Package 'sae2'

February 20, 2015

Type Package

Title Small Area Estimation: Time-series Models

Version 0.1-1

Date 2015-01-26

Author Robert E. Fay, Mamadou Diallo

Maintainer Robert Fay <bobfay@hotmail.com>

Depends R (>= 2.14.0), MASS

Suggests sae

Description Time series models for small area estimation based on area-level models.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-26 12:23:20

R topics documented:

`sae2-package` *Small Area Estimation: Time-series Models.*

Description

Time series models to supplement the functionality of the package `sae`, specifically EBLUP fitting of the original Rao-Yu model and also of a modified ("dynamic") version. Both univariate and multivariate applications are supported. The models are non-spatial, in contrast to the spatial models in `sae`.

Details

Package:	sae2
Type:	Package
Version:	0.1-1
Date:	2015-01-26
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.

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 Victimization Survey. Proceedings of the Joint Statistical Meetings. American Statistical Association, pp. 1544-1557.

- Rao, J.N.K. (2003). Small Area Estimation. Wiley, New York.

- 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`

`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, T, vardir, method = c("REML","ML"),
         MAXITER = 1000, PRECISION = .1e-05, data, ...)

eblupRY(formula, D, T, vardir, method = c("REML","ML"),
         MAXITER = 1000, PRECISION = .1e-05, data, ...)
```

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*T 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*T and be sorted in ascending order by time within each component within each domain, which is the same sorting requirement as for the univariate model. Further details of the model specification are given under Details.
D	The total number of domains.
T	The number of time instances (constant for all domains).
vardir	For the univariate model, the sampling covariance matrix for the direct estimates of the D*T elements of the dependent variable. The covariance matrix should be in the form of a square matrix with D*T 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 T 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 T rows and columns. For the multivariate model, the square covariance matrix for the D*NV*T elements of the dependent variables. The matrix must be in the form of a square matrix with D*NV*T 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*T 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, with a default value of 100.
PRECISION	The convergence tolerance limit for the Fisher-scoring algorithm, with a default value of .000001.
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.
...	Other parameters passed to reml.dyn, mle.dyn, reml.Rao.Yu or mle.Rao.Yu.

Details

A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

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 functions within R, then calling one of the functions reml.dyn, mle.dyn, reml.Rao.Yu or mle.Rao.Yu. As a last step, eblupDyn and eblupRY finalize the returned list.

The additional parameters passed to reml.dyn etc. include contrast.matrix, which specifies linear combinations of estimates within domains, such as the sum over dependent variables or across time. Corresponding MSE estimates are provided for the contrasts. Another argument is ids, which accepts a data frame with D rows of domain identifiers that is included in the list returned by eblupDyn or eblupRY. Other parameters affect convergence or provide starting values. If iter.history is set to TRUE, the returned object will include more items with values of statistics at each step of the iteration; see reml.dyn for details.

MSE estimation for REML for both the Rao-Yu and dynamic models follows the results summarized in Rao (2003, pp. 98-105). 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*T with the eblup estimates. In the multivariate case, a data frame of D*T 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.
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.

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. (2003). Small Area Estimation. Wiley, New York.

- 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
T <- 5 # number of years
set.seed(1)
data <- data.frame(Y= mvrnormSeries(D=D, T=T, rho.dyn=.9, sigma.v.dyn=1,
   sigma.u.dyn=.19, sigma.e=diag(5)), X=rep(1:T, times=D))
result.dyn  <- eblupDyn(Y ~ X, D, T, 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
T <- length(unique(spacetime$Time)) # number of time instants
# Fit model ST with AR(1) time effects for each domain
resultST <- eblupSTFH(Y ~ X1 + X2, D, T, Var, spacetimeprox,
                      data=spacetime)
resultT  <- eblupDyn(Y ~ X1 + X2, D, T, vardir = diag(spacetime$Var),
                      data=spacetime, ids=spacetime$Area)
resultT.RY  <- eblupRY(Y ~ X1 + X2, D, T, vardir = diag(spacetime$Var),
                      data=spacetime, ids=spacetime$Area)
resultST$fit
resultT$fit
resultT.RY$fit
rowsT <- seq(T, T*D, by=T)
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])
```

