Package 'lmf'

June 24, 2022

Type Package

Title Functions for Estimation and Inference of Selection in
Age-Structured Populations

Version 1.2.1

Date 2013-10-31

Author Thomas Kvalnes

Maintainer Thomas Kvalnes <thomas.kvalnes@ntnu.no>

Description Provides methods for estimation and statistical
inference on directional and fluctuating selection in age-structured
populations.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-24 11:41:43 UTC

R topics documented:

---

lmf-package   *Functions for estimation and inference of selection in age-structured populations*

Description

This R package estimates directional and fluctuating selection in age-structured populations and provides methods for statistical inference using the procedures developed by Engen et al. 2012. Also compatible with data without age-structure.

Details

Package:	lmf
Type:	Package
Version:	1.2
Date:	2013-10-31
License:	GPL-2
LazyLoad:	yes

Author(s)

Thomas Kvalnes, Centre for Biodiversity Dynamics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Maintainer: Thomas Kvalnes <thomas.kvalnes@ntnu.no>

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

---

atCfn   *Calculate corrected temporal mean coefficients of selection*

Description

Calculate the temporal mean coefficients of selection (alpha) corrected for sampling error, i.e. the best linear predictor (BLP) of alpha.

Usage

```
atCfn(aM, M, At, at)
```

Arguments

aM	the estimated temporal mean selection coefficients.
M	the estimated temporal covariance matrix (fluctuating selection).
At	a list containing the named variance-covariance matrix for each year. Sorted by year.
at	a list containing the named vectors of the estimated selection coefficient for each year. Sorted by year.

Details

Further details are found in Engen et al. 2012.

Value

Returns a vector with the named best linear predictors for the temporal mean coefficients of selection (alpha).

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

[lmf](lmf)

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Extract aM, M, At and at
aM <- lmf.1$aM
M <- lmf.1$M
At <- lmf.1$At
at <- lmf.1$at
#Calculate Best Linear Predictor (BLP)
atCfn(aM = aM, M = M, At = At, at = at)
```

---

boot.lmf	*Bootstrap resampling for class "lmf"*

Description

Generates bootstrap replicates of the estimated parameters in a "lmf" model. Ordinary bootstrap is performed for the projection matrix, while both parametric and ordinary (non-parametric) resampling is available for the remaining parameters in the model. In addition, bootstrapping under any choosen null hypothesis is available for hypothesis testing.

Usage

```
boot.lmf(object, nboot = 1000, what = c("projection", "alpha", "H0", "all"),
  asim = c("ordinary", "parametric"), sig.dj = TRUE,
  H0exp = list(alpha = NULL, M = NULL),
  H0con = c("fs", "nfs", "ds", "nds"), method = c("BFGS"),
  control = list(maxit = 500, reltol = sqrt(.Machine$double.eps)), ...)
```

Arguments

object	a fitted object of of class "lmf".
nboot	the number og bootstraps desired.
what	which set of parameters to bootstrap. Options are "projection" to only resample projection matrix, growth rate (λ), stable age distribution (u) and reproductive values (v). "alpha" to resample demographic and environmental variances as well as all the estimates selection parameters. "H0" to resample temporal coefficients of selection under a given null hypothesis (This options requires specification of the additional arguments H0exp and H0con). "all" (default) to resample all the above mentioned parameters (also here H0exp and H0con must be specified for hypothesis testing or only "projection" and "alpha" will be resampled).
asim	the type of bootstrap for the parameters other than the projection matrix (which is always ordinary bootstrapped). Options are "parametric" (default) and "ordinary".

sig.dj	logical, TRUE(default) to include uncertainty in the estimation of the demographic variance when bootstrapping alpha estimates.
H0exp	a list with the first element a vector containing the expected temproal mean coefficients of selection (alpha) and the second element a matrix containing the elements of the expected temporal variance-covariance matrix (M) under the null hypothesis. This argument needs to be specified to perform hypothesis testing.
H0con	the conditions under which the null hypothesis should be tested. Options are "fs" to assume fluctuating selection, "nfs" to assume no fluctuating selection, "ds" to assume directional selection and "nds" to assume no directional selection. "nds" is not implemented due to increased risk of Type I error if the assumption is not correct, but is included here for completeness.
method	defines what optimalization algorithm to be used in the maximization of the loglikelihood. Alternatives are: "Nelder-Mead", "BFGS" (default), "CG", "L-BFGS-B" and "SANN". Not all are applicable here. See ?optim for details.
control	a list of control parameters for the maximization of the likelihood. maxit sets the maximum number of iterations to use before convergence and reltol sets the relative threshold for improvement in the likelihood which desides whether to continue maximation or end. See ?optim for details.
...	additional arguments to be passed to optim for the maximization of the loglikelihood. See ?optim for options.

Details

The resampling procedures preserve the observed ratios of the different age classes during resampling of the projection matrix.

Ordinary bootstrap will often be subject to bias due to few years of data in most available data sets within biology (generally « 40), thus the parametric bootstrap is recomended for most purposes.

The bootstrap procedure is closely associated with the method deployed in lmf and further details can be found in Engen et al. 2012.

Different from Engen et al. 2012, the sigma2.dj is defined as independent gamma distributed variables with *shape* = $\frac{(EX)^2}{Var(X)}$ and *rate* = $\frac{EX}{Var(X)}$. Where $X = \hat{\sigma}_{dj}^2$ and using the mean and variance from in the paper.

Value

boot.lmf returns a object of class "boot.lmf".

The function summary is used to obtain and print a summary of the bootstrap replicates and to print results from tests of hypotheses. For construction of confidene intervals for the parameters the function ci.boot.lmf is used.

An object of class "boot.lmf" is a list containing at most the following components:

running.time	the total time used for computation.
optim.time	the time used for maximation of the loglikelihood.
call	the matched call.

asim	the value specified of asim.
nboot	the number of bootstrap replicates generated.
uage	the unique age classes in the data set.
nage	the number of unique age classes in the data set.
npar	the number of parameters in the model.
uyear	the unique years in the data set.
nyear	the number of unique years in the data set.
l	the estimated projection matrix.
lboot	the bootstrap replicates of the projection matrix.
lambda	the deterministic multiplicative growth rate of the population.
u	the stable age distribution.
v	the vector of reproductive values for each age class.
luvboot	the bootstrap replicates of λ, u and v.
sigma2.dj	a list containing the demographic variance for each age class. Sorted by age class.
djboot	the bootstrap replicates of sigma2.dj.
sigma2.d	the total demographic variance of the population.
dboot	the bootstrap replicates of sigma2.d.
Atboot	the bootstrap replicates of the yearly variance-covariance matrices. The unscaled variance-covariance matrices are kept constant, but each set of yearly estimates are scaled by the bootstrapped sigma2.dj.
atboot	the bootstrap replicates of the yearly coefficients of selection. This can be performed "parametric"(default) or "ordinary".
M	the estimated temporal covariance matrix (fluctuating selection).
aM	the estimated temporal mean coefficients of selection.
Mboot	the bootstrap replicates of M.
aMboot	the bootstrap replicates of aM.
atCboot	the bootstrap replicates of the best linear predictor for the estimated yearly coefficients of selection (i.e. corrected for sampling errors).
Anf	the estimated temporal covariance matrix assuming no fluctuating selection.
anf	the estimated temporal mean selection coefficients assuming no fluctuating selection.
Anfboot	the bootstrap replicates of Anf.
anfboot	the bootstrap replicates of anf.
sigma2.e	the environmental variance of the population.
eboot	the bootstrap replicates of sigma2.e.
eCboot	the bootstrap replicates of sigma2.eC.
H0aMboot	the bootstrap replicates of aM under the specified null hypothesis H0exp and the assumption of fluctuating selection (Hexp = "fs").

H0anfboot	the bootstrap replicates of anf under the specified null hypothesis H0exp and the assumption of no fluctuating selection (Hexp = "nfs").
H0atnfboot	the bootstrap replicates of at under the specified null hypothesis H0exp and the assumption of directional selection (Hexp = "ds"). These bootstrap replicates are used to generate H0Mnfboot.
H0Mnfboot	the bootstrap replicates of M under the specified null hypothesis H0exp and the assumption of directional selection (Hexp = "ds").

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

[lmf](#), [ci.boot.lmf](#)

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Bootstrap parameters
b.1 <- boot.lmf(object = lmf.1, nboot = 10, sig.dj = TRUE,
 what = "all", asim = "parametric")
#Print
b.1
#Summary
summary(b.1)
#View density plots
plot(b.1)
#Test of hypoteses
b.2 <- boot.lmf(object = lmf.1, nboot = 10, sig.dj = TRUE,
 what = "H0", H0exp = list(rep(0, 3), matrix(0, ncol = 3, nrow = 3)),
 asim = "parametric")
#Summary
summary(b.2)
```

