Package 'sae2'

August 23, 2023

Version 1.2-1

Date 2023-08-22

Title Small Area Estimation: Time-Series Models

Author Robert Fay [aut, cre],
Mamadou Diallo [aut]

Maintainer Robert Fay <bobfay@hotmail.com>

Depends R (>= 2.14.0), MASS, survey, stats

Suggests sae

Description Time series area-level models for small area estimation.
The package supplements the functionality of the sae package.
Specifically, it includes EBLUP fitting of the original Rao-Yu model,
which in the original form did not have a spatial component. The
package also offers a modified ('dynamic') version of the Rao-Yu model,
replacing the assumption of stationarity. Both univariate and
multivariate applications are supported. Of particular note is the
allowance for covariance of the area-level sample estimates over time,
as encountered in rotating panel designs such as the U.S. National
Crime Victimization Survey or present in a time-series of 5-year
estimates from the American Community Survey. Key references to the
methods include
J.N.K. Rao and I. Molina (2015, ISBN:9781118735787),
J.N.K. Rao and M. Yu (1994) <doi:10.2307/3315407>, and
R.E. Fay and R.A. Herriot (1979) <doi:10.1080/01621459.1979.10482505>.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-23 09:10:02 UTC

R topics documented:

---

sae2-package   *Small Area Estimation: Time-Series Models.*

Description

Time series are-level models for small area estimation. The package supplements the functionality
of the package sae. Specifically, it includes EBLUP fitting of the original Rao-Yu model, which
did not have a spatial component. It also offers a modified ("dynamic") version of the Rao-Yu
model, replacing the assumption of stationarity. Both univariate and multivariate applications are
supported. Of particular note is the allowance for covariance of the area-level sample estimates
over time, as encountered in rotating panel designs such as the U.S. National Crime Victimization
Survey or present in a time-series of 5-year estimates from the American Community Survey.

Details

Package:	sae2
Type:	Package
Version:	1.2-1
Date:	2023-08-22
License:	GPL-2

The package provides two primary functions, eblupRY and eblupDyn, to fit non-spatial time-series
small area models to area-level data. The function mvrnormSeries provides simulated data under
either model. Functions geo_ratios and vcovgen can assist in preparing the input.

Author(s)

Robert E. Fay, Mamadou S. Diallo

Maintainer: Robert E. Fay <bobfay@hotmail.com>

References

- Fay, R.E. and Herriot, R.A. (1979). Estimation of income from small places: An application of
James-Stein procedures to census data. Journal of the American Statistical Association 74, 269-277.

- Fay, R.E., Planty, M. and Diallo, M.S. (2013). Small area estimates from the National Crime Vic-
timization Survey. Proceedings of the Joint Statistical Meetings. American Statistical Association,
pp. 1544-1557.

- Rao, J.N.K. and Molina, I. (2015). Small Area Estimation, 2nd ed. Wiley, Hoboken, NJ.

- Rao, J.N.K. and Yu, M. (1994). Small area estimation by combining time series and cross-sectional data. Canadian Journal of Statistics 22, 511-528.

See Also

[sae](sae)

| dynRYfit | *Internal fitting function for Dynamic and Rao-Yu models* |

Description

Function designed to be called by either `eblupDyn` or `eblupRY` to produce EBLUP small area estimates of the dynamic or Rao-Yu time series models through either ML or REML estimation of the variance components. For completeness, the function is documented here, but users are encouraged to base applications on calls to the more convenient `eblupDyn` or `eblupRY`.

