 1(2), 270-280. doi:[10.1177/2515245918771304](https://doi.org/10.1177/2515245918771304), <https://doi.org/10.1177/2515245918771304>.

See Also

[gamprior_repar](#)

Examples

```
# obtain location and scale for lognormal distribution with mean = 1 and sd = 10
m = 1; s = 10
logpars = logprior_repar(mean = m, sd = s)
mu = logpars[1] # location parameter
sigma = logpars[2] # scale parameter

# test: sample from the lognormal distribution with the obtained parameters
lnorm_sample = rlnorm(10000000, meanlog = mu, sdlog = sigma)
m_emp = mean(lnorm_sample); m_emp # estimated mean
s_emp = sd(lnorm_sample); s_emp # estimated sd
```

```
# suppose, upper mean and sd reflect prior belief about the Weibull shape parameter and
# calculate an 80% ROPE based on the parameters
rope = qlnorm(p = c(0.1,0.9), meanlog = logpars[1], sdlog = logpars[2])
rope
```

muscu

Simulated musculoskeletal pain time-to-event dataset

Description

A simulated time-to-event (tte) dataset representing musculoskeletal pain following bisphosphonate intake.

Usage

muscu

Format

A data frame with 19 777 rows and 2 variables:

time event time (in days) or censoring time (365 days),
status event indicator; 1 = event observed, 0 = censored.

Details

The data was generated using [sim.datagen_tte](#) with parameters derived from the case study presented in Dyck and Sauzet (2025).

References

Dyck J, Sauzet O (2025). “The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions.” preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.

See Also

[sim.datagen_tte](#)

pgw

The power generalized Weibull distribution

Description

Survival, hazard, cumulative distribution, density, quantile and sampling function for the power generalized Weibull (PgW) distribution with parameters scale, shape and powershape.

Usage

```
spgw(x, scale = 1, shape = 1, powershape = 1, log = FALSE)
hpgw(x, scale = 1, shape = 1, powershape = 1, log = FALSE)
ppgw(x, scale = 1, shape = 1, powershape = 1)
dpgw(x, scale = 1, shape = 1, powershape = 1, log = FALSE)
qpgw(p, scale = 1, shape = 1, powershape = 1)
rpgw(n, scale = 1, shape = 1, powershape = 1)
```

Arguments

x	vector of quantiles
scale	scale parameter
shape	shape parameter
powershape	power shape parameter
log	FALSE (default); if TRUE, the logarithm of the survival probability is returned
p	vector of probabilities
n	number of observations

Details

The survival function of the PgW distribution is:

$$S(x) = \exp \left\{ 1 - \left[1 + \left(\frac{x}{\theta} \right)^\nu \right]^{\frac{1}{\gamma}} \right\}.$$

The hazard function is

$$h(x) = \frac{\nu}{\gamma \theta^\nu} \cdot x^{\nu-1} \cdot \left[1 + \left(\frac{x}{\theta} \right)^\nu \right]^{\frac{1}{\gamma-1}}$$

The cumulative distribution function is then $F(x) = 1 - S(x)$ and the density function is $S(x) \cdot h(x)$. The quantile function is the inverse of the cumulative distribution function $F^{-1}(x)$.

If both shape parameters equal 1, the PgW distribution reduces to the exponential distribution (see [dexp](#)) with rate = 1/scale. If the power shape parameter equals 1, the PgW distribution simplifies to the Weibull distribution (see [dweibull](#)) with the same parametrization.

If parameter values are not specified, they are set as scale = 1, shape = 1, powershape = 1 per default.

Value

A vector of cumulative probability values evaluated at each of the inserted quantiles.

References

Nikulin M, Wu HI, others (2016). *The Cox model and its applications*. Springer.

plot_pgw

Visualize the power generalized Weibull distribution

Description

Visualize the power generalized Weibull distribution

Usage

```
plot_pgw(scale = 1, shape = 1, powershape = 1, ...)
```

Arguments

scale	scale parameter
shape	shape parameter
powershape	power shape parameter
...	optional plot arguments (with exceptions, eg. type, lwd, main)

Value

A plot of the PDF, CDF, hazard and survival function

See Also

[pgw](#)

An interactive version of this plot is available on <https://janoleko.shinyapps.io/pgwd/>.

