Package ‘Temporal’

September 23, 2023

Title Parametric Time to Event Analysis

Version 0.3.0.1

Description Performs maximum likelihood based estimation and inference on time to event data, possibly subject to non-informative right censoring. FitParaSurv() provides maximum likelihood estimates of model parameters and distributional characteristics, including the mean, median, variance, and restricted mean. CompParaSurv() compares the mean, median, and restricted mean survival experiences of two treatment groups. Candidate distributions include the exponential, gamma, generalized gamma, log-normal, and Weibull.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

Imports dplyr, expint, methods, numDeriv, stats, tidyr

RoxygenNote 7.2.3

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Zachary McCaw [aut, cre] (<https://orcid.org/0000-0002-2006-9828>)

Maintainer Zachary McCaw <zmccaw@alumni.harvard.edu>

Repository CRAN

Date/Publication 2023-09-23 17:40:02 UTC

R topics documented:

`CheckArm` *Check Arm*

Description

Check whether treatment arm is properly formatted.

Usage

```
CheckArm(arm)
```

Arguments

`arm`	0/1, treatment arm.

Value

None.

`CheckDist` *Check Distribution*

Description

Check whether the distribution selected is available.

Usage

```
CheckDist(dist)
```

Arguments

`dist`	String, distribution name.

Value

None.

`CheckInit` *Check Initialization*

Description

Check whether the initialization is valid.

Usage

```
CheckInit(dist, init)
```

Arguments

`dist`	String, distribution name.
`init`	List of named parameters.

Value

None.

`CheckStatus` *Status Check*

Description

Function to ensure the status indicator is properly formatted

Usage

```
CheckStatus(status)
```

Arguments

`status`	0/1 status indicator.

Value

None.

CheckTheta *Check Theta*

Description

Function to check the appropriate number of parameters are supplied for the selected distribution. Used by GenData.

Usage

```
CheckTheta(dist, theta)
```

Arguments

dist	String, distribution.
theta	Numeric, parameter vector.

Value

None.

CompParaSurv *Compare Parametric Survival Distribution*

Description

Compares the means and medians of parametric survival distributions fit to two treatment arms. Available distributions include: exponential, gamma, generalized gamma, log-normal, and Weibull.

Usage

```
CompParaSurv(
  data,
  arm_name = "arm",
  dist1 = "weibull",
  dist0 = NULL,
  eps = 1e-06,
  init1 = NULL,
  init0 = NULL,
  maxit = 10,
  report = FALSE,
  reps = NULL,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

`data` Data.frame.

`arm_name` Name of the column containing the treatment group, coded as 1 for treatment, 0 for reference.

`dist1` Distribution to fit for the target group. Selected from among: exp, gamma, gengamma, log-normal, and weibull.

`dist0` Distribution to fit for the reference group. Same choices as for the target group. If omitted, defaults to the distribution specified for the target group.

`eps` Tolerance for Newton-Raphson iterations.

`init1` Initial parameter values for the target group.

`init0` Initial parameter values for the reference group.

`maxit` Maximum number of Newton-Raphson iterations.

`report` Report fitting progress?

`reps` Number of permutation replicates, if requesting permutation p-values.

`sig` Significance level, for constructing confidence intervals.

`status_name` Name of the status indicator, 1 if observed, 0 if censored.

`tau` Optional truncation times for calculating RMST.

`time_name` Name of column containing the time to event.

Details

Status should be coded as 0 for censored and 1 for observed. Arm is coded as 0 for reference, 1 for target. Tau is an optional numeric vector of truncation times for calculating restricted mean survival time, which is the area under the survival curve up to the specified truncation point.

Value

An object of class `contrast` containing the following:

Model1 The fitted model for the target group.

Model0 The fitted model for the reference group.

Contrast Contrasts of means and medians.

RMST Contrasts of the RMSTs, if ‘tau‘ was specified.

Examples

```
set.seed(100)
# Weibull and Weibull, different means and medians.
n <- 1e3

# Generate data.
df1 <- GenData(n = n, dist = "weibull", theta = c(1, 1), p = 0.2)
df1$arm <- 1
df0 <- GenData(n = n, dist = "weibull", theta = c(1, 2), p = 0.2)
```