Usage

```
dynRYfit(y,  X,  M,  TI,  NV=1,  vcov_e,  maxiter=100,
         iter.tol=.1e-5,  ncolx=NULL,  sig2_u=1,  sig2_v=1,
         rho=.8,  rho_u =.4,  delta=NULL,  rho.fixed=NULL,
         y.include=NULL, ids=NULL,  contrast.matrix=NULL,
         baby.steps=TRUE,  dampening=NULL,  iter.history=NULL,
         sig2.min.factor=.0001,  max.rho_u=.98,  max.rho=NULL,
         tol=.Machine$double.eps, y.rescale=NULL,
         llike.only=FALSE,  method=c("REML", "ML"),
         model=c("dyn", "RY"))
```

Arguments

y	For a univariate model, the dependent variable sorted in ascending order by time within domain. For a multivariate model, the dependent variables sorted in ascending order by time within variable within domain.
X	A matrix of independent variables with the same number of rows as the length of y.
M	The total number of domains, equivalent to D in `eblupDyn` and `eblupRY`.
TI	The number of time instances (constant for all domains).
NV	The number of dependent variables.
vcov_e	For the univariate model, the sampling covariance matrix for the direct estimates of the M*TI elements of the dependent variable. The covariance matrix should be in the form of a square matrix with M*TI rows and columns. Non-zero covariances between domains are not allowed, so the matrix must have block diagonal form with M blocks, each of which is a square matrix with TI rows and columns. Note that within domain, non-zero covariances are allowed over time.

	For the multivariate model, the square covariance matrix for the `M*NV*TI` elements of the dependent variables. The matrix should be in the form of a square matrix with `M*NV*TI` rows and columns. Time should vary within variable, which should vary within domain. Non-zero covariances between domains are not allowed, but non-zero covariances may be present across time and between variables.
maxiter	The maximum number of iterations allowed for the Fisher-scoring algorithm, with a default value of 100.
iter.tol	The convergence tolerance limit for the Fisher-scoring algorithm, with a default value of .000001.
ncolx	For a univariate model, the number of columns of X. For a multivariate model, a vector of length NV must be specified giving the number of columns of X used for each dependent variable.
sig2_u	An initial starting value or values for the variance of the random increments.
sig2_v	An initial starting value or values for a domain level random effect. In the Rao-Yu model, the random effect is constant over time, whereas in the dynamic model it is an initial effect subject to dampening over time.
rho	The correlation across time. This correlation is assumed to be the same for the dependent variables in the multivariate model.
rho_u	For NV > 1 only, the (NV*(NV-1))/2 starting values for the correlations between the random effects of the different dependent variables. If a single value is given, it will be used for the (NV*(NV-1))/2 components. The sort order corresponds to a lower triangle of the covariance matrix.
delta	The random effect components in the preferred internal order. Specification of `delta` will override any specification of `sig2_u`, `sig2_v`, `rho`, or `rho_u`. In the univariate case, delta should contain `sig2_u`, `sig2_v`, and `rho`. In the bivariate case, `delta` should be of length 6 with `sig2_u` and `sig2_v`, each of length 2, `rho`, and `rho_u`. Similarly, for 3 dependent variables, the length of `delta` is 10, with the last 3 elements `rho_u` in lower triangular form.
rho.fixed	If TRUE, the value of `rho` imbedded in `delta`, if specified, or else given by `rho` will remain fixed during the iterations. Among other features, this allows the likelihood function for trial values of rho to be computed at the maximum over the other random effect parameters.
y.include	If specified, vector of length `M` to indicate which domains to include in the estimation, with 1 signalling inclusion and 0 exclusion. Estimates for the excluded domains will be based on the fixed effects model only.
ids	A data frame with `M` rows giving ids for each of the domains. The data frame is copied into the returned object.
contrast.matrix	A matrix of coefficients of contrasts. The matrix must have `TI*NV` rows, but it can contain an arbitrary number of columns. Within each domain, the coefficients are applied to the `TI*NV` EBLUP estimates.
baby.steps	Unless specified as FALSE, the first five iterations of the Fisher scoring algorithm are dampened by factors of `c(.0625, .125, .25, .5, .75)`. These heuristically derived factors appear to lessen drastic overshooting of the true maximum in the initial iterations.

dampening   A factor used to dampen the changes in the random effect parameters. Unlike `baby.steps`, its effect persists during all of the iterations until convergence. Note that the "factory setting" of this parameter is 1 for the dynamic model but .9 for the Rao-Yu model.

iter.history   If TRUE, key values are saved during each iteration and included as additional items, described below, in the returned list: `delta.hist`, `llikelihood.hist`, `adj.hist`, `s.hist`, `ix.hist`, `adj.factor.hist`, and `warning.hist`.

