Package 'Temporal'

October 7, 2018

Title Parametric Time to Event Analysis

Version 0.1.1

Description

Performs 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. compParaSurv() compares the mean and median survival experiences of two treatment arms. Candidate distributions currently include the exponential, gamma, generalized gamma, log-logistic, log-normal, and Weibull.

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppEigen

Imports expint, methods, numDeriv, plyr, Rcpp, stats

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Zachary McCaw [aut, cre]

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

Repository CRAN

Date/Publication 2018-10-07 18:32:20 UTC

R topics documented:

`compParaSurv` *Compare Parametric Survival Distribution*

Description

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

Usage

```
compParaSurv(time, status, arm, dist1 = "weibull", dist0, init1 = NULL,
  init0 = NULL, sig = 0.05, eps = 1e-06, maxit = 10, report = F)
```

Arguments

time	Observation time.
status	Status indicator, coded as 1 if an event was observed, 0 if censored.
arm	Treatment indicator, coded as 1 for the target group, 0 for the reference group.
dist1	Distribution to fit for the target group. Selected from among: exp, gamma, gengamma, log-logistic, 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.
init1	List of initial parameter values for the target group. See individual distributions for naming convention.
init0	List of initial parameter values for the reference group. See individual distributions for naming convention.
sig	Significance level, for CIs.
eps	Tolerance for Newton-Raphson iterations.
maxit	Maximum number of NR iterations.
report	Report fitting progress?

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
set.seed(100);
## Weibull and Weibull, different means and medians
# Target group
D1 = rWeibull(n=1e3,a=1,l=1,p=0.2);
D1$arm = 1;
# Reference group
D0 = rWeibull(n=1e3,a=1,l=2,p=0.2);
D0$arm = 0;
# Overall data set
D = rbind(D1,D0);
# Comparison
E = compParaSurv(time=D$time,status=D$status,arm=D$arm,dist1="weibull");

## Gamma and Weibull, different means and medians
# Target group
D1 = rGamma(n=1e3,a=2,l=2,p=0.2);
```

```
D1$arm = 1;
# Reference group
D0 = rWeibull(n=1e3,a=2,l=2/sqrt(pi),p=0.2);
D0$arm = 0;
# Overall data set
D = rbind(D1,D0);
# Comparison
E = compParaSurv(time=D$time,status=D$status,arm=D$arm,dist1="gamma",dist0="weibull");

## Weibull and Log-normal, different means, same median
# Target group
D1 = rLogNormal(n=1e3,m=0,s=2,p=0.2);
D1$arm = 1;
# Reference group
D0 = rWeibull(n=1e3,a=2,l=sqrt(log(2)),p=0.2);
D0$arm = 0;
# Overall data set
D = rbind(D1,D0);
# Comparison
E = compParaSurv(time=D$time,status=D$status,arm=D$arm,dist1="log-normal",dist0="weibull");
```

`contrast-class` *Contrast of Survival Distributions.*

Description

Defines the object class returned by the comparison function.

Slots

`Dist1` Distribution fit to the target group.

`Dist0` Distribution fit to the reference group.

`Model1` Fitted model for the target group.

`Model0` Fitted model for the reference group.

`Contrast` Model contrasts.

`det` *Matrix Determinant*

Description

Calculates the determinant of A.

Usage

```
det(A)
```

Arguments

A	Numeric matrix.

Value

Scalar.

---

dIncGamma *Derivatives of the Incomplete Gamma Function*

Description

Evaluates derivatives of the upper incomplete gamma function, defined as:

$$\Gamma(\alpha, s) = \int_s^\infty u^{\alpha-1} e^{-u} du$$

Usage

```
dIncGamma(a, s, Dirn = "a", Order = 1)
```

Arguments

a	Value of the shape α at which to evaluate.
s	Value of the lower limit s at which to evaluate.
Dirn	Direction in which to differentiate. Select from among "a", "s", and "as".
Order	Order of the derivative, if the direction is either "a", or "s". Select from among 1 and 2.

Value

Scalar value of the partial derivative in the direction of interest.

Examples