```
df0$arm <- 0
data <- rbind(df1, df0)

# Comparison.
comp <- CompParaSurv(data, dist1 = "weibull")

# Add RMST at time 1.
comp <- CompParaSurv(data, dist1 = "weibull", tau = 1)

# Calculate permutation p-values (slow).
comp <-  CompParaSurv(data, dist1 = "weibull", tau = 1, reps = 100)
```

contrast-class *Contrast of Survival Distributions.*

Description

Defines the object class returned by the comparison function.

Slots

`Dist1` Distribution fit to the target group, string.

`Dist0` Distribution fit to the reference group, string.

`Model1` Fitted model for the target group, fit.

`Model0` Fitted model for the reference group, fit.

`Location` Contrasts of means and medians, data.frame.

`RMST` Contrasts of RMSTs, data.frame.

ContrastLocs *Contrast Locations*

Description

Compare the means and medians of the fitted distributions for two treatment arms.

Usage

```
ContrastLocs(fit1, fit0, sig = 0.05)
```

Arguments

`fit1`	Fitted parametric survival distribution for arm 1.
`fit0`	Fitted parametric survival distribution for arm 0.
`sig`	Significance level.

Value

Data.frame contrasting the difference and ratio of the mean and median at each time point.

ContrastRMSTs *Contrast RMSTs*

Description

Compare the restricted mean survival times of the fitted distributions for two treatment arms.

Usage

```
ContrastRMSTs(fit1, fit0, sig = 0.05)
```

Arguments

`fit1`	Fitted parametric survival distribution for arm 1.
`fit0`	Fitted parametric survival distribution for arm 0.
`sig`	Significance level, for

Value

Data.frame contrasting the difference and ratio of RMSTs at each time point.

DefaultParam *Set Default Parameters*

Description

Function to select default parameter values for each distribution.

Usage

```
DefaultParam(dist)
```

Arguments

`dist`	String, distribution name.

Value

Numeric parameter ist.

DistProperName *Distributions*

Description

Distributions

Usage

```
DistProperName(dist)
```

Arguments

dist	Argument passed to FitParaSurv

Value

String.

EstDiff *Difference of Estimates*

Description

Calculate CIs and p-value for the difference of estimates.

Usage

```
EstDiff(est1, se1, est0, se0, sig = 0.05)
```

Arguments

est1	Arm 1 estimate.
se1	Arm 1 standard error.
est0	Arm 0 estimate.
se0	Arm 0 standard error.
sig	Significance level.

Value

Data.frame containing estimated difference, its standard error, lower and upper confidence bounds, and a p-value assessing the null hypothesis of no difference.

`EstRatio` *Ratio of Estimates*

Description

Calculate CIs and p-value for the ratio of estimates.

Usage

```
EstRatio(est1, se1, est0, se0, sig = 0.05)
```

Arguments

`est1`	Arm 1 estimate.
`se1`	Arm 1 standard error.
`est0`	Arm 0 estimate.
`se0`	Arm 0 standard error.
`sig`	Significance level.

Value

Data.frame containing estimated ratio, its standard error, lower and upper confidence bounds, and a p-value assessing the null hypothesis that the ratio is unity.

`ExtractObsEst` *Extract Observed Estimates*

Description

Helper function for permutation inference.

Usage

```
ExtractObsEst(fit1, fit0)
```

Arguments

`fit1`	Fitted parametric survival distribution for arm 1.
`fit0`	Fitted parametric survival distribution for arm 0.

Value

Numeric vector.

fit-class *Fitted Survival Distribution*

Description

Defines the object class returned by fitting functions.

Slots

`Distribution` Fitted distribution, string.

`Parameters` Parameters, data.frame.

`Information` Information components, matrix.

`Outcome` Properties of the fitted distribution, data.frame.

`RMST` Estimated restricted mean survival times, data.frame

`S` Fitted survival function, function.

FitExp *Exponential Distribution Parameter Estimation*

Description

Estimates parameters for exponential event times subject to non-informative right censoring. The exponential distribution is parameterized in terms of the rate λ:

$$f(t) = \lambda e^{-\lambda t}, t > 0$$

Usage

```
FitExp(
  data,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

`data`	Data.frame.
`sig`	Significance level, for CIs.
`status_name`	Name of the status indicator, 1 if observed, 0 if censored.
`tau`	Optional truncation times for calculating RMSTs.
`time_name`	Name of column containing the time to event.