Examples

```
plot_pgw(scale = 2, shape = 5, powershape = 10)
```

<code>sim.datagen_tte</code>	<i>Generate simulated time-to-event data</i>
------------------------------	--

Description

Simulation of time-to-event (tte) data.

Usage

```
sim.datagen_tte(genpar)
```

Arguments

<code>genpar</code>	A vector containing 6 numeric elements: <ol style="list-style-type: none"> 1. sample size N, 2. background rate br (observed in population on average), 3. adverse drug reaction (ADR) rate adr as proportion of the background rate, 4. relative proportion of ADR mean time $m.rel$ of the observation period (OP), 5. relative standard deviation $rel.sd$, 6. length of the OP $censor$.
---------------------	--

Details

After specification of the input, the data simulation works as follows:

The absolute number of events due to background causes (other than ADR) is generated by a binomial distribution with probability br . The absolute number of events caused by the ADR is generated with a binomial distribution with probability $br \cdot adr$. In sum, the expected number of events within the data set is $n \cdot br(1 + adr)$.

For the br cases, the event-times are generated using a uniform distribution on the interval $[0, censor]$. For the ADR cases, event-times are obtained from a normal distribution. The mean of the normal distribution is specified as relative proportion $m.rel$ of the OP. The standard deviation is defined as $rel.sd \cdot censor$. All generated event-times ≤ 0 (due to the normal distribution's support) are set to 1. All generated event-times $\geq censor$ (due to the normal distribution's support) resampled. The continuous values are rounded to integer. The data set is filled up with censored observations (status = 0) at time $censor$.

For more details, see Dyck and Sauzet (2025).

Value

A data frame of size N with variables `time` (integer) indicating event- or censoring time and `status` (binary) indicating whether the event was observed or the observation was censored.

References

Dyck J, Sauzet O (2025). “The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions.” preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.

See Also[tte](#)**Examples**

```
sim.datagen_tte(c(100, 0.1, 1, 0.5, 0.05, 365))
```

sim.merge_results *Merge result table batches from simulation study*

Description

Merges result table batches from simulation study obtained from using [sim.run](#) or [sim.run_parallel](#).

Usage

```
sim.merge_results(pc_list, save = T, bayes = T)
```

Arguments

pc_list	list of parameter combinations obtained from sim.setup_sim_pars
save	if TRUE (default), merged table is saved as res_b.RData or res_f.RData in same path where batches are stored; else, it is returned to global environment
bayes	TRUE (default), results of Bayesian simulations are merged, else, results of frequentist simulations are merged

Value

Dataframe containing all simulation results (one repetition of one simulation scenario per row). The simulation parameters are stored in the first 9 columns. The remaining columns contain

1. posterior summary statistics and percentiles (ie information on the posterior distribution) for each shape parameter and
2. posterior credibility intervals (as specified in \$test list element obtained from [sim.setup_sim_pars](#)).

sim.priors_template *Template for prior specification in simulation study*

Description

Generates a structured template for specifying prior means and prior standard deviations (sds) for the Weibull, double Weibull, or Power generalized Weibull (PgW) model parameters to be inserted into [sim.setup_sim_pars](#).

Usage

```
sim.priors_template(tte.dist = c("w", "dw", "pgw"), prior.sds = NULL)
```

Arguments

- `tte.dist` character vector specifying one or multiple modelling approaches; options are "w", "dw", "pgw" (see [bwsp_mode1](#))
- `prior.sds` numeric value setting the same prior sd for all scale and shape parameters across all included model types. Defaults to NULL, leaving all sd fields as NA.

Details

The returned list contains one data frame per time-to-event (tte) distribution (w, dw, pgw). For each chosen tte distribution, rows corresponding to different levels of prior belief about the hazard function are provided, namely "none", "beginning", "middle", and "end" (Dyck and Sauzet 2025). Given the template, prior means and sds must be filled by the user before simulation.

Value

A named list containing three data frames \$w, \$dw, and \$pgw. Each is a data frame specifying prior beliefs and placeholder entries for prior means and sds.