```
# First partial in shape at (a=1,s=1)
dIncGamma(a=1,s=1,Dirn="a",Order=1);
# Second partial in lower limit at (a=1,s=1)
dIncGamma(a=1,s=1,Dirn="s",Order=2);
# Mixed partial at (a=1,s=1);
dIncGamma(a=1,s=1,Dirn="as");
```

dLogIncGamma *Derivatives of the Log Incomplete Gamma Function*

Description

Evaluates derivatives of the log of the upper log incomplete gamma function, defined as:

$$\ln \Gamma(\alpha, s) = \int_s^\infty u^{\alpha-1} e^{-u} du$$

Usage

```
dLogIncGamma(a, s, Dirn = "a", Order = 1)
```

Arguments

a	Value of the shape α at which to evaluate.
s	Value of lower limit s at which to evaluate.
Dirn	Direction in which to differentiate. Select from among "a", "s", and "as".
Order	Order of the derivative, if the direction is either "a", or "s". Select from among 1 and 2.

Value

Scalar value of the partial derivative in the direction of interest.

Examples

```
# First partial in shape at (a=1,s=1)
dLogIncGamma(a=1,s=1,Dirn="a",Order=1);
# Second partial in lower limit at (a=1,s=1)
dLogIncGamma(a=1,s=1,Dirn="s",Order=2);
# Mixed partial at (a=1,s=1);
dLogIncGamma(a=1,s=1,Dirn="as");
```

dLogNormSurv *Logarithmic Derivatives of the Normal Survival Function*

Description

Evaluates logarithmic derivatives of the normal survival, defined as:

$$\ln \Phi(s) = \ln \int_s^\infty \frac{e^{-(u^2)/2}}{\sqrt{2\pi}} du$$

Usage

```
dLogNormSurv(s, Order = 1)
```

Arguments

s	Value of the lower limit s at which to evaluate.
Order	Order of the derivative, select from among 1 and 2.

Value

Scalar value of the derivative.

---

estDiff *Difference of Estimates*

Description

Calculate CIs and p-value for the difference of estimated parameters

Usage

```
estDiff(t1, s1, t0, s0, sig = 0.05)
```

Arguments

t1	Treatment estimate
s1	Treatment standard error
t0	Reference estimate
s0	Reference 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 estimated parameters

Usage

```
estRatio(t1, s1, t0, s0, sig = 0.05)
```

Arguments

t1	Treatment estimate
s1	Treatment standard error
t0	Reference estimate
s0	Reference 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.

fit-class *Fitted Survival Distribution*

Description

Defines the object class returned by fitting functions.

Slots

`Distribution` Fitted distribution.

`Parameters` Parameters.

`Information` Information components.

`Outcome` Properties of the fitted distribution.

`fit.Exp` *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

```
fit.Exp(time, status, sig = 0.05)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
# Simulate
D = rWeibull(n=1e3,a=1,l=2);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="exp");
```

`fit.Gamma` *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

```
fit.Gamma(time, status, sig = 0.05, init = NULL, eps = 1e-06,
  maxit = 10, report = F)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.
`init`	List of initial parameter values, including the log of the shape parameter "la" and the log of the rate parameter "ll".
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
# Simulate
D = rGamma(n=1e3,a=2,l=2);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="gamma");
```

`fit.GenGamma` *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

```
fit.GenGamma(time, status, sig = 0.05, init = NULL, eps = 1e-06,
  maxit = 10, report = F)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.
`init`	List of initial parameter values, including the logs of the shape parameters "la" and "lb", and the log of the rate parameter "ll".
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
set.seed(100);
# Simulate
D = rGenGamma(n=1e4,a=2,b=2,l=2);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="gengamma");
```

`fit.LogLogistic` *Log-Logistic Distribution Parameter Estimation*

Description

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

$$f(t) = \frac{\alpha\lambda(\lambda t)^{\alpha-1}}{[1+(\lambda t)^{\alpha}]^2}, t > 0$$

Usage

```
fit.LogLogistic(time, status, sig = 0.05, init = NULL, eps = 1e-06,
  maxit = 10, report = F)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.
`init`	List of initial parameter values, including the log of the shape parameter "la" and the log of the rate parameter "ll".
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

Details

For the log-logistic distribution, the mean is only defined if the shape parameter $\alpha > 1$, and the variance if the shape parameter $\alpha > 2$. Consequently, estimates of the fitted mean and variance are only returned if the estimated shape parameter exceeds these thresholds. For $\alpha \ll 1$, the fitting function may fail.