Value

An object of class `fit` containing the following:

Parameters The estimated model parameters.

Information The observed information matrix.

Outcome The fitted mean, median, and variance of the time to event distribution.

RMST The estimated RMSTs, if tau was specified.

Examples

```
# Generate exponential event time data with 20% censoring.
data <- GenData(n = 1e3, dist = "exp", theta = c(2), p = 0.2)

# Estimate parameters.
fit <- FitParaSurv(data, dist = "exp")
```

`FitGamma` *Gamma Distribution Parameter Estimation*

Description

Estimates parameters for gamma event times subject to non-informative right censoring. The gamma distribution is parameterized in terms of the shape α and rate λ:

$$f(t) = \frac{\lambda}{\Gamma(\alpha)}(\lambda t)^{\alpha-1}e^{-\lambda t}, t > 0$$

Usage

```
FitGamma(
  data,
  eps = 1e-06,
  init = list(),
  maxit = 10,
  report = FALSE,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

`data`	Data.frame.
`eps`	Tolerance for Newton-Raphson iterations.
`init`	List with initial values for the ‘shape‘ α and ‘rate‘ λ.

`maxit` Maximum number of NR iterations.

`report` Report fitting progress?

`sig` Significance level, for CIs.

`status_name` Name of the status indicator, 1 if observed, 0 if censored.

`tau` Optional truncation times for calculating RMSTs.

`time_name` Name of column containing the time to event.

Value

An object of class `fit` containing the following:

Parameters The estimated shape α and rate λ.

Information The observed information matrix.

Outcome The fitted mean, median, and variance.

RMST The estimated RMSTs, if tau was specified.

Examples

```
# Generate Gamma data with 20% censoring.
data <- GenData(n = 1e3, dist = "gamma", theta = c(2, 2), p = 0.2)

# Estimate parameters.
fit <- FitParaSurv(data, dist = "gamma")
```

`FitGammaComplete` *Gamma Parameter Estimation without Censoring*

Description

Paramter estimation for gamma event times without censoring.

Usage

```
FitGammaComplete(data, eps = 1e-06)
```

Arguments

`data` Data.frame.

`eps` Tolerance for Newton-Raphson iterations.

Value

Numeric vector containing the estimated shape and rate parameters.

FitGenGamma *Generalized Gamma Distribution Parameter Estimation*

Description

Estimates parameters for generalized gamma event times subject to non-informative right censoring. The gamma distribution is parameterized in terms of the shape parameters (α, β), and the rate λ:

$$f(t) = \frac{\beta\lambda}{\Gamma(\alpha)}(\lambda t)^{\alpha\beta-1}e^{-(\lambda t)^{\beta}}, t > 0$$

Usage

```
FitGenGamma(
  data,
  beta_lower = 0.1,
  beta_upper = 10,
  eps = 1e-06,
  init = list(),
  maxit = 10,
  report = FALSE,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

data	Data.frame.
beta_lower	If dist="gen-gamma", lower limit on possible values for beta.
beta_upper	If dist="gen-gamma", upper limit on possible values for beta.
eps	Tolerance for Newton-Raphson iterations.
init	List with initial values for the shape ‘alpha‘, ‘beta‘ and rate ‘lambda‘ parameters.
maxit	Maximum number of NR iterations.
report	Report fitting progress?
sig	Significance level, for CIs.
status_name	Name of the status indicator, 1 if observed, 0 if censored.
tau	Optional truncation times for calculating RMSTs.
time_name	Name of column containing the time to event.