See Also

[sim.setup_sim_pars](#)

Examples

```
#### prior elicitation -----
# try a few prior parameter combinations and see whether the resulting hazard
# roughly matches the prior belief about the hazard form

# Expected event time can also be taken into account for some guidance, but
# should not be prioritized.
# The reason is that we do not expect the model to accurately fit the hazard of the
# data, but only catch the rough form by distinguishing the cases
# constant vs decreasing vs unimodal vs increasing hazard.

# set prior means for Power generalized Weibull parameters:
plot_pgw(scale = 1, shape = 1, powershape = 1)      # under prior belief "none"
plot_pgw(scale = 20, shape = 5.5, powershape = 14) # under prior belief "beginning"
plot_pgw(scale = 180, shape = 1, powershape = 1)    # under prior belief "middle"
plot_pgw(scale = 300, shape = 4, powershape = 1)    # under prior belief "end"

#### specify parameter combinations for simulation study -----
fp_list = sim.priors_template(tte.dist = c("pgw"),
                               prior.sds = 10) # setup prior template
# fill in prior template with values chosen in prior elicitation
fp_list$pgw[,2] = c(1, 1, 20, 300)   # scale prior means
fp_list$pgw$shape.mean_pgw <- c(1, 0.207, 5.5, 4) # shape prior means
fp_list$pgw[,6] = c(1, 1, 14, 1)     # powershape prior means

fp_list # filled fitpars.list ready for sim.setup_sim_pars()
```

sim.run*Run simulation***Description**

Runs simulations for all data generating processes, model and test alternatives specified in `pc_list`. If part of the simulations is already run and saved in the determined resultpath, only missing simulations are run and saved.

Usage

```
sim.run(pc_list, subset_ind = NULL)
```

Arguments

<code>pc_list</code>	list of parameter combinations generated with sim.setup_sim_pars
<code>subset_ind</code>	vector of integers specifying which rows of <code>pc_list\$pc_table</code> to be considered in simulation runs; defaults all rows

Details

This function runs the simulations in serial. To parallelise the execution, see [sim.run_parallel](#).

Examples

```
## Not run:
sim.run(pc_list, subset_ind = NULL) # run all simulations

# To run only a subset of simulation scenarios, specify the indices of the rows
# in pc_list$pc_table that you want to run. For example, to run the first 10 scenarios:
sim.run(pc_list, subset_ind = 1:10) # run first 10 simulation scenarios

## End(Not run)
```

sim.run_parallel*Run simulation (in parallel)***Description**

Runs simulations for all data generating processes, model and test alternatives specified in `pc_list`. If part of the simulations is already run and saved in the determined resultpath, only missing simulations are run and saved.

Usage

```
sim.run_parallel(pc_list, subset_ind = NULL)
```

Arguments

pc_list	list of parameter combinations obtained from <code>sim.setup_sim_pars</code>
subset_ind	vector of integers specifying which rows of pc_list\$pc_table to be considered in simulation runs

Details

Simulation runs can be parallelized using the `plan` command (see example).

Examples

```
## Not run:
# install.packages(future)
future::plan(multisession, workers = availableCores()) # or another plan strategy
sim.run_parallel(pc_list, subset_ind = NULL) # run all simulations

# To run only a subset of simulation scenarios, specify the indices of the rows
# in pc_list$pc_table that you want to run. For example, to run the first 10 scenarios:

# install.packages(future)
future::plan(multisession, workers = availableCores()) # or another plan strategy
sim.run_parallel(pc_list, subset_ind = 1:10) # run first 10 simulation scenarios

## End(Not run)
```

`sim.setup_sim_pars` *Set up simulation parameters*

Description

Sets up parameters for a simulation study to tune the Bayesian Weibull shape parameter (BWSP) test. Simulation parameters encompass data generating process (DGP) parameters (`N`, ..., `study.period`), tuning parameters for the BWSP test (`tte.dist`, ..., `sensitivity.option`), and additional parameters (`reps`, ..., `stanmod.warmup`).