---

ci.boot.lmf   *Confidence intervals for class "boot.lmf"*

Description

Constructs confidence intervals (CIs) for the bootstrapped parameters in an object of class "boot.lmf".

Usage

```
ci.boot.lmf(x, clevel = 0.05)
```

Arguments

x	an object of class "boot.lmf".
clevel	the confidence level required.

Details

`ci.boot.lmf` construct confidence intervals (CIs) from the quantiles of the bootstrap replicates and uses the function `quantile`.

See Engen et al. 2012 for details on the method for estimating and bootstrapping the parameters.

Value

`ci.boot.lmf` returns a list containing the following components:

call	the matched call.
nboot	the number of bootstrap replicates generated.
what	which set of parameters which has been to bootstrapped. See ?boot.lmf for details.
clevel	the confidence level specified.
uage	the unique age classes in the data set.
nage	the number of unique age classes in the data set.
l	CI for the projection matrix.
luv	CI for λ, u and v.
sigma2.dj	CI for the demographic variance for each age class.
sigma2.d	CI for the total demographic variance
M	CI for the estimated temporal covariance matrix.
aM	CI for the estimated temporal mean coefficients of selection
sigma2.e	CI for the environmental variance
Anf	CI for the estimated temporal covariance matrix assuming no fluctuating selection.
anf	CI for the estimated temporal mean selection coefficients assuming no fluctuating selection.

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

lmf, boot.lmf, quantile

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
              age = age, year = year, data = sparrowdata)
#Bootstrap parameters
b.1 <- boot.lmf(object = lmf.1, nboot = 10, sig.dj = TRUE,
 what = "all", asim = "parametric")
#Generate CI
ci.boot.lmf(b.1)
```

---

  eigenl   *Calculating lambda, u and v*

Description

Calculates the deterministic multiplicative growth rate (lambda), the stable age distribution (u) and the reproductive values (v) from a given projection matrix.

Usage

```
eigenl(pm)
```

Arguments

pm   a projection matrix. For instance, the output from the function procomp.

Details

Given a projection matrix (l), this function calculates the real dominant eigenvalue (lambda), and the left (u) and right (v) eigenvectors, defined by l*u = lambda*u and v*l = lambda*v.

Value

eigenl returns a list containing the following components:

lambda	the deterministic multiplicative growth rate of the population from which the projection matrix was estimated
u	the stable age distribution of the population from which the projection matrix was estimated
u	the reproductive values of the population from which the projection matrix was estimated

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

[procomp](#), [promat](#)

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#The uniue age classes
unique.age <- unique(sparrowdata$age)
#Estimate the projection matrix
pro.comp <- procomp(a = sparrowdata, uage = unique.age)
projection.matrix <- promat(pc = pro.comp, nage = length(unique.age))
#Estimate lambda, u and v
eigenl(pm = projection.matrix)
```

---

fs   *Estimate temporal coefficients of selection*

Description

fs is a function which estimates the temporal covariance matrix (i.e. fluctuating selection) and the temporal mean alpha coefficients (i.e. directional selection) through a numerical maximization of a loglikelihood function.