Value

An object of class `fit` containing the following:

Parameters The estimated shape (α, β) and rate λ parameters.

Information The observed information matrix.

Outcome The fitted mean, median, and variance.

RMST The estimated RMSTs, if tau was specified.

Examples

```
set.seed(103)
# Generate generalized gamma data with 20% censoring.
data <- GenData(n = 1e4, dist = "gen-gamma", theta = c(2, 2, 2), p = 0.2)

# Estimate parameters.
fit <- FitParaSurv(data, dist = "gen-gamma", report = TRUE)
```

`FitGenGammaComplete` *Generalized Gamma Parameter Estimation without Censoring*

Description

Paramter estimation for generalized gamma event times without censoring.

Usage

```
FitGenGammaComplete(data, beta_lower = 0.1, beta_upper = 10)
```

Arguments

`data`	Data.frame.
`beta_lower`	Lower limit on possible values for beta.
`beta_upper`	Upper limit on possible values for beta.

Value

Numeric vector containing the estimated shape and rate parameters.

FitLogNormal *Log-Normal Distribution Parameter Estimation*

Description

Estimates parameters for log-normal event times subject to non-informative right censoring. The log-normal distribution is parameterized in terms of the location μ and scale σ:

$$f(t) = \phi\left(\frac{\ln t - \mu}{\sigma}\right)\frac{1}{t\sigma}, t > 0$$

Usage

```
FitLogNormal(
  data,
  eps = 1e-06,
  init = list(),
  maxit = 10,
  report = FALSE,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

`data`	Data.frame.
`eps`	Tolerance for Newton-Raphson iterations.
`init`	List with initial values for the location (‘loc‘) μ and ‘scale‘ σ.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?
`sig`	Significance level, for CIs.
`status_name`	Name of the status indicator, 1 if observed, 0 if censored.
`tau`	Optional truncation times for calculating RMSTs.
`time_name`	Name of column containing the time to event.

Value

An object of class `fit` containing the following:

Parameters The estimated location μ and scale σ.

Information The observed information matrix.

Outcome The fitted mean, median, and variance.

RMST The estimated RMSTs, if tau was specified.

Examples

```
# Generate log-normal data with 20% censoring.
data <- GenData(n = 1e3, dist = "log-normal", theta = c(0, 2), p = 0.2)

# Estimate parameters.
fit <- FitParaSurv(data, dist = "log-normal")
```

FitLogNormComplete *Log-Normal Parameter Estimation without Censoring*

Description

Log-Normal Parameter Estimation without Censoring

Usage

```
FitLogNormComplete(data)
```

Arguments

data	Data.frame.

Value

Numeric vector containing the estimate location and scale parameters.

FitParaSurv *Fit Parametric Survival Distribution*

Description

Estimates parametric survival distributions using event times subject to non-informative right censoring. Available distributions include: exponential, gamma, generalized gamma, log-normal, and Weibull.

Usage

```
FitParaSurv(
  data,
  beta_lower = 0.1,
  beta_upper = 10,
  dist = "weibull",
  eps = 1e-06,
  init = NULL,
  maxit = 10,
  report = FALSE,
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

data Data.frame containing the time to event and status.

beta_lower If dist="gen-gamma", lower limit on possible values for beta.

beta_upper If dist="gen-gamma", upper limit on possible values for beta.

dist String, distribution to fit, selected from among: exp, gamma, gen-gamma log-normal, and weibull.

eps Tolerance for Newton-Raphson iterations.

init List of initial parameters. See individual distributions for the expected parameters.

maxit Maximum number of NR iterations.

report Report fitting progress?

sig Significance level, for CIs.

status_name Name of the status indicator, 1 if observed, 0 if censored.

tau Optional truncation time for calculating RMSTs.

time_name Name of column containing the time to event.

Value

An object of class fit containing the following:

Parameters The estimated shape and rate parameters.

Information The observed information matrix.

Outcome The fitted mean, median, and variance.

RMST The estimated RMSTs, if tau was specified.