Usage

```
sim.setup_sim_pars(
  N,
  br,
  adr.rate,
  adr.relsd,
  study.period,
  tte.dist,
  prior.dist,
  fitpars.list,
  post.ci.type,
```

```

  cred.level,
  sensitivity.option,
  reps = 100,
  batch.size = 10,
  resultpath = paste0(getwd(), "/results_raw"),
  stanmod.chains = 4,
  stanmod.iter = 11000,
  stanmod.warmup = 1000
)

```

Arguments

<code>N</code>	vector of sample sizes
<code>br</code>	vector of background rates (observed in population on average)
<code>adr.rate</code>	vector of adverse drug reaction rates as proportions of the background rates
<code>adr.relsd</code>	vector of relative standard deviations from the adverse drug reaction times
<code>study.period</code>	scalar specifying the length of the study period
<code>tte.dist</code>	character vector specifying one or multiple modelling approaches; options are "w", "dw", "pgw" (see bwsp_model)
<code>prior.dist</code>	character indicating the prior distribution for the parameters of the tte distribution; options are "fg", "fl", "gg", "ll" (see bwsp_model)
<code>fitpars.list</code>	list with one data.frame per tte distribution containing the prior specifications for model fitting; setup with sim.priors_template
<code>post.ci.type</code>	character indicating whether to extract equal tailed intervals ("ETI") or highest posterior density intervals ("HDI") as credibility interval (CI) for BWSP testing (see bwsp_test details)
<code>cred.level</code>	vector of credibility levels used for construction of region of practical equivalence (ROPE) and posterior CI
<code>sensitivity.option</code>	vector of sensitivity options for the BWSP test (see bwsp_test)
<code>reps</code>	number of repetitions for each simulation scenario, default is 100
<code>batch.size</code>	number of simulation repetitions to be saved in a batch (see details); default is 10
<code>resultpath</code>	directory where intermediate results of the simulation are saved
<code>stanmod.chains</code>	number of Markov chains (see sampling); default is 4
<code>stanmod.iter</code>	total number of iterations per chain including warmup (see sampling); default is 11000
<code>stanmod.warmup</code>	number of warmup (aka burnin) iterations per chain (see sampling); default is 1000

Details

The purpose of the simulation study is to evaluate the performance of different BWSP tests for data scenarios of interest following the tuning scheme developed in Dyck and Sauzet (2025).

DGP parameters (`N`, ..., `study.period`) should reflect the data characteristics of interest. Given the intention to apply the WSP test to a specific real data set, the DGP parameters should reflect its features. Within simulation, data are generated with [sim.datagen_tte](#).

Tuning parameters for the BWSP test (`tte.dist`, ..., `sensitivity.option`) lead to a range of tuning combinations evaluated during the simulation study to find the best test tuning. Among them, argument `fitpars.list` contains the prior means and sds for the prior distributions (`prior.dist`) for all scale and shape parameters. A template for the `fitpars.list` to be filled can be generated with `sim.priors_template`. Note, that the `tte.dist` argument in `sim.priors_template` and in `sim.setup_sim_pars` must match.

Additional parameters (`reps`, ..., `stanmod.warmup`) specify simulation settings and specifications for posterior sampling. Simulation settings encompass the number of repetitions per simulation scenario, the directory in which to save results and batch saving. Batch saving is done to prevent losing simulation results in case of an interruption of simulation e.g. due to termination of the R session. Posterior sampling specifications encompass the number of chains, iterations and the length of the warmup phase. See [sampling](#) for more details on the posterior estimation function.

Value

A list containing all simulation specifications in the format required for `sim.run`.

References

Dyck J, Sauzet O (2025). "The BPgWSP test: a Bayesian Weibull Shape Parameter signal detection test for adverse drug reactions." preprint, 2412.05463, <https://arxiv.org/abs/2412.05463>.

Examples

```
batch.size = 10,
resultpath = paste0(getwd(),"/simulation_results"),
stanmod.iter = 11000,
stanmod.warmup = 1000
)
```

`pc_list`

tte	<i>test time-to-event dataset</i>
------------	-----------------------------------

Description

Simulated time-to-event (tte) data generated with `sim.datagen_tte(c(100, 0.1, 1, 0.5, 0.05, 365))`.

Usage

`tte`

Format

A data frame with 100 rows and 2 variables:

time event-time if an event was observed, or censoring time if no event was observed,

status event status; 1 if an event was observed, 0 if no event was observed.

See Also

[sim.datagen_tte](#)

tte2priordat	<i>Prior and data setup for Bayesian survival model fitting</i>
---------------------	---

Description

Prepare time-to-event data and prior specifications for model fitting with [bwsp_model](#).

