Package 'mind'

October 27, 2022

Type Package

Title Multivariate Model Based Inference for Domains

Version 1.1.0

Depends R (>= 4.1.0), data.table, MASS, Matrix

Imports dplyr,stats, JWileymisc, tm, methods

Description Allows users to produce estimates and MSE for multivariate variables using Linear Mixed Model. The package follows the approach of Datta, Day and Basawa (1999) <doi:10.1016/S0378-3758(98)00147-5>.

License EUPL

Encoding UTF-8

NeedsCompilation no

Author Michele D'Alo' [aut],
Stefano Falorsi [aut],
Andrea Fasulo [aut, cre]

Maintainer Andrea Fasulo <fasulo@istat.it>

BuildResaveData best

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2022-10-27 11:32:36 UTC

R topics documented:

benchSAE *Benchmark for SAE*

Description

Benchmarked values based on ratio adjustment of the SAE estimates

Usage

```
benchSAE(estim_sae,benchmark_area,area,name_dom,estimator,Nest)
```

Arguments

estim_sae	a data frame containing the arguments area, name_dom, estimator and Nest
benchmark_area	a data frame identifing the area to be benchmarked among with the benchmark value
area	character, identified the name of the area to be benchmarked
name_dom	character, identified the name of the domains
estimator	vector, contain the name of the estimates to be benchmarked
Nest	character, identified the name of the total population size for domain

Details

The benchSAE function allows (i) to benchmark more than one SAE estimates at times and (ii) to specified the broadarea to be benchmarked, from the national level to a more disaggregated level.

Value

benchSAE produces a data.frame with a column of domain indicator and set of columns (as specified in estimator) of benchmarked point predictions.

Author(s)

Developed by Michele D'Alò

References

Datta,G. S., Ghosh, M., Steorts, R. and Maples, J. (2010) Bayesian benchmarking with applications to small area estimation. Test, 20, 574–588.

Examples

```
# Load example data

data(data_s);data(univ)

tot<-aggregate(tot~dom+pro,univ,sum)

# One random effect at domain level


  formula<-as.formula(cbind(emp,unemp,inact)~(1|mun)+
                        factor(sexage)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea

  univ_1<-univ[,-6]

# Estimate mind model benchmarking the unemployment variable

  example.1<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_1,MSE=FALSE)
  SAEest<-example.1$EBLUP[,c(1,3)]
  SAEest<-merge(SAEest,tot)
  SAEest$unemp<-SAEest$unemp/SAEest$tot
  bench<-data.frame(pro=c(10,11),bb=c(0.27,0.31))

# Benchmark the unemployment point estimates

  SAEest_bench<-benchSAE(estim_sae=SAEest,
                         benchmark_area=bench,
                         area="pro",
                         name_dom="dom",
                         estimator="unemp",
                         Nest="tot")
```

---

data_s *Synthetic sample dataset for Multivatiate Linear Mixed Model*

Description

Synthetic data frame containing a sample of 9962 individuals along with socio-economic indicators.

Usage

```
data(data_s)
```

Format

A data frame with 9962 observations on 10 variables:

`dom` domain of interest codes, corresponding to the munipal codes

`emp` binary variable, 1 for employed 0 otherwise

`unemp` binary variable, 1 for unemployed 0 otherwise

`inact` binary variable, 1 for inactive 0 otherwise

`sexage` cross classification of age and sex

`edu` educational level

`fore` binary variable, 2 for foreigner 1 otherwise

`mun` municipal codes

`pro` provincial codes

`occ_stat` occupational status, 1 for employed 2 for unemployed 3 for inactive

Details

The informations on the sample unit are the same collected in the syntethic population dataframe univ appart from the information on the occupational status that are present only for the sample units.

Examples

```
# Load example data
data(data_s)
summary(data_s)
```

`mind.unit` *Fitting Unit level Multivariate Linear Mixed Model*

Description

`mind.unit` is used to fit unit level multivariate linear mixed models [D'Alo', Falorsi 2021, FAO 2021]. It can be used to carry out different estimators (EBLUP, Synthetic and Projection) and the Mean Squared Error (MSE) for unplanned domain, analysis of the random effect and study of the variance components.

Usage

```
mind.unit(formula,dom,data,universe,weights=NA,broadarea=NA,
max_iter=200,max_diff=1e-05,phi_u0=0.05,MSE=TRUE,REML=TRUE)
```

Arguments

`formula`	an object of class "formula": a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`dom`	numeric, the domain of interest. See also 'Details'.
`data`	a data frame containing the variables in the model, e.g. `data_s`.

universe	a data frame containing the complete list of the units belonging to the target population, along with the corresponding values of the auxiliary variables. Also an aggregated version of the universe information is possible, e.g. univ. See also 'Details'.
weights	an optional column of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used with weights; otherwise ordinary least squares is used. See also 'Details'.
broadarea	an optional character to be used if a broadarea is required in the model. See also 'Details'.
max_iter	integer scalar. Numeber of maximum iteration for the optimization of the REML criterion (default=200) .
max_diff	double number. Stopping criteria to be satisfied to achieve the REML convergence (defualt=1e-05).
phi_u0	double number. Initialization value for the ratio among the variance components effect and the variance of the errors [Saei, Chambers 2003] (defualt=0.05)
MSE	logical scalar. Should the MSE be computed (defualt=TRUE)?
REML	logical scalar. Should the estimates be chosen to optimize the REML criterion (as opposed to the maximum-likelihood)?

Details

A typical predictor for a Multivariate Linear Mixed Model has the form `responses ~ random.terms+fixed.terms` where `responses` is the multivariate response, `random.terms` is a series of terms which specifies random intercept and `fixed.terms` is a series of terms which specified a linear predictor for `responses`.
The `responses` can be specified as a column (so the `responses` have m different values as the modalities are) or as a m-column with the columns giving the presence and absence of the modalities (using `cbind` function).
The `random.terms` in the formula will be re-ordered when both domain and marginal effect are presents so that domain effects come first, followed by the marginal. The `random.terms` must be `numeric` varaibles.
In the actual version of `mind` (i) only qualitative `fixed.terms` are allowed.

The mandatory argument `dom` must be numeric and must not contain any missing value (NA).

The mandatory argument `universe` is a `data.frame` conteining the auxiliary information referenced in the `formula` for each unit of the population of interest.
For computational reason it is possible use an aggregated version of the population information using the profile derived by the `random.terms` and `fixed.terms`. In this case a column equal to the summation of the population units for each profile is required.
See univ for more details.

Non-`NULL` `weights` can be used to indicate that different observations have different variances. If no `weights` are specified all the units have an unitary weight. If specified must be present in `data`.

`broadarea` represents the grupping factor specifying the partitioning of the data. If non-`NULL` `broadarea` is includes different `mind.unit` fits should be performed according to `broadarea`. Must