See Also

- Between group comparison of survival experience CompParaSurv
- Exponential distribution FitExp
- Gamma distribution FitGamma
- Generalized gamma distribution FitGenGamma
- Log-normal distribution FitLogNormal
- Weibull distribution FitWeibull

Examples

```
# Generate Gamma data with 20% censoring.
data <- GenData(n = 1e3, dist = "gamma", theta = c(2, 2), p = 0.2)
# Fit gamma distribution.
fit <- FitParaSurv(data, dist = "gamma")

# Generate Weibull data with 10% censoring.
data <- GenData(n = 1e3, dist = "weibull", theta = c(2, 2), p = 0.1)
# Fit weibull distribution, calculate RMST at tau=0.5.
fit <- FitParaSurv(data, dist = "weibull", tau = 0.5)
```

`FitWeibull` *Weibull Distribution Parameter Estimation*

Description

Estimates parameters for Weibull event times subject to non-informative right censoring. The Weibull distribution is parameterized in terms of the shape α and rate λ:

$$f(t) = \alpha\lambda^{\alpha}t^{\alpha-1}e^{-(\lambda t)^{\alpha}}, t > 0$$

Usage

```
FitWeibull(
  data,
  init = list(),
  sig = 0.05,
  status_name = "status",
  tau = NULL,
  time_name = "time"
)
```

Arguments

`data`	Data.frame.
`init`	List containing the initial value for the shape, α.
`sig`	Significance level, for CIs.
`status_name`	Name of the status indicator, 1 if observed, 0 if censored.
`tau`	Optional truncation times for calculating RMSTs.
`time_name`	Name of column containing the time to event.

Value

An object of class `fit` containing the following:

Parameters The estimated shape α and rate λ.

Information The observed information matrix.

Outcome The fitted mean, median, and variance.

RMST The estimated RMSTs, if tau was specified.

Examples

```
# Generate Weibull data with 20% censoring.
data <- GenData(n = 1e3, dist = "weibull", theta = c(2, 2), p = 0.2)

# Estimate parameters.
fit <- FitParaSurv(data, dist = "weibull")
```

GammaInfo *Gamma Observed Information*

Description

Observed information for gamme event times without censoring.

Usage

```
GammaInfo(data, shape, rate)
```

Arguments

`data`	Data.frame.
`shape`	Shape parameter α.
`rate`	Rate parameter λ.

Value

Numeric information matrix.

GammaScore *Gamma Profile Score for Shape*

Description

Profile score equation for gamma event times without censoring.

Usage

```
GammaScore(data, shape)
```

Arguments

`data`	Data.frame.
`shape`	Shape parameter.

Value

Numeric score.

GenData *Data Generation with Censoring*

Description

Generates data from survival distributions as parameterized in this package, with optional non-informative random right censoring.

Usage

```
GenData(n, dist = "exp", theta = NULL, p = 0)
```

Arguments

n	Integer sample size.
dist	String, distribution name selected from among: "exp","gamma","gen-gamma","log-normal","weibull".
theta	Numeric parameter vector. Elements will vary according to the distribution.
p	Expected censoring proportion.

Details

The parameter vector theta should contain the following elements, in order, depending on the distribution:

Exponential Rate λ.

Gamma Shape α, rate λ.

Generalized Gamma Shape 1 α, shape 2 β, rate λ.

Log-Normal Locaion μ, scale σ.

Weibull Shape α, rate λ.

Value

Data.frame including the observation times and status.

Examples

```
# Gamma event times with shape 2 and rate 2.
# Expected censoring proportion of 20%.
data <- GenData(n = 1e3, dist = "gamma", theta = c(2, 2), p = 0.20)

# Generalized gamma event times with shapes (2,3) and rate 1.
# Expected censoring proportion of 15%.
data <- GenData(n = 1e3, dist = "gen-gamma", theta = c(2, 3, 1), p = 0.15)

# Log-normal event times with location 0 and rate 1.
# Expected censoring proportion of 10%.
```

```
data <- GenData(n = 1e3, dist = "log-normal", theta = c(0, 1), p = 0.10)

# Weibull event times with shape 2 and rate 2.
# Expected censoring proportion of 5%.
data <- GenData(n = 1e3, dist = "weibull", theta = c(2, 2), p = 0.05)
```

GenGammaObsInfo *Generalized Gamma Observed Information*

Description

Observed information for the generalized gamma log likelihood in the absence of censoring.