Usage

```
fs(At, at, npar, nyear, method, control, ...)
```

Arguments

At	a list containing the named yearly variance-covariance matrices. Sorted by year.
at	a list containing the named yearly vectors of the estimated selection coefficients. Sorted by year.
npar	the number of parameters in the model for the estimates selection coefficients.
nyear	the number of years with estimates of selection.
method	defines what optimalization algorithm to be used in the maximization of the loglikelihood. Alternatives are: "Nelder-Mead", "BFGS" (default), "CG", "L-BFGS-B" and "SANN". Not all are applicable here. See ?optim for details.
control	a list of control parameters for the maximization of the likelihood. maxit sets the maximum number of iterations to use before convergence and reltol sets the relative threshold for improvement in the likelihood which desides whether to continue maximation or end. See ?optim for details.
...	additional arguments to be passed to optim for the maximization of the loglikelihood. See ?optim for options.

Details

fs is used internally in lmf when estimating temporal coefficients of selection, but given the right data can be executed as a standalone. The input to the arguments At and at are estimated within lmf by correctly combining estimates of coefficients and variance-covariance within each age class and year to a estimate for each year.

fs used optim for the numerical maximization of the log likelihood function lnL.M. Problems of non-positive definite matrices appearing as maximas, due to numerical rounding, are solved with nearPD, which implement the smallest possible numerical changes of some componets of the matrices to achive positive definiteness. These changes does not affect the results in any significant way.

Engen et al. 2012 describe the maximum likelihood method in details.

Value

fs returns a list containing the following components:

convergence	"yes" indicates that the numerical maximation of the likelihood successfully converged before reaching the iteration limit maxit.
iterations	the number of iterations of the function in the numerical maximation of the likelihood.
M	the estimated temporal covariance matrix (fluctuating selection).
aM	the estimated temporal mean selection coefficients.

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

[lmf](), [lnL.M](), [nearPD]()

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
              age = age, year = year, data = sparrowdata)
#Extract At and at
At <- lmf.1$At
at <- lmf.1$at
#Estimate temporal selection coefficients
flusel <- fs(At = At, at = at, npar = 3, nyear = 7, method = "BFGS",
  control = list(maxit = 500, reltol = sqrt(.Machine$double.eps)))
#View output
str(flusel)
flusel
```

---

inv   *Calculate the inverse of a matrix*

Description

Calculates the inverse of a symmetric positive definite matrix from its Cholesky composition.

Usage

```
inv(a)
```

Arguments

a   a symmetric positive definite matrix

Details

This is an interface to the LAPACK routine DPOTRI implemented in the function `chol2inv`. The function calculates cholesky decomposition of the given matrix and inputs this into `chol2inv`.

Value

The inverse of the given matrix.

Author(s)

Thomas Kvalnes

References

Anderson. E. et al. 1999. LAPACK Users' Guide. Third Edition. SIAM [Available online at: http://www.netlib.org/lapack/lug/lapack_lug.html].

See Also

chol, chol2inv

Examples

```
#Example matrix
mat <- matrix(c(1, 0.5, -3, 0.5, 3, 0.5, -3, 0.5, 12), ncol = 3)
#Show that it is positive definite
eigen(mat)$values
#Calculate inverse
inv(mat)
```

---

lm.extract   *Extract linear regression components*

Description

lm.extract fit a linear model and extract coefficients, unscaled covariance matrix, residual variance, fitted values, residuals, degrees of freedom, and leverage and cook's distance for each data point.

Usage

```
lm.extract(formula, data, na.action = na.exclude)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted on the format response ~ terms.
data	a data set containing the variables in the model.
na.action	a function which indicate what should happend when the data contain NAs. The default is na.exclude (see ?na.fail).

Details

lm.extract works through calls to lm, residuals, predict, df.residuals, deviance, vcov, lm.influence and cooks.distance. Consult these functions for further details. The function was written for internal use with lmf, but can be executed as a standalone.

Value

`lm.extract` returns a list containing the following components:

ajt	a named vector of coefficients
res	the residuals
fit	the fitted values
dof	the degrees of freedom
sigma.djt	the residual standard error
Ajt.us	a named unscaled variance-covariance matrix
leverage	the estimated leverage for each data point. I.e. a vector containing the diagonal of the 'hat' matrix (see `lm.influence?`)
cook	the estimated Cook's distance for each data point (see `cooks.distance?`)

Author(s)

Thomas Kvalnes

See Also

`lm`, `summary.lm`

Examples

```
#Simulated data
xx <- rnorm(n = 100, mean = 10, sd = 2)
yy <- xx + 10 + rnorm(n = 100, 0, 2)
#Extract linear model components
extract <- lm.extract(formula = yy ~ xx, data = data.frame(xx = xx, yy = yy))
str(extract)
#Plot the xx-yy relation
plot(xx, yy)
abline(a = extract$ajt[1], b = extract$ajt[2])
```

---

lmf   *Fitting age-structured selection model*

Description

`lmf` fit linear models within each combination of year and age class and estimates coefficients of selection using maximum likelihood procedures. `lmf` is compatible with populations without age-structure.