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
# Simulate
D = rLogLogistic(n=1e3,a=4,l=2);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="log-logistic");
```

`fit.LogNormal` *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

```
fit.LogNormal(time, status, sig = 0.05, init = NULL, eps = 1e-06,
  maxit = 10, report = F)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.
`init`	List of initial parameter values, including the location "m", and the log of the scale parameter "ls".
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
# Simulate
D = rLogNormal(n=1e3,m=0,s=1);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="log-normal");
```

`fit.Weibull` *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

```
fit.Weibull(time, status, sig = 0.05, init = NULL)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`sig`	Significance level, for CIs.
`init`	List containing an initial value for the shape parameter "a".

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.

See Also

- Fitting function for parametric survival distributions `fitParaSurv`

Examples

```
# Simulate
D = rWeibull(n=1e3,a=2,l=2);
# Estimate
M = fitParaSurv(time=D$time,status=D$status,dist="weibull");
```

`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-logistic, log-normal, and Weibull.

Usage

```
fitParaSurv(time, status, dist = "weibull", sig = 0.05, init = NULL,
  eps = 1e-06, maxit = 10, report = F)
```

Arguments

`time`	Observation times.
`status`	Status indicator, coded as 1 if an event was observed, 0 if censored.
`dist`	Distribution to fit, selected from among: exp, gamma, gengamma, log-logistic, log-normal, and weibull.
`sig`	Significance level, for CIs.
`init`	List of initial parameter values. See individual distributions for naming convention.
`eps`	Tolerance for Newton-Raphson iterations.
`maxit`	Maximum number of NR iterations.
`report`	Report fitting progress?

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.

See Also

- Between group comparison of survival experience `compParaSurv`
- Exponential distribution `fit.Exp`
- Gamma distribution `fit.Gamma`
- Generalized gamma distribution `fit.GenGamma`
- Log-logistic distribution `fit.LogLogistic`
- Log-normal distribution `fit.LogNormal`
- Weibull distribution `fit.Weibull`

Examples

```
# Generate cenored gamma data
D = rGamma(n=1e3,a=2,l=2,p=0.2);
# Fit gamma distribution
M = fitParaSurv(time=D$time,status=D$status,dist="gamma");

# Generate cenored weibull data
D = rWeibull(n=1e3,a=2,l=2,p=0.2);
# Fit weibull distribution
M = fitParaSurv(time=D$time,status=D$status,dist="weibull");
```

---

init.GenGamma *Initialization for Generalized Gamma*

Description

Initializes the parameters for the generalized gamma distribution using moment estimators.

Usage

```
init.GenGamma(tobs, L = 0.01, U = 10)
```

Arguments

tobs	Observed event times.
L	Lower bound on shape parameter α.
U	Upper bound on shape parameter α.

Value

List containing the estimated shape and rate parameters on log scale.

---

matInv *Matrix Inverse*

Description

Calcualtes A^{-1}.

Usage

```
matInv(A)
```

Arguments

A	Numeric matrix.

Value

Numeric matrix.

`matIP` *Matrix Inner Product*

Description

Calculates the product $A'B$.

Usage

```
matIP(A, B)
```

Arguments

`A`	Numeric matrix.
`B`	Numeric matrix.

Value

Numeric matrix.

`matOP` *Fast Outer Product*

Description

Calculates the outer product XY'.

Usage

```
matOP(X, Y)
```

Arguments

`X`	Numeric matrix.
`Y`	Numeric matrix.

Value

Numeric matrix.

matQF *Quadratic Form*

Description

Calculates the quadratic form $X'AX$.

Usage

```
matQF(X, A)
```

Arguments

X	Numeric matrix.
A	Numeric matrix.

Value

Numeric matrix.

MMP *Matrix Matrix Product*

Description

Calculates the product AB.

Usage

```
MMP(A, B)
```

Arguments

A	Numeric matrix.
B	Numeric matrix.