Usage

```
tte2priordat(
  dat,
  tte.dist = c("w", "dw", "pgw"),
  scale.mean,
  scale.sd,
  shape.mean,
  shape.sd,
  scale_c.mean = NULL,
  scale_c.sd = NULL,
  shape_c.mean = NULL,
  shape_c.sd = NULL,
  powershape.mean = NULL,
  powershape.sd = NULL
)
```

Arguments

dat	matrix or data frame with time in the first column and event status in the second column
tte.dist	character indicating the modelling approach; options are "w", "dw", "pgw"
scale.mean	prior mean of the scale parameter
scale.sd	prior standard deviation (sd) of the scale parameter
shape.mean	prior mean of the shape parameter
shape.sd	prior sd of the shape parameter
scale_c.mean	prior mean of the scale parameter for censored-at-half data (only for tte.dist="dw")
scale_c.sd	prior sd of the scale parameter for censored-at-half data (only for tte.dist="dw")
shape_c.mean	prior mean of the shape parameter for censored-at-half data (only for tte.dist="dw")
shape_c.sd	prior sd of the shape parameter for censored-at-half data (only for tte.dist="dw")
powershape.mean	prior mean of the power shape parameter (only for tte.dist="pgw")
powershape.sd	prior sd of the power shape parameter (only for tte.dist="pgw")

Details

Only the parameters relevant to the chosen tte.dist must be provided, that is:

- for "w": scale.mean, scale.sd, shape.mean, shape.sd
- for "dw": scale.mean, scale.sd, shape.mean, shape.sd, scale_c.mean, scale_c.sd, shape_c.mean, shape_c.sd
- for "pgw": scale.mean, scale.sd, shape.mean, shape.sd, powershape.mean, powershape.sd

Prior means suitable to reflect the prior belief can be worked out by plotting the hazard and estimating the expected event time under different parameter combinations using `plot_pgw`(powershape = 1 reduces the power generalized Weibull distribution to Weibull) or <https://janoleko.shinyapps.io/pgwd/>.

Prior standard deviations should reflect the uncertainty about the prior belief (i.e. set smaller standard deviation in case of high certainty about prior belief vs. larger standard deviation in case of low certainty).

Value

A named list in the format expected by [bwsp_model](#).

Examples

```
tte2priordat(dat = tte, tte.dist = "w", scale.mean = 10, scale.sd = 2, shape.mean = 1.5, shape.sd = 15)
tte2priordat(dat = tte, tte.dist = "dw", scale.mean = 10, scale.sd = 2, shape.mean = 1.5, shape.sd = 15,
             scale_c.mean = 5, scale_c.sd = 1, shape_c.mean = 1, shape_c.sd = 10)
tte2priordat(dat = tte, tte.dist = "pgw", scale.mean = 10, scale.sd = 2, shape.mean = 1.5, shape.sd = 15,
             powershape.mean = 3, powershape.sd = 20)
```

Index

* datasets
 muscu, 20
 tte, 30

bwsp_model, 3, 25, 28, 30, 32
bwsp_test, 3, 4, 9, 28

dexp, 21
dpgw (pgw), 21
dweibull, 21

eval.calc_perf, 8, 12, 14
eval.eff_sample_sizes, 10, 10, 11, 12
eval.execution_times, 10, 10, 11, 12
eval.non_conv_cases, 10, 11, 11, 12
eval.rank_auc, 8, 12, 14
eval.roc_curve, 13

fwsp_model, 14, 16
fwsp_test, 9, 16

gamprior_repar, 17, 19

hpgw (pgw), 21

logprior_repar, 18, 18

muscu, 20

nlm, 15

performance, 9
pgw, 21, 22
plan, 27
plot_pgw, 22, 31
ppgw (pgw), 21

qgamma, 17
qlnorm, 18, 19
qpgw (pgw), 21

rpgw, 15
rpgw (pgw), 21
rweibull, 15

sampling, 3, 28, 29

sim.datagen_tte, 20, 23, 28, 30
sim.merge_results, 9, 24
sim.priors_template, 24, 28, 29
sim.run, 24, 26, 29
sim.run_parallel, 24, 26, 26
sim.setup_sim_pars, 8, 10, 11, 24–27, 27, 29
spgw (pgw), 21

tte, 24, 30
tte2priordat, 3, 30

WSPsignal (WSPsignal-package), 2
WSPsignal-package, 2