Usage

```
lmf(formula, age, year, data, na.action = na.exclude,
method = c("BFGS"), control = list(maxit = 500,
reltol = sqrt(.Machine$double.eps)), ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted when estimating coefficients of selection. Format: response ~ terms. The detail of model specification are given under 'Details'.
age	used to define the name of the age column in the data set. Use NULL if no age data are available in the data set.
year	used to define the name of the year column in the data set.
data	data set with individual ids (optional), year of reproduction (year), maternal age (age; may be omitted if a model without age is desired), number of female offspring (recruits), survival to the next reproductive event (survival) and phenotypic measurements. Age classes should have a natural order of increasing age. E.g. 1, 2, 3, ...
na.action	a function which indicate what should happend when the data contain NAs. The default is na.exclude (see ?na.fail).
method	defines what optimalization algorithm to be used in the maximization of the loglikelihood. Alternatives are: "Nelder-Mead", "BFGS" (default), "CG", "L-BFGS-B" and "SANN". Not all are applicable here. See ?optim for details.
control	a list of control parameters for the maximization of the likelihood. maxit sets the maximum number of iterations to use before convergence and reltol sets the relative threshold for improvement in the likelihood which desides whether to continue maximation or end. See ?optim for details.
...	additional arguments to be passed to optim for the maximization of the loglikelihood. See ?optim for options.

Details

lmf use formulas for model specification. These should be formatted as decribed under arguments. Note however that your response should be specified as a two-column matrix with the columns recruits and survival. The first column should give the number of recruits that an individual produced a given year and the second column should contain information on whether the individual survived or not (1 or 0) to the next breeding season. These two columns will be used to calculate the individual reproductive values (Wj) which the model will substitute for the response in the age and year specific linear regressions (i.e. Wj ~ terms).

Value

lmf returns an object of class "lmf".

The function summary is used to obtain and print a summary of the results. For construction of confidene intervals or perform statistical inference on the parameters the function boot.lmf is used.

An object of class "lmf" is a list containing the following components:

running.time	the total time used for computation.
optim.time	the time used for maximation of the loglikelihood.
call	the matched call.

npar	the number of parameters in the model.
uage	the unique age classes in the data set.
nage	the number of unique age classes in the data set.
maxage	the final age class.
l	the estimated projection matrix.
lambda	the deterministic multiplicative growth rate of the population.
u	the stable age distribution.
v	the vector of reproductive values for each age class.
uyear	the unique years in the data set.
nyear	the number of unique years in the data set.
nobs	the number of observations (counting individual-year).
nobs.age	the number of observations per age class.
indnr	assigned individual numbers (1:nobs).
ajt	a list containing the named vectors of the estimated selection coefficient for each age class within each year. Sorted by age class and year.
Ajt.us	a list containing the named unscaled variance-covariance matrix for each age class within each year. Sorted by age class and year.
sigma.djt	a list containing the vectors of residual standard errors from the linear regression for each age class within each year. Sorted by age class and year.
dof	a list containing the vectors of degrees of freedom (dof) from the linear regression for each age class within each year. Sorted by age class and year.
res	a list containing the vectors of residuals from the linear regression for each age class within each year. Sorted by age class and year.
fit	a list containing the vectors of fitted values from the linear regression for each age class within each year. Sorted by age class and year.
leverage	a list containing the vectors of estimated leverage for each data point from the linear regression for each age class within each year (see `lm.influence?`). Sorted by age class and year.
cook	a list containing the vectors of estimated Cook's distance for each data point from the linear regression for each age class within each year (see `cooks.distance?`). Sorted by age class and year.
sigma2.dj	a list containing the demographic variance for each age class. Sorted by age class.
sigma2.dj.dof	a list containing the degrees of freedom (dof) for the demographic variance for each age class. Sorted by age class.
sigma2.dj.sd	a list containing the standard deviation (sd) for the demographic variance for each age class. Sorted by age class.
sigma2.d	the total demographic variance of the population.
sigma2.d.dof	the degrees of freedom (dof) for the total demographic variance of the population.

sigma2.d.sd	the standard deviation (sd) for the total demographic variance of the population.
Ajt	a list containing the named variance-covariance matrix (scaled by sigma2.dj) for each age class within each year. Sorted by age class and year.
at	a list containing the named vectors of the estimated selection coefficient for each year. Sorted by year.
At	a list containing the named variance-covariance matrix (scaled by sigma2.dj) for each year. Sorted by year.
convergence	"yes" indicates that the numerical maximation of the likelihood successfully converged before reaching the iteration limit maxit.
iterations	the number of iterations of the function in the numerical maximation of the likelihood.
M	the estimated temporal covariance matrix (fluctuating selection).
aM	the estimated temporal mean selection coefficients.
atC	the best linear predictor for the estimated yearly selection coefficients (i.e. corrected for sampling errors).
Anf	the estimated temporal covariance matrix assuming no fluctuating selection.
anf	the estimated temporal mean selection coefficients assuming no fluctuating selection.
sigma2.e	the environmental variance of the population.
data.set	the data set used in the analyses with a column of individual reproductive values addad

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

procomp, promat, eigenl, lm.extract, fs, atCfn, nfs, boot.lmf

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#View diagnostic plots
plot(lmf.1)
#View output
print(lmf.1)
#Print summary
summary(lmf.1)
```

___

lnL.M   *Maximum likelihood function for temporal coefficients of selection*

___

Description

lnL.M calculates the loglikelihood of the maximum likelihood function for temporal coefficients of selection from Engen et al. 2012. Returns the estimate of alpha for a given temporal variance-covariance matrix (M) if desired.

Usage

```
lnL.M(D, At, at, npar, ret.alphas = FALSE)
```

Arguments

D	a vector with the non-zero elements of the upper triangular matrix of the Cholesky decomposition of a temporal variance-covariance matrix M.
At	a list containing the named yearly variance-covariance matrices. Sorted by year.
at	a list containing the named yearly vectors of the estimated selection coefficients. Sorted by year.
npar	the number of parameters in the model for the estimates selection coefficients.
ret.alphas	logical. If TRUE the function returns the vector with the estimates of the temporal mean selection coefficients (alpha) for the given temporal variance-covariance matrix M. FALSE (default) makes the function return the loglikelihood for the given M.

Details

The function was developed for internal use in fs, but can be applied as a standalone.

The upper triangular matrix (D) of the Cholesky decomposition of M is defined as M = t(D)%*%D.

Details of the method is provided in Engen et al. 2012.

Value

lnL.M returns the loglikelihood estimate (for ret.alpha = FALSE) or the temporal mean selection coefficients (for ret.alpha = TRUE) for a given temporal variance-covariance matrix M.