               If `iter.history` is not specified, these items will be returned only if the calculations are begun but not successfully completed.

sig2.min.factor

               A factor to multiply the minimum direct variance to use as a minimum value for any of the variance components. The iterations will be constrained not to go below the resulting bounds.

max.rho_u   A maximum allowed value for the estimated rho_u. The default value is .98.

max.rho   A maximum allowed value for rho. By default, [eblupRY](#) sets this value to .98.

tol   A tolerance value used by matrix routines to prevent numerical instability. The value may be set to a lower value to encourage covergence, but appropriate caution should be applied.

y.rescale   In the univariate case, a scaler multiplier for all of the y values. If the y values are either too small or too large, the information matrix may become unstable. Setting this value to 1 has no effect; setting it to 10 or 100 rescales very small y values to a more appropriate range. Similarly, positive values less than 1 may be used to rescale large y values. The effect of rescaling is removed before normal return from the function, within the limits of normal precision.

               In the multivariate case, `y.rescale` may be a vector of length NV to rescale each component separately.

llike.only   Compute the log-likelihood (ML) or restricted log-likelihood (REML) without further iteration, typically from values specified by `delta`.

method   Use restricted maximum likelihood (″REML″) or maximum likelihood (″ML″).

model   Dynamic (″dyn″) or Rao-Yu (″RY″).

Details

Many of arguments can be used to control the iterations if the defaults lead to convergence difficulties.

`llike.only` in combination with `delta` permits a point-by-point investigation of the likelihood surface.

The primary functions `eblupDyn` and `eblupRY` determine X, NV, `colx`, and `model`, but the remaining parameters can be passed to `dynRYfit` through `eblupDyn` or `eblupRY`.

Value

eblup   In the univariate case, a vector of length M*TI with the eblup estimates in the same sort order as y. In the multivariate case, a matrix of M*TI rows and NV columns.

`fit`	A list summarizing the fit of the model with the following: • `model`: form of the model: TI - Dynamic or RaoYu; REML or ML. • `covergence`: a logical value indicating whether the convergence criterion was met.
`parm`	A labelled vector with the estimated variance components, correlations, and number of iterations.
`coef`	A labelled vector of coefficients for the fixed effects of the model or models.
`ids`	A data frame with D rows and one or more columns of numeric or character domain identifiers.
`delta`	An ordered vector of the variance components (see above). It may be used as starting values for additional iterations.
`eblup.mse`	MSE estimates for eblup.
`eblup.g1`	The g1 term of the MSE estimate.
`eblup.g2`	The g2 term of the MSE estimate.
`eblup.g3`	The g3 term of the MSE estimate.
`est.fixed`	Estimates based on fixed effects only.
`est.fixed.var`	The variance-covariance matrix for the estimates in `coef`.
`eblup.wt1`	Weights given to the direct estimate in forming `eblup`.
`eblup.wt2`	Weights given to the direct estimate, including effects through estimating the fixed effect coefficients.
`contrast.est`	Estimates requested by the specified contrasts.
`contrast.mse`	MSE estimates for `contrast.est`.
`contrast.g1`	The g1 term in the estimation of `contrast.mse`.
`contrast.g2`	The g2 term in the estimation of `contrast.mse`.
`contrast.g3`	The g3 term in the estimation of `contrast.mse`.
`contrast.fixed.est`	
	Contrast estimates based on the fixed effect model.
`contrast.fixed.var`	
	Variance estimates for the fixed effect model.
`contrast.wt1`	Weight wt1 given to the direct estimate in estimating the contrasts.
`contrast.wt2`	Weight wt2 in estimating the contrasts.
`inf.mat`	Information matrix for the components of `delta`.
`var.coef`	Variance covariance matrix for `coef`.
`delta.hist`	Values of `delta` at each iteration.
`llikelihood.hist`	
	Values of the log-likelihood (ML) or restricted log-likelihood (REML) at each iteration.
`adj.hist`	Number of cycles in the internal loop to determine `adj.factor` within each iteration.
`inf.mat.hist`	Values of `inf.mat` at each iteration.