Usage

```
GenGammaObsInfo(data, alpha, beta, lambda)
```

Arguments

`data`	Data.frame.
`alpha`	First shape parameter.
`beta`	Second shape parameter.
`lambda`	Rate parameter.

Value

Numeric observed information matrix.

GenGammaProfileLogLik *Generalized Gamma Profile Log Likelihood*

Description

Profile log likelihood of the generalized gamma distribution as a function of the second shape parameter β.

Usage

```
GenGammaProfileLogLik(data, beta)
```

Arguments

`data`	Data.frame.
`beta`	Second shape parameter.

Value

Numeric profile log likelihood.

`GenGammaRate` *Generalized Gamma Rate MLE*

Description

Profile MLE of the generalized gamma rate given the shape parameters.

Usage

```
GenGammaRate(data, alpha, beta)
```

Arguments

`data`	Data.frame.
`alpha`	First shape parameter.
`beta`	Second shape parameter.

Value

Numeric MLE of the rate λ.

`GenGammaScore` *Generalized Gamma Score Equation*

Description

Score equation for the generalized gamma log likelihood in the absence of censoring.

Usage

```
GenGammaScore(data, alpha, beta, lambda)
```

Arguments

`data`	Data.frame.
`alpha`	First shape parameter.
`beta`	Second shape parameter.
`lambda`	Rate parameter.

Value

Numeric score vector.

`GenGammaShape` *Generalized Gamma Shape MLE*

Description

Profile MLE of the first shape parameter α of the generalized gamma given the second shape parameter β.

Usage

```
GenGammaShape(data, beta)
```

Arguments

`data`	Data.frame.
`beta`	Second shape parameter.

Value

Numeric MLE of the rate α.

`LogNormInfo` *Log-Normal Observed Information*

Description

Observed information for log-normal event times without censoring.

Usage

```
LogNormInfo(data, loc, scale, log_scale = FALSE)
```

Arguments

`data`	Data.frame.
`loc`	Location parameter.
`scale`	Scale parameter.
`log_scale`	Is the scale parameter logged?

Value

Numeric score.

`LogNormScore` *Log-Normal Score Equation*

Description

Score equation for log-normal event times without censoring.

Usage

```
LogNormScore(data, loc, scale)
```

Arguments

`data`	Data.frame.
`loc`	Location parameter.
`scale`	Scale parameter.

Value

Numeric score.

`NewtonRaphson` *Newton Raphson Estimation*

Description

Newton Raphson Estimation

Usage

```
NewtonRaphson(init, obj, eps = 1e-06, maxit = 10, report = FALSE)
```

Arguments

`init`	Initial value.
`obj`	Objective function.
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

Value

Numeric parameter estimate.

NRUpdate *Newton Raphson Update Iteration*

Description

Newton Raphson Update Iteration

Usage

```
NRUpdate(obj, state)
```

Arguments

obj	Objective function.
state	List containing the parameter vector ‘theta‘.

Value

List containing the updated parameter vector ‘theta‘ and the objective increment ‘delta‘.

ParaRMST *Restricted Mean Survival Time*

Description

Calculates the RMST as the area under a fitted parametric survival distribution.

Usage

```
ParaRMST(fit, tau, sig = 0.05)
```

Arguments

fit	Fitted parametric survival distribution.
tau	Numeric vector of truncation times.
sig	Significance level, for CIs.

Value

Data.frame containing the estimated RMST at each truncation time.