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

lmf, fs, chol

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Extract At and at
At <- lmf.1$At
at <- lmf.1$at
#Estimate D
D <- as.vector(chol((Reduce('+', At) / length(At))))
D <- D[D != 0]
#Estimate loglikelihood
lnL.M(D = D, At = At, at = at, npar = 3, ret.alphas = FALSE)
#Estimate temporal mean selection coefficients
lnL.M(D = D, At = At, at = at, npar = 3, ret.alphas = TRUE)
```

---

nearPD *Find nearest positive definite matrix*

Description

Compute the nearest positive definite matrix to an approximate one, typically a correlation or variance-covariance matrix.

Usage

```
nearPD(x, corr = FALSE, keepDiag = FALSE, do2eigen = TRUE, doSym = FALSE,
doDykstra = TRUE, only.values = FALSE, only.matrix = TRUE, eig.tol = 1e-06,
conv.tol = 1e-07, posd.tol = 1e-08, maxit = 100, trace = FALSE)
```

Arguments

x	numeric $n * n$ approximately positive definite matrix, typically an approximation to a correlation or covariance matrix.
corr	logical indicating if the matrix should be a *correlation* matrix.
keepDiag	logical, generalizing corr: if TRUE, the resulting matrix should have the same diagonal (diag(x)) as the input matrix.
do2eigen	logical indicating if a posdefify eigen step should be applied to the result of the Higham algorithm.
doSym	logical indicating if X <- (X + t(X))/2 should be done, after X <- tcrossprod(Qd, Q). Some doubt if this is necessary.

doDykstra	logical indicating if Dykstra's correction should be used; true by default. If false, the algorithm is basically the direct fixpoint iteration $Y(k) = P_U(P_S(Y(k-1)))$.
only.values	logical; if TRUE, the result is just the vector of eigen values of the approximating matrix.
only.matrix	logical indicating if only the matrix should be returned.
eig.tol	defines relative positiveness of eigenvalues compared to largest one, λ_1. Eigen values λ_k are treated as if zero when $\lambda_k/\lambda_1 = eig.tol$.
conv.tol	convergence tolerance for Higham algorithm.
posd.tol	tolerance for enforcing positive definiteness (in the final posdefify step when do2eigen is TRUE).
maxit	maximum number of iterations allowed.
trace	logical or integer specifying if convergence monitoring should be traced.

Details

This function is identical to nearPD in package **Matrix** as far as the algorithmic method is concerned, but has an addition of the argument only.matrix to ease its application within the function fs, has lost the argument ensureSymmetry and have a small change in the list returned when only.matrix = FALSE.

Please see nearPD in package **Matrix** for further details.

Value

nearPD returns a numeric vector of eigen values of the approximating matrix if only.values = TRUE, returns the computed positive definite matrix if only.matrix = TRUE and else returns a list with the following componets:

mat	matrix of class "dpoMatrix", the computed positive-definite matrix.
eigenvalues	numeric vector of eigenvalues of mat.
corr	logical, just the argument corr.
normF	the Frobenius norm ([norm](x-X, ″F″)) of the difference between the original and the resulting matrix.
iterations	number of iterations needed.
converged	logical indicating if iterations converged.

Author(s)

Jens Oehlschlaegel donated a first version. Subsequent changes by the Matrix package authors and present modifications by Thomas Kvalnes.

References

Cheng, S.H. and Higham, N. 1998. A Modified Cholesky Algorithm Based on a Symmetric Indefinite Factorization. SIAM Journal on Matrix Analysis and Applications, 19, 1097-1110.

Knol, D.L. and ten Berge, J.M.F. 1989. Least-squares approximation of an improper correlation matrix by a proper one. Psychometrika, 54, 53-61.

Higham, N. 2002. Computing the nearest correlation matrix - a problem from finance. IMA Journal of Numerical Analysis, 22, 329-343.

See Also

fs, lmf, nearPD, posdefify

Examples

```
#Simulated non-positive definite (PD) matrix
nonPD <- matrix(c(2.04e-03, 3.54e-05, 7.52e-03, 3.54e-05, 6.15e-07,
  1.30e-04, 7.52e-03, 1.30e-04, 2.76e-02), ncol = 3)
#View eigenvalues (PD = only positive eigenvalues)
eigen(nonPD)
#Calculate PD matrix
PD <- nearPD(nonPD, only.matrix = TRUE)
PD
#View eigenvalues
eigen(PD)
#More thorough examples are given in the help pages for nearPD
 #in the Matrix package.
```

---

nfs	*Estimate temporal coefficients of selection assuming no fluctuating selection*

Description

nfs is a function which estimates the temporal covariance matrix and the temporal mean alpha coefficients (i.e. directional selection) under the assumption of no fluctuating selection.

Usage

```
nfs(At, at, npar, nyear)
```

Arguments

At	a list containing the named yearly variance-covariance matrices. Sorted by year.
at	a list containing the named yearly vectors of the estimated selection coefficients. Sorted by year.
npar	the number of parameters in the model for the estimates selection coefficients.
nyear	the number of years with estimates of selection.

Details

nfs is used internally in `lmf` when estimating temporal mean coefficients of selection under the assumption of no fluctuating selection, but given the right data can be executed as a standalone. The input to the arguments At and at are estimated within `lmf` by correctly combining estimates of coefficients and variance-covariance within each age class and year to a estimate for each year.

Engen et al. 2012 describe the method in details.

Value

nfs returns a list containing the following components:

Anf	the estimated temporal covariance matrix under the assumption of no flucutating selection.
anf	the estimated temporal mean selection coefficients under the assumption of no fluctuating selection.

Author(s)

Thomas Kvalnes

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

`lmf`, `fs`

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
              age = age, year = year, data = sparrowdata)
#Extract At and at
At <- lmf.1$At
at <- lmf.1$at
#Estimate temporal selection coefficients under the assumption of no
  #fluctuating selection
noflusel <- nfs(At = At, at = at, npar = 3, nyear = 7)
#View output
str(noflusel)
noflusel
```

---

plot.boot.lmf *Plotting function for class "boot.lmf"*

Description

Plots density plots for bootstrap replicates of parameters from objects of class "lmf".