s.hist	Vector to be multiplied by the inverse information matrix to determine the change in the parameters.
ix.hist	List of parameters eligible for change at each iteration. Parameters with estimated changes out of bounds will not be eligible.
adj.factor.hist	
	Adjustment to the vector change in the parameters at each iteration.
warning.hist	A 4-row matrix of warnings at each iteration, where warning 1 is set to 1 for the iteration if the algorithm has not found an increase in the restricted log likelihood or log likelihood, warning 2 is set to 1 if the maximum number of iterations is reached, warning 3 is set to 1 if the estimated variance-covariance matrix becomes singular, and warning 4 is set to 1 if the coefficients of the fixed effects cannot be estimated.

Author(s)

Robert E. Fay, Mamadou Diallo

---

eblupDyn   *EBLUP Fit of the Dynamic and Rao-Yu Time Series Models*

Description

Functions for producing EBLUP small area estimates of the dynamic or Rao-Yu time series models through either ML or REML estimation of the variance components. The functions can fit univariate or multivariate models.

Usage

```
eblupDyn(formula,  D,  TI,  vardir,  method = c("REML", "ML"),
         MAXITER = 1000,  PRECISION = .1e-05,  data,
         max.rho = NULL,  dampening = 1, ...)


eblupRY(formula,  D,  TI,  vardir,  method = c("REML", "ML"),
        MAXITER = 1000,  PRECISION = .1e-05,  data,
        max.rho = .98, dampening = 0.9, ...)
```

Arguments

formula	For a univariate model, a formula for the linear regression relationship between the dependent variable and the independent variable(s). The variables included in formula must have length equal to D*TI and be sorted in ascending order by time within each domain.
	For a multivariate model, a list of formulas, one for each dependent variable. The number of dependent variables, NV, is determined from the length of the list. The dependent variables included in the formulas must each have length equal

	to `D*TI` and be sorted in ascending order by time within each component within each domain, which is an extension of the sorting requirement for the univariate model. Further details of the model specification are given under Details.
D	The total number of domains.
TI	The number of time instances, typically years (constant for all domains).
vardir	For the univariate model, the sampling covariance matrix for the direct estimates of the `D*TI` elements of the dependent variable. The covariance matrix should be in the form of a square matrix with `D*TI` rows and columns. Non-zero covariances between domains are not allowed, so the matrix must have a block diagonal form with `D` blocks, each of which is a square matrix with `TI` rows and columns. Note that within domain, non-zero covariances are allowed over time.
	Alternatively, `vardir` can be a list of `D` covariance matrices, each with `TI` rows and columns.
	For the multivariate model, the square covariance matrix for the `D*NV*TI` elements of the dependent variables. The matrix must be in the form of a square matrix with `D*NV*TI` rows and columns. The variances and covariances should be in the sort order of time within dependent variable within domain. Non-zero covariances between domains are not allowed, but non-zero covariances may be present across time and between components.
	Alternatively, `vardir` can be a list of `D` covariance matrices, each with `NV*TI` rows and columns.
method	Whether restricted maximum likelihood `REML` or maximum likelihood `ML` should be used.
MAXITER	The maximum number of iterations allowed for the Fisher-scoring algorithm.
PRECISION	The convergence tolerance limit for the Fisher-scoring algorithm.
data	An optional data frame containing the variables named in `formula`. By default the variables are taken from the environment from which `eblupDyn` is called. Because `vardir` will be of a different size than the variables in `formula`, `data` will not be searched for `vardir`.
max.rho	If not `NULL`, the maximum value allowed for `rho`. Note the different defaults for `eblupDyn` and `eblupRY`.
dampening	A multiplier of the computed update to parameters after iteration 5. A value less than 1 may slow the iterations but lessens the chance of overshooting the optimum choice. The default values were determined experimentally, but may be modified.
...	Other parameters passed to `dynRYfit` that affect convergence, provide starting values, or request specific results. The exceptions are `y`, `X`, `NV`, `M`, `ncolx`, and `model`, which will be set by either `eblupDyn` or `eblupRY`.