Examples

```
# Generate Weibull data with 20% censoring.
data <- GenData(n = 1e3, dist = "weibull", theta = c(2, 0.5), p = 0.2)

# Fit Weibull distribution.
fit <- FitParaSurv(data, dist = "weibull")

# Calculate RMSTs.
rmst <- ParaRMST(fit = fit, tau = c(0.5, 1.0, 1.5, 2.0))

# Generate gamma data with 10% censoring.
data <- GenData(n = 1e3, dist = "gamma", theta = c(2, 2), p = 0.10)

# Fit gamma distribution.
fit <- FitParaSurv(data, dist = "gamma")

# Calculate RMSTs.
rmst <- ParaRMST(fit = fit, tau = c(0.5, 1.0, 1.5, 2.0))
```

PermP *Permutation P Value*

Description

Calculates permutation p-values for location and RMST estimates.

Usage

```
PermP(df1, df0, fit0, fit1, eps, init1, init0, maxit, reps, tau)
```

Arguments

df1	Target data.frame containing time and status.
df0	Reference data.frame containing time and status.
fit0	Fitted parametric survival distribution for arm 0.
fit1	Fitted parametric survival distribution for arm 1.
eps	Tolerance for Newton-Raphson iterations.
init1	Initial parameter values for the target group.
init0	Initial parameter values for the reference group.
maxit	Maximum number of Newton-Raphson iterations.
reps	Number of permutation replicates.
tau	Optional truncation times for calculating RMST.

Value

Numeric vector of permutation p-values.

`print.contrast` *Print Method for a Contrast of Survival Distributions.*

Description

Print method for an object of class `contrast`.

Usage

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

Arguments

`x`	A `contrast` object.
`...`	Unused.

`print.fit` *Print Method for Fitted Survival Distributions*

Description

Print method for objects of class `fit`.

Usage

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

Arguments

`x`	An object of class `fit`.
`...`	Unused.

QF *Quadratic Form*

Description

Quadratic Form

Usage

```
QF(x, A)
```

Arguments

x	Numeric vector.
A	Numeric matrix.

Value

Numeric scalar.

qWeibull *Quantile Function for the Weibull Distribution*

Description

Quantile function for the Weibull distribution. See FitWeibull for the parameterization.

Usage

```
qWeibull(p, a = 1, l = 1)
```

Arguments

p	Probability.
a	Shape.
l	Rate.

Value

Scalar quantile.

rGamma *Simulation from the Gamma Distribution*

Description

Generates gamma event times with shape parameter α and rate parameter λ. See FitGamma for the parameterization. If a censoring proportion p is provided, the event times are subject to non-informative random right censoring.

Usage

```
rGamma(n, a = 1, l = 1, p = 0)
```

Arguments

n	Sample size.
a	Shape.
l	Rate.
p	Expected censoring proportion.

Value

Data.frame including the observation times and status.

rGenGamma *Simulation from the Generalized Gamma Distribution*

Description

Generates generalized gamma event times with shape parameters (α, β), and rate parameter λ. See FitGenGamma for the parameterization. If a censoring proportion p is provided, the event times are subject to non-informative random right censoring.

Usage

```
rGenGamma(n, a = 1, b = 1, l = 1, p = 0)
```

Arguments

n	Sample size.
a	First shape parameter, α.
b	Second shape parameter, β. For the standard gamma distribution, set $\beta = 1$.
l	Rate.
p	Expected censoring proportion.

Value

Data.frame including the observation times and status indicators.

`rLogNormal` *Simulation from the Log-Normal Distribution*

Description

Generates log-normal event times with location parameter μ and scale parameter σ. See `FitLogNormal` for the parameterization. If a censoring proportion p is provided, the event times are subject to non-informative random right censoring.

Usage

```
rLogNormal(n, m = 0, s = 1, p = 0)
```

Arguments

`n`	Sample size.
`m`	Location.
`s`	Scale.
`p`	Expected censoring proportion.

Value

Data.frame including the observation time and status.

`RoundDF` *Round Data Frames*

Description

Round Data Frames

Usage

```
RoundDF(df, digits = 3)
```

Arguments

`df`	Data.frame.
`digits`	Integer.

Value

Data.frame.

rWeibull *Simulation from the Weibull Distribution*

Description

Generates Weibull event times with shape parameter α and rate parameter λ. See FitWeibull for the parameterization. If a censoring proportion p is provided, the deviates are subject to non-informative random right censoring.