Usage

```
## S3 method for class 'boot.lmf'
  ## S3 method for class 'boot.lmf'
plot(x, what = c("all"), ...)
```

Arguments

x   an object of class "boot.lmf".

what   what parameters is to be plotted. Options are "projection", "alpha" and "all".

...   additional arguments to be passed to methods, such as graphical parameters (see
            par).

Details

Density of parameters is calculated using the function density.

Author(s)

Thomas Kvalnes

See Also

boot.lmf, density, plot.default

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
                age = age, year = year, data = sparrowdata)
#Bootstrap parameters
b.1 <- boot.lmf(object = lmf.1, nboot = 10, sig.dj = TRUE,
 what = "all", asim = "parametric")
#Density plot
plot(b.1)
```

---

plot.lmf   *Plot diagnostics for an lmf object*

Description

Plots four diagnostic plots for objects of class "lmf": a plot of residuals against fitted values, a Normal Q-Q plot, a Scale-Location plot of the square root of standardized residuals against fitted values and a plot of standardized residuals against leverage with indications of cooks distance.

Usage

```
## S3 method for class 'lmf'
  ## S3 method for class 'lmf'
plot(x, what = "total", ...)
```

Arguments

x	an object of class "lmf".
what	the part of the model for which diagnostic plots should be generated. Options are 'total' (default) for the overall plots and 'age-year' for plots for each separate linear regression for each age within each year.
...	additional arguments to be passed through to plotting functions.

Details

The function is constructed partly from modifications of script from the plot.lm function in package **stats**. See ?plot.lm for details on the plots provided.

Author(s)

Thomas Kvalnes

See Also

lmf, plot.lm

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Diagnostic plots across all age classes and years
par(mfrow = c(2, 2))
plot(lmf.1)
#Diagnostic plots for each age class within each year (the
#actual linear regressions)
```

```
par(mfrow = c(2, 2))
plot(lmf.1, what = "age-year")
```

---

projection   *The projection matrix*

Description

Estimates the components of the projection matrix and constructs the matrix given data on fecundity and survival in a age-structured population.

Usage

```
procomp(a, uage)
promat(pc, nage)
```

Arguments

a	a data set with fecundity and survival of individuals with known age over several years. Each individual can be represented once a year.
uage	the unique age classes in the data set.
pc	a matrix with columns age, fecundity and survival, i.e. the output from running the function procomp.
nage	the number of unique age classes in the data set.

Details

This function was made for internal use in lmf, but can be applied as a standalone.

The data set needs to contain the columns recruits (number of female offspring), survival (1 if the individual survives to the next reproductive event and 0 if it dies) and age.

The contruction of the projection matrix is divided into two functions for ease of storing and handling the estimates.

Value

procomp returns a matrix with the components of the projection matrix having columns age, fecundity and survival, while promat takes the output from procomp and returns the projection matrix.

Author(s)

Thomas Kvalnes

References

Caswell, H. 2000. Matrix Population Models, 2nd edn. Sinauer, Sunderland, Massachusetts. Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

See Also

eigenl

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#The uniue age classes
unique.age <- unique(sparrowdata$age)
#Estimate the components of the projection matrix
projection.components <- procomp(a = sparrowdata, uage = unique.age)
#View components
projection.components
#Set up the projection matrix
promat(pc = projection.components, nage = length(unique.age))
```

---

rmnorm   *Generate random numbers from the multivariate normal distribution*

Description

rmnorm generate random numbers from a multivariate normal distribution.

Usage

```
rmnorm(n = 1, mean = rep(0, d), varcov)
```

Arguments

n	the number of random vectors to be generated.
mean	a vector with means of length *d*.
varcov	a variance-covariance matrix with dimentions *d * d*.

Details

This is a modification of the function rmnorm provided in **mnormt**. The function works around problems of non-positive definite variance-covariance matrices due to numerical rounding by use of the function nearPD. Furthermore, when only a single random vector is generated, the function now returns a named random vector with names inherited from the colum names of the variance-covariance matrix.

Value

For n > 1 rmnorm returns a matrix of n rows of random vectors, while for n = 1 rmnorm returns a named random vector.

Author(s)

Fortran code of SADMVN and most auxiliary functions by Alan Genz, some additional auxiliary functions by people referred to within his program. Porting to R and additional R code by Adelchi Azzalini, with current modifications by Thomas Kvalnes.

References

Genz, A. 1992. Numerical Computation of Multivariate Normal Probabilities. Journal of Computational and Graphical Statiststics, 1, 141-149.

Genz, A. 1993. Comparison of methods for the computation of multivariate normal probabilities. Computing Science and Statistics, 25, 400-405.

Genz, A.: Fortran code available at http://www.math.wsu.edu/math/faculty/genz/software/fort77/mvn.f

See Also

Normal, rnorm, rmnorm, nearPD

Examples

```
#Variance-covariance matrix
varcov <- matrix(c(2.047737e-03, 3.540039e-05, 0.0075178920, 3.540039e-05,
6.122832e-07, 0.0001299661, 7.517892e-03, 1.299661e-04, 0.0276005740), ncol = 3)
#Set names
nam <- c("a", "b", "c")
dimnames(varcov) <- list(nam, nam)
#Check positive definiteness (all positive eigenvalues = positive definite)
eigen(varcov) $values
#Mean
mean <- c(1, 0.3, 0.5)
#Generate n = 1 random vector
rmnorm(n = 1, mean = mean, varcov = varcov)
#Generate n = 10 random vectors
rmnorm(n = 10, mean = mean, varcov = varcov)
#Generate n = 1 random vectors when varcov is non-positive definite
#Non-positive definite varcov matrix
varcov2 <- matrix(c(2.04e-03, 3.54e-05, 7.52e-03, 3.54e-05, 6.15e-07,
  1.30e-04, 7.52e-03, 1.30e-04, 2.76e-02), ncol = 3)
  dimnames(varcov2) <- dimnames(varcov)
eigen(varcov2)
#Random vector
rmnorm(n = 1, mean = mean, varcov = varcov2)
```

---

se	*Standard error*

Description

Calculates the standard error of the values in x

Usage