Details

A typical model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` is a series of terms that specifies a linear predictor for response.

A formula has an implied intercept term. To remove this use either `y ~ x - 1` or `y ~ 0 + x`. See formula for more details of allowed formulae.

eblupDyn and eblupRY parse `formula` by calling core R functions to determine X, then calling `dynRYfit`. As a last step, eblupDyn or eblupRY finalize the list that they return.

The additional parameters passed to `dynRYfit` may include `contrast.matrix`, which specifies linear combinations of estimates within domains, such as the sum over dependent variables or moving averages across time. Corresponding MSE estimates are provided for the contrasts.

The argument `ids` accepts a data frame with D rows of domain identifiers. These ids are returned in the list from eblupDyn or eblupRY.

If `iter.history` is set to `TRUE`, the returned object will include additional items with values of statistics at each step of the iteration; see `dynRYfit` for details on `delta.hist`, `llikelihood.hist`, `adj.hist`, `inf.mat.hist`, `s.hist`, `ix.hist`, `adj.factor.hist`, and `warning.hist`. The default action is to include the history only if the iterations fail, in which case the history might suggest what went wrong. In the case of convergence, the history is usually not of interest, in which case omitting it reduces the size of the returned object.

MSE estimation for REML for both the Rao-Yu and dynamic models follows the results summarized in Rao and Molina (2015, pp. 98-111). The MSE estimates incorporate g1, g2, and g3 terms. Our simulations show that the REML estimates have somewhat smaller MSEs than the ML estimates, but this is not reflected in the comparison of the estimated MSEs returned by the functions. The MSE estimates under REML perform quite well on average. The MSE estimates for ML use the same estimator as for REML, but they are modest underestimates of the true MSE in the same simulations.

Value

eblup	In the univariate case, a vector of length D*TI with the eblup estimates. In the multivariate case, a data frame of D*TI rows and NV columns.
fit	A list summarizing the fit of the model with the following:

- model: form of the model: T - Dynamic or Rao-Yu; REML or ML.
- covergence: a logical value indicating whether the convergence criterion was met.
- iterations: number of iterations performed by the Fisher-scoring algorithm.
- estcoef: a data frame with the estimated model coefficients (beta) in the first column , their asymptotic standard errors (std.error) in the second column, the t statistics (tvalue) in the third column, and the p-values (pvalue) of the significance of each coefficient in last column.
- estvarcomp: a data frame with the estimated values of the variances and correlation coefficients in the first column (estimate) and their asymptotic standard errors in the second column (std.error).
- goodness: the log-likelihood and, if REML, the restricted log-likelihood.

parm	A labelled vector with the estimated variance components, correlations, and number of iterations.
coef	A labelled vector of coefficients of the model or models.
ids	A data frame with D rows and one or more columns of numeric or character domain identifiers.

delta	An ordered vector of the variance components, which may be used as starting values for additional iterations, see `dynRYfit`.
eblup.mse	MSE estimates for eblup.
eblup.g1	The g1 term of the MSE estimate.
eblup.g2	The g2 term of the MSE estimate.
eblup.g3	The g3 term of the MSE estimate.
est.fixed	Estimates based on fixed effects only.
est.fixed.var	The variance-covariance matrix for the estimates in `coef`.
eblup.wt1	Weights given to the direct estimate in forming `eblup`.
eblup.wt2	Weights given to the direct estimate, including effects through estimating the fixed effect coefficients.
contrast.est	Estimates requested by the specified contrasts.
contrast.mse	MSE estimates for `contrast.est`.
contrast.g1	The g1 term in the estimation of `contrast.mse`.
contrast.g2	The g2 term in the estimation of `contrast.mse`.
contrast.g3	The g3 term in the estimation of `contrast.mse`.
contrast.fixed.est	
	Contrast estimates based on the fixed effect model.
contrast.fixed.var	
	Variance estimates for the fixed effect model.
contrast.wt1	Weight wt1 given to the direct estimate in estimating the contrasts.
contrast.wt2	Weight wt2 in estimating the contrasts.
inf.mat	Information matrix for the components of `delta`.
var.coef	Variance covariance matrix for `coef`.
model	The formula or list of formulas implemented.