Usage

```
rWeibull(n, a = 1, l = 1, p = 0)
```

Arguments

n	Sample size.
a	Shape.
l	Rate.
p	Expected censoring proportion.

Value

Data.frame including the observation time and status.

show,contrast-method *Show Method for a Contrast of Survival Distributions.*

Description

Show Method for a Contrast of Survival Distributions.

Usage

```
## S4 method for signature 'contrast'
show(object)
```

Arguments

object	An object of class contrast.

show,fit-method *Show Method for Fitted Survival Distributions*

Description

Show Method for Fitted Survival Distributions

Usage

```
## S4 method for signature 'fit'
show(object)
```

Arguments

`object`	An object of class `fit`.

SurvFunc *Survival Functions*

Description

Constructs the survival function for a parameter distribution.

Usage

```
SurvFunc(dist, theta)
```

Arguments

`dist`	String, distribution name.
`theta`	Numeric parameter vector.

Details

The parameter vector theta should contain the following elements, in order, according to the distribution:

Exponential Rate λ.

Gamma Shape α, rate λ.

Generalized Gamma Shape 1 α, shape 2 β, rate λ.

Log-Normal Locaion μ, scale σ.

Weibull Shape α, rate λ.

Value

Survival function.

Examples

```
# Survival function for the generalized gamma.
surv <- SurvFunc(dist = "gen-gamma", theta = c(2, 2, 2))

# Evaluation.
surv(1.0)
```

SurvLogLik *Log Likelihood*

Description

Evaluates the log-likelihood for a parametric survival distribution.

Usage

```
SurvLogLik(
  data,
  dist,
  theta,
  log_scale = FALSE,
  status_name = "status",
  time_name = "time"
)
```

Arguments

`data`	Data.frame
`dist`	Distribution, from among: "exp","gamma","gen-gamma","log-normal","weibull".
`theta`	Parameters, which will vary according to the distribution.
`log_scale`	Are strictly positive parameters on log-scale?
`status_name`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`time_name`	Name of column containing the time to event.

Details

The parameter vector theta should contain the following elements, in order, depending on the distribution:

Exponential Rate λ.

Gamma Shape α, rate λ.

Generalized Gamma Shape 1 α, shape 2 β, rate λ.

Log-Normal Location μ, scale σ.

Weibull Shape α, rate λ.

Value

Scalar value of the log likelihood.

Examples

```
# Generate gamma event time data with 10% censoring.
data <- GenData(n = 1e3, dist = "gamma", theta = c(2, 2), p = 0.1)

# Evaluate log likelihood.
ll <- SurvLogLik(data, dist = "gamma", theta = c(2, 2))

# Generate Weibull event time data with 20% censoring.
data <- GenData(n = 1e3, dist = "weibull", theta = c(2, 2), p = 0.2)

# Evaluate log likelihood.
ll <- SurvLogLik(data, dist = "weibull", theta = c(2, 2))
```

WeiInfo *Weibull Information Matrix.*

Description

Information matrix for the Weibull shape and rate parameters.

Usage

```
WeiInfo(data, shape, rate)
```

Arguments

`data`	Data.frame.
`shape`	Shape parameter, alpha.
`rate`	Rate parameter, lambda.

Value

Numeric information matrix.

`WeiInit` *Weibull Initialization.*

Description

Weibull Initialization.

Usage

```
WeiInit(data, init)
```

Arguments

`data`	Data.frame.
`init`	Initialization list.

Value

Numeric initial value for shape.

`WeiRate` *Weibull Rate MLE*

Description

Profile MLE of the Weibull rate as a function of the shape.

Usage

```
WeiRate(data, shape)
```

Arguments

`data`	Data.frame.
`shape`	Shape parameter.

Value

Numeric rate.

WeiScore *Weibull Profile Score for Shape*

Description

Profile score equation for the Weibull shape parameter.

Usage

```
WeiScore(data, shape)
```

Arguments

`data`	Data.frame.
`shape`	Shape parameter.

Value

Numeric score.

Index