```
se(x, na.rm = FALSE)
```

Arguments

x	a numeric vector.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

The standard error of a length one vector is NA and if x is not a vector an error will be printed.

The standard error is defined as:

$$SE = \sqrt{\frac{var(x)}{length(x)}}$$

Value

The standard error of x is returned.

Author(s)

Thomas Kvalnes

See Also

var, length, sqrt

Examples

```
#The standard error of samples with the same mean and standard deviation, but
  #of different sizes.
se(rnorm(n = 10, mean = 5, sd = 1))
se(rnorm(n = 100, mean = 5, sd = 1))
se(rnorm(n = 1000, mean = 5, sd = 1))
se(rnorm(n = 10000, mean = 5, sd = 1))
```

---

sparrowdata	*House Sparrow data set*

Description

A data set on a wild population of house sparrow *Passer domesticus*.

Usage

```
data(sparrowdata)
```

Format

A data frame with 116 observations on the following 7 variables.

`id` unique individual identifier

`year` the year of breeding

`age` the age of the individual

`survival` equals 1 if the individual survives, otherwise 0

`recruits` the number of recruits produced, i.e. offspring that survives to the next breeding season

`weight` the body mass of the individuals at fledgling, i.e. when the bird was ready to leave the nest

`tars` the tarsus length of the individuals at fledgling, i.e. when the bird was ready to leave the nest

Details

Further details on the data set is provided by Engen et al. 2012.

References

Engen, S., Saether, B.-E., Kvalnes, T. and Jensen, H. 2012. Estimating fluctuating selection in age-structured populations. Journal of Evolutionary Biology, 25, 1487-1499.

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#View data
str(sparrowdata)
head(sparrowdata, 10)
```

---

summary.boot.lmf   *Summarizing bootstraps of lmf fits*

Description

summary method for class "boot.lmf".

Usage

```
  ## S3 method for class 'boot.lmf'
summary(object, ret.bootstraps = FALSE, ...)
  ## S3 method for class 'summary.boot.lmf'
print(x, digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

`object`	an object of class "boot.lmf". empty
`ret.bootstraps`	logical. If TRUE the bootstrap replicates are returned in the output. empty
`x`	an object of class "summary.boot.lmf". empty
`digits`	the number of significant digits to use when printing. empty
`signif.stars`	logical. If TRUE, 'significance stars' are printed for each coefficient.
`...`	further arguments passed to or from other methods.

Details

`summary.boot.lmf` formats bootstrap replicates in a user-friendly way, and formats the temporal coefficients and variance-covariance matrix into easily read tables for hypothesis tests.

Value

The function `summary.boot.lmf` computes and returns a list of summary statistics of the bootstrap replicates of a fitted `lmf` model given in `object`.

An object of class "summary.boot.lmf" is a list containing at most the following components:

`call`	the matched call.
`nboot`	the number of bootstrap replicates generated.
`lest`	the estimated projection matrix.
`lboot.mean`	the bootstrap mean projection matrix.
`lbias`	the bootstrap bias of the components of the projection matrix.
`lboot.sd`	the boostrap standard deviation of the components of the projection matrix.
`luv`	the estimate, bootstrap mean, bias and standard deviation of the deterministic multiplicative growth rate of the population (λ), the stable age distribution (u) and the reproductive values (v).
`sigma2.e`	the estimate, bootstrap mean, bias and standard deviation of the environmental variance of the population.
`sigma2.dd`	the estimate, bootstrap mean, bias and standard deviation of the demographic variances (by age class and in total).
`aM`	the estimate, bootstrap mean, bias and standard deviation of the estimated temporal mean selection coefficients.
`Mest`	the estimated temporal variance-covariance matrix (M).
`Mboot.mean`	the bootstrap mean temporal variance-covariance matrix.
`Mbias`	the bootstrap bias of the components of the temporal variance-covariance matrix.
`Mboot.sd`	the boostrap standard deviation of the components of the temporal variance-covariance matrix.
`anf`	the estimate, bootstrap mean, bias and standard deviation of the estimated temporal mean selection coefficients under the assumption of no fluctuating selection.

Anfest	the estimated temporal variance-covariance matrix under the assumtion of no fluctuating selection.
Anfboot.mean	the bootstrap mean temporal variance-covariance matrix under the assumption of no fluctuating selection.
Anfbias	the bootstrap bias of the components of the temporal variance-covariance matrix under the assumption of no fluctuating selection.
Anfboot.sd	the boostrap standard deviation of the components of the temporal variance-covariance matrix under the assumption of no fluctuating selection.
coefficients.aH0aMboot	
	the estimated temporal mean selection coefficients, with bootstrapped standard errors, number of successes with regard to the null hypothesis and associated p-values. All under the specified null hypothesis H0exp and the assumption of fluctuating selection (Hexp = "fs").
coefficients.aH0anfboot	
	the estimated temporal mean selection coefficients under the assumtion of no fluctuating selection, with bootstrapped standard errors, number of successes with regard to the null hypothesis and associated p-values. All under the specified null hypothesis H0exp and the assumption of no fluctuating selection (Hexp = "nfs").
coefficients.aH0Mnfboot	
	the estimated temporal components of the variance-covariance matrix, with bootstrapped standard errors, number of successes with regard to the null hypothesis and associated p-values. All under the specified null hypothesis H0exp and the assumption of directional selection (Hexp = "ds").
lluvboot	the bootstrap replicates of the projection matrix (columns f (fecundity) and s (survial)), lambda, the stable age distribution (u) and the reproductive values (v). Numbers in the column names indicate age class.
deboot	the bootstrap replicates of the demographic and environmental variances. Numbers in the column names indicate age class.
atAboot	the bootstrap replicates of the yearly coefficients of selection (at) and variance-covariance matrix (At). The first column indicate boostrap number and the second the boostrapped year. The subsequent columns contain coefficients (where (Intercept) (at) is the first coefficient), and components of the variance-covariance matrix (where (Intercept)-(Intercept) (At) is the first component (from the diagonal) of the matrix)
aMMboot	the bootstrap replicates of the temporal mean coefficients of selection (aM) and variance-covariance matrix (M). The first columns contain coefficients (where (Intercept) (a(M)) is the first coefficient), and subsequent columns contain the components of the variance-covariance matrix (where (Intercept)-(Intercept) (M) is the first component (from the diagonal) of the matrix)
atCboot	the bootstrap replicates of the yearly coefficients of selection (atC) corrected for sampling error. The first column indicate boostrap number, the second the boostrapped year and the subsequent columns contain the boostrapped coefficients.
anfAboot	the bootstrap replicates of the temporal mean coefficients of selection (anf) and variance-covariance matrix under the assumption of no fluctuating selection.

The first columns contain coefficients (where (Intercept) (a(M=0)) is the first coefficient), and subsequent columns contain the components of the variance-covariance matrix (where (Intercept)-(Intercept) (At(M=0)) is the first component (from the diagonal) of the matrix)

H0aMboot	the bootstrap replicates of aM under the specified null hypothesis H0exp and the assumption of fluctuating selection (Hexp = "fs").
H0anfboot	the bootstrap replicates of anf under the specified null hypothesis H0exp and the assumption of no fluctuating selection (Hexp = "nfs").
H0atnfboot	the bootstrap replicates of at under the specified null hypothesis H0exp and the assumption of directional selection (Hexp = "ds"). These bootstrap replicates are used to generate H0Mnfboot.
H0Mnfboot	the bootstrap replicates of M under the specified null hypothesis H0exp and the assumption of directional selection (Hexp = "ds").

Author(s)

Thomas Kvalnes

See Also

lmf, summary, boot.lmf

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Bootstrap parameters
b.1 <- boot.lmf(object = lmf.1, nboot = 10, sig.dj = TRUE,
 what = "all", asim = "parametric")
#Summary
summary(b.1)
```