Author(s)

Robert E. Fay, Mamadou Diallo

References

- Fay, R.E. and Herriot, R.A. (1979). Estimation of income from small places: An application of James-Stein procedures to census data. Journal of the American Statistical Association 74, 269-277.

- Fay, R.E., Planty, M. and Diallo, M.S. (2013). Small area estimates from the National Crime Victimization Survey. Proceedings of the Joint Statistical Meetings. American Statistical Association, pp. 1544-1557.

- Rao, J.N.K. and Molina, I. (2015). Small Area Estimation, 2nd ed. Wiley, Hoboken, NJ.

- Rao, J.N.K. and Yu, M. (1994). Small area estimation by combining time series and cross-sectional data. Canadian Journal of Statistics 22, 511-528.

Examples

```
D <- 20 # number of domains
TI <- 5 # number of years
set.seed(1)
data <- data.frame(Y= mvrnormSeries(D=D, TI=TI, rho.dyn=.9, sigma.v.dyn=1,
    sigma.u.dyn=.19, sigma.e=diag(5)), X=rep(1:TI, times=D))
result.dyn  <- eblupDyn(Y ~ X, D, TI, vardir = diag(100), data=data)
result.dyn$fit

require(sae)
data(spacetime)      # Load data set from sae package
data(spacetimeprox)  # Load proximity matrix

D <- nrow(spacetimeprox)           # number of domains
TI <- length(unique(spacetime$Time)) # number of time instants
# Fit model ST with AR(1) time effects for each domain
resultST <- eblupSTFH(Y ~ X1 + X2, D, TI, Var, spacetimeprox,
                      data=spacetime)
resultT  <- eblupDyn(Y ~ X1 + X2, D, TI, vardir = diag(spacetime$Var),
                      data=spacetime, ids=spacetime$Area)
resultT.RY  <- eblupRY(Y ~ X1 + X2, D, TI, vardir = diag(spacetime$Var),
                      data=spacetime, ids=spacetime$Area)
resultST$fit
resultT$fit
resultT.RY$fit
rowsT <- seq(TI, TI*D, by=TI)
data.frame(Domain=spacetime$Area[rowsT], Y=spacetime$Y[rowsT],
              EBLUP_ST=resultST$eblup[rowsT],
              EBLUB_Dyn=resultT$eblup[rowsT],
              EBLUP_RY=resultT.RY$eblup[rowsT])
```

---

geo_ratios	*Compute rates or ratios for a set of geographic entities over a set of years*

Description

The function computes rates or ratios by a geographic code and the variable Year. If designvars is specified, the function also returns a data frame with linear substitutes to compute Taylor series variances.

Usage

```
geo_ratios(data,  geocode,  numerators,  denominators,  geonames,
           new.names,  designvars)
```

Arguments

`data`	A data frame with the required variables, including a variable named `Year`.
`geocode`	A character variable with the name of the geographic variable for which separate estimates are of interest.
`numerators`	A character vector listing the names in `data` of the numerators of the ratios.
`denominators`	A character vector listing the names in `data` of the denominators of the ratios. If a single value is given, it will be used for all of the ratios.
`geonames`	An optional data frame containing geocode and one or more geographic variables, such as names, that will be merged into the results. There should be only one row for each value of geocode.
`new.names`	An optional character vector of the same length as `numerators` naming the resulting ratios. If `new.names` is not specified, the output ratios will have the same names as `numerators`.
`designvars`	Optional. If given, a character vector naming one or more survey design variables in `data` to use in forming linear substitutes for variance calculation of the ratios cross-classified by `Year` and the variable named by geocode. The vector should not include the geocode variable or `Year`.