`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, T, 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)
```

Arguments

`NV`	The number of variables.
`D`	The number of domains.

T	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*T rows and columns, or as a list of D matrices, each with NV*T rows and columns. The rows should vary over T 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.

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

For NV=1, a multivariate normal random vector with mean zero and length D*T. For NV>1, a matrix with D*T rows and NV columns.

Author(s)

Robert E. Fay

See Also

mvrnorm

Examples

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

`reml.dyn` *Internal fitting functions for Dynamic and Rao-Yu models*

Description

Functions called by `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. The functions can fit univariate or multivariate models.

Usage

```
reml.dyn(y, X, M, T, NV=1, vcov_e, maxiter=100, iter.tol=.1e-5,
         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=1, iter.history=FALSE,
         sig2.min.factor=.0001, max.rho_u=.98, tol=.Machine$double.eps,
         y.rescale=NULL)

mle.dyn(y, X, M, T, NV=1, vcov_e, maxiter=100, iter.tol=.1e-5,
         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=1, iter.history=FALSE,
         sig2.min.factor=.0001, max.rho_u=.98, tol=.Machine$double.eps,
         y.rescale=NULL)

reml.Rao.Yu(y, X, M, T, NV=1, vcov_e, maxiter=100, iter.tol=.1e-5,
         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=.9, iter.history=FALSE,
         sig2.min.factor=.0001, max.rho_u=.98, max.rho=.98,
         tol=.Machine$double.eps, y.rescale=NULL)

mle.Rao.Yu(y, X, M, T, NV=1, vcov_e, maxiter=100, iter.tol=.1e-5,
         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=.9, iter.history=FALSE,
         sig2.min.factor=.0001, max.rho_u=.98, max.rho=.98,
         tol=.Machine$double.eps, y.rescale=NULL)
```

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.
T	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*T elements of the dependent variable. The covariance matrix should be in the form of a square matrix with M*T 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 T 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*T the dependent variables. The matrix should be in the form of a square matrix with M*NV*T 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 components.
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.
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 (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 rho, sig2_u, sig2_v, or rho_u.
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 domain will be based on the fixed effect model only.

`ids` A data frame with `M` rows giving ids for each of the domains. The data frame is copied to the returned object.

`contrast.matrix`
A matrix of coefficients of contrasts. The matrix must have `T*NV` rows, but it can contain an arbitrary number of columnts. Within each domain, the coefficients are applied to the `T*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 of the iterations and included as additional items in the returned list:

- `delta.hist:` values of `delta` at each step.
- `likelihood.hist:` values of the log-likelihood (for ML) or restricted log-likelihood (for REML) at each step.
- `adj.hist:` adjustment at each step to prevent violation of boundary constraints.
- `s:` value of `s` used to compute next step in the iteration.
- `ix.hist:` components of `delta` subject to alteration in this step.
- `adj.factor:` the final multiplier of the iteration step.
- `warning.hist:` an indicator of warning messages generated at each step.

`sig2.min.factor`
A factor to multiply the minimum direct variance to set a minimum value for any of the variance components.

`max.rho_u` A maximum allowed value for the estimated `rho_u`.

`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` A scaler multiplier for all of the `y` values. If the `y` values are either too small or two 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.

`max.rho` For the Rao-Yu model only, a maximum allowed value for `rho`.

Details

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

of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed.

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.

The additional parameters include contrast.matrix, which specifies linear combinations of estimates within domains, such as the sum over components or across time. Corresponding MSE estimates are provided.

Other parameters affect convergence or provide starting values.

Value

eblup	In the univariate case, a vector of length M*T with the eblup estimates. In the multivariate case, a matrix of M*T rows and NV columns.
fit	A list summarizing the fit of the model with the following: • model: form of the model: T - 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 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.
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`.

Author(s)

Robert E. Fay, Mamadou Diallo

Index