Value

Numeric matrix.

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.

qLogLogistic *Quantile Function for the Log-Logistic Distribution*

Description

Quantile function for the log-logistic distribution. See `fit.LogLogistic` for the parameterization.

Usage

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

Arguments

p	Probability.
a	Shape.
l	Rate.

Value

Scalar quantile.

Examples

```
# Median of standard log-logistic distribution
qLogLogistic(p=0.5);
```

qWeibull *Quantile Function for the Weibull Distribution*

Description

Quantile function for the Weibull distribution. See `fit.Weibull` 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 `fit.Gamma` 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

A data.frame including the observation times and status indicators.

Examples

```
# Gamma event times with shape 2 and rate 2
# Expected censoring proportion of 20%
D = rGamma(n=1e3,a=2,l=2,p=0.2);
```

rGenGamma *Simulation from the Generalized Gamma Distribution*

Description

Generates generalized gamma event times with shape parameters (α, β), and rate parameter λ. See `fit.GenGamma` 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

A data.frame including the observation times and status indicators.

Examples

```
# Generalized gamma event times with shapes (2,2) and rate 2
# Expected censoring proportion of 20%
D = rGenGamma(n=1e3,a=2,b=2,l=2,p=0.2);
```

---

rLogLogistic *Simulation from the Log-Logistic Distribution*

Description

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

Usage

```
rLogLogistic(n, a = 4, l = 1, p = 0)
```

Arguments

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

Value

A data.frame including the observation times and status indicators.

Examples

```
# Log-logistic event times with shape 4 and rate 1
# Expected censoring proportion of 20%
D = rLogLogistic(n=1e3,a=4,l=1,p=0.2);
```

rLogNormal *Simulation from the Log-Normal Distribution*

Description

Generates log-normal event times with location parameter μ and scale parameter σ. See `fit.LogNormal` 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

A data.frame including the observation times and status indicators.

Examples

```
# Log-normal event times with location 0 and scale 1
# Expected censoring proportion of 20%
D = rLogNormal(n=1e3,m=0,s=1,p=0.2);
```

rWeibull *Simulation from the Weibull Distribution*

Description

Generates Weibull event times with shape parameter α and rate parameter λ. See `fit.Weibull` 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

A data.frame including the observation times and status indicators.

Examples

```
# Weibull event times with shape 2 and rate 2
# Expected censoring proportion of 20%
D = rWeibull(n=1e3,a=2,l=2,p=0.2);
```

SchurC *Schur complement*

Description

Calculates the efficient information $I_{bb} - I_{ba} I_{aa}^{-1} I_{ab}$.

Usage

```
SchurC(Ibb, Iaa, Iba)
```

Arguments

Ibb	Information of target parameter
Iaa	Information of nuisance parameter
Iba	Cross information between target and nuisance parameters

Value

Numeric matrix.

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.

survLogLik *Survival Distribution Log Likelihood*

Description

Evaluation of the log-likelihood for parametric survival distribution.

Usage

```
survLogLik(time, status, dist, theta, log.scale = T)
```

Arguments

time	Observation times.
status	Status indicator, coded as 1 if an event was observed, 0 if censored.
dist	Distribution.
theta	List of parameters, which will vary according to the distribution.
log.scale	Are positive parameters on log-scale?

Value

Scalar value of the log likelihood.

Examples

```
## Not run:
# Simulate
D = rWeibull(n=1e3,a=2,l=2,p=0.2);
# Log likelihood
ll = survLogLik(time=D$time,status=D$status,
dist="weibull",theta=list("a"=2,"l"=2),log.scale=F);

## End(Not run)
```

Temporal *Temporal: Utilities for Parametric Survival Analysis*

Description

This package performs estimation and inference on parametric survival curves. See fitParaSurv for maximum likelihood estimation of model parameters. See compParaSurv for contrasting the survival experience of two treatment arms.

Author(s)

Zachary R. McCaw

`tr` *Matrix Trace*

Description

Calculates the trace of a matrix A.

Usage

```
tr(A)
```

Arguments

`A`	Numeric matrix.

Value

Scalar.

Index