Details

For programming simplicity, the function enforces the requirement that names should not be repeated in either `numerators` or `new.names`. Names may be repeated in `denominators`.

Rather than a typical survey file, the function expects the data frame `data` to contain weighted estimates for each analytic variable. As a simple example, to find the variance of the weighted mean of y with weights w, `data` should contain w and y $*$ w. For convenience, the weighted estimates can still be assigned their original names in `data`, such as y. In this case,

`numerators = y, denominators=w`

would create the appropriate linear substitutes for the variance of the weighted mean.

This design of the function allows complex possibilities, such as estimating the variance of a rate where the numerator is based on one weight and the denominator is based on another. For example, estimation for the National Crime Victimization Survey requires this capability.

Value

If `designvars` is not specified, a named list with one element, a data frame containing the ratios sorted by geocode and `Year`.

If `designvars` is specified, a second element is added to the list, a data frame giving the totals of the linear substitutes by `Year`, geocode, and `designvars`. The elements of the list are named `estimates` and `linear.subs`.

Author(s)

Robert E. Fay

References

- Woodruff, R.S. (1971). A simple method for approximating the variance of a complex estimate. Journal of the American Statistical Association 66, 411-414.

See Also

vcovgen

Examples

```
require(survey)
require(MASS)
D <- 20 # number of domains
T <- 5 # number of years
samp <- 16 # number of sample cases per domain
set.seed(1)
# use conditional.mean=TRUE to generate true small area values
# without sampling error
Y.list <- mvrnormSeries(D=D, T=T, rho.dyn=.9, sigma.v.dyn=1,
    sigma.u.dyn=.19, sigma.e=diag(5), conditional.mean=TRUE)
# generate sampling errors
e <- rnorm(samp * T * D, mean=0, sd=4)
Y <- Y.list[[2]] + tapply(e, rep(1:100, each=16), mean)
data <- data.frame(Y=Y, X=rep(1:T, times=D))
# model fit with the true sampling variances
result.dyn  <- eblupDyn(Y ~ X, D, T, vardir = diag(100), data=data)
# individual level observations consistent with Y
Y2 <- rep(Y.list[[2]], each=16) + e
data2 <- data.frame(Y=Y2, X=rep(rep(1:T, each=samp), times=D),
                    Year=rep(rep(1:T, each=samp), times=D),
                    weight=rep(1, times=samp*T*D),
                    d=rep(1:D, each=samp*T),
                    strata=rep(1:(D*T), each=samp),
                    ids=1:(D*T*samp))
# geo_ratios with designvars specified
geo.results <- geo_ratios(data2, geocode="d", numerators="Y",
                          denominators="weight",
                          designvars=c("strata", "ids"))
# illustrative check
max(abs(geo.results[[1]]$Y - Y))
vcov.list <- vcovgen(geo.results[[2]], year.list=1:5, geocode="d",
      designvars=c("strata", "ids"))
vcov.list[[1]]
# model fitted with directly estimated variance-covariances
result2.dyn <- eblupDyn(Y ~ X, D, T, vardir=vcov.list, data=data)
cor(result.dyn$eblup, result2.dyn$eblup)
```

---

mvrnormSeries   *Generate data under the Dynamic or Rao-Yu Time Series Models*

Description

Function to generate data under a Rao-Yu time series model, a dynamic model, or a mixture of both. The function can produce either univariate or multivariate observations. All components of the returned random variable have unconditional mean zero. The function calls mvrnorm in **MASS**.

Usage

```
mvrnormSeries(NV=1,  D,  TI,  sigma.e,
    rho.dyn,  sigma.v.dyn,  sigma.u.dyn, rho.u.dyn,
    rho.RY,    sigma.v.RY,    sigma.u.RY,    rho.u.RY,  tol=1e-6,
    conditional.mean=FALSE)
```

Arguments

NV	The number of variables.
D	The number of domains.
TI	The number of time instances (constant for all domains).
sigma.e	The covariance matrix for the variation due to sampling, specified either as a single square matrix with NV*TI rows and columns, or as a list of D matrices, each with NV*TI rows and columns. The rows should vary over TI more quickly than over NV. Sampling covariances between domains are assumed to be zero.
rho.dyn	The temporal correlation parameter in the dynamic model. This parameter is not a true correlation, however, and it may exceed 1.
sigma.v.dyn	A vector of length NV with the v (nu) component of the variance under the dynamic model. This component measures the variance of the random effect at the beginning of the series.
sigma.u.dyn	A vector of length NV with the u component of the variance under the dynamic model.
rho.u.dyn	For NV>1, the cross-sectional correlation(s) under the model. The vector should specify (NV*(NV-1))/2 correlations, in lower triangular form. For example, for NV=3, the correlations should be specified in the order (2,1), (3,1), (3,2).
rho.RY	The temporal correlation parameter in the Rao-Yu model. This is a true correlation, unlike the parameter in the dynamic model.
sigma.v.RY	A vector of length NV with the v (nu) component of the variance under the Rao-Yu model. This component reflects a constant random effect for each domain unchanging over time.
sigma.u.RY	A vector of length NV with the u component of the variance under the Rao-Yu model.

rho.u.RY	For NV>1, the cross-sectional correlation under the model. The vector should specify (NV*(NV-1))/2 correlations, in lower triangular form. For example, for NV=3, the correlations should be specified in the order (2,1), (3,1), (3,2).
tol	A tolerance parameter used by mvrnorm in **MASS** to determine if the covariance matrix is non-singular.
conditional.mean	
	If true, the function will also return the generated values of the random effects.

Details

The function assembles the covariance matrix from the covariance matrix under the dynamic model (if specified), the Rao-Yu model (if specified) and a required sampling covariance matrix.

Value

If conditional.mean=FALSE, then for NV=1, a multivariate normal random vector with mean zero and length D*TI. For NV>1, a matrix with D*TI rows and NV columns.

If conditional.mean=TRUE, a list with the first element as above and a second element that is the sum of the random effects without the sampling error. Simulation studies can evaluate the small area estimates using the first element of the list as input against the second element of the list, which is the target of the small area estimation.

Author(s)

Robert E. Fay

See Also

[mvrnorm](#)

Examples

```
set.seed(7)
mvrnormSeries(D=2, TI=5, sigma.e=diag(5), rho.dyn=.8,
    sigma.v.dyn=2, sigma.u.dyn=.72, conditional.mean=TRUE)
mvrnormSeries(NV=2, D=2, TI=5, sigma.e=diag(10), rho.dyn=.8,
    sigma.v.dyn=2, sigma.u.dyn=.72, rho.u.dyn=.8)
```

| vcovgen | *Estimate variance/covariance matrices using linear substitutes* |

Description

The function computes estimates of variance/covariance matrices for sampling with replacement at the first stage from a stratified sample, using the linear substitutes produced by [geo_ratios](#). The function produces a list of estimated covariance matrices corresponding the geographic areas identified by geocode.

Usage

```
vcovgen(linear.subs, year.list, geocode, designvars)
```

Arguments

linear.subs	A data frame with linear substitutes for each Y variable, Year, and the variables with the names given by geocode and designvars.
year.list	A character or numeric vector of the years to include.
geocode	A character variable with the name of the geographic variable for which separate estimates are of interest.
designvars	A character vector of length 2, naming the strata and first-stage units.

Details

The function reformats the second element of the list output from geo_ratios in order to estimate covariances across time for each separate geographic area identified by the variable with the name indicated by geocode. It then calls functions from the **survey** package to estimate covariances.

See the example in geo_ratios.

Value

A list of estimated covariance matrices, sorted by the geographic names. The list may be used by eblupDyn or eblupRY, although many applications require the estimated variance/covariance matrices to be smoothed.

Author(s)

Robert E. Fay

Index