---

summary.lmf *Summarizing lmf fits*

Description

summary method for class "lmf".

Usage

```
  ## S3 method for class 'lmf'
summary(object, what.level = c("age", "year", "total"), ...)
  ## S3 method for class 'summary.lmf'
print(x, digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

`object`	an object of class "lmf".
`what.level`	the lowest level of organisation at which parameters should be summarized. Options are "age" to summarize from the level of specific age classes, "year" to summarize from the level of years and "total" (default) to summarize the temporal parameters.
`x`	an object of class "summary.lmf".
`digits`	the number of significant digits to use when printing.
`signif.stars`	logical. If TRUE, 'significance stars' are printed for each coefficient.
`...`	further arguments passed to or from other methods.

Details

`print.summary.lmf` formats the coefficients, standard errors and variance-covariance matrices into easily read tables. The tests of significance of the coefficients of selection should be considered as suggestive and should be interpreted with caution. For tests of signifcanse a robust approach is provided through the function `boot.lmf`.

Value

The function `summary.lmf` computes and returns a list of summary statistics of the fitted `lmf` model given in `object`.

An object of class "summary.lmf" is a list containing at most the following components:

`call`	the matched call.
`what.level`	the lowest level of organisation at which parameters should be summarized.
`uage`	the unique age classes in the data set.
`nage`	the number of unique age classes in the data set.
`uyear`	the unique years in the data set.
`nyear`	the number of unique years in the data set.
`l`	the estimated projection matrix.
`lambda`	the deterministic multiplicative growth rate of the population.
`u`	the stable age distribution.
`v`	the vector of reproductive values for each age class.
`sigma2.e`	the environmental variance of the population.
`sigma2.dj`	a list containing the demographic variance for each age class. Sorted by age class.
`sigma2.dj.dof`	a list containing the degrees of freedom (dof) for the demographic variance for each age class. Sorted by age class.
`sigma2.dj.sd`	a list containing the standard deviation (sd) for the demographic variance for each age class. Sorted by age class.
`sigma2.d`	the total demographic variance of the population.

sigma2.d.dof	the degrees of freedom (dof) for the total demographic variance of the population.
sigma2.d.sd	the standard deviation (sd) for the total demographic variance of the population.
coefficients.ajt	
	the estimated selection coefficients for each age class within each year, with standard errors, t values and p-values.
Ajt	a list containing the named variance-covariance matrix for each age class within each year. Sorted by age class and year.
coefficients.at	
	the estimated selection coefficients within each year, with standard errors, t values and p-values.
At	a list containing the named variance-covariance matrix for each year. Sorted by year.
coefficients.atC	
	the estimated selection coefficients within each year corrected for sampling error, with standard errors, t values and p-values.
coefficients.aM	
	the estimated temporal mean selection coefficients, with standard errors, t values and p-values.
M	the estimated temporal covariance matrix (fluctuating selection).
coefficients.anf	
	the estimated temporal mean selection coefficients under the assumtion of no fluctuating selection, with standard errors, t values and p-values.
Anf	the estimated temporal covariance matrix assuming no fluctuating selection.

Author(s)

Thomas Kvalnes

See Also

lmf, summary, boot.lmf

Examples

```
#Data set from Engen et al. 2012
data(sparrowdata)
#Fit model
lmf.1 <- lmf(formula = cbind(recruits, survival) ~ weight + tars,
             age = age, year = year, data = sparrowdata)
#Summary
summary(lmf.1)
#Summary from within years
summary(lmf.1, what.level = "year")
#Summary from within age classes
summary(lmf.1, what.level = "age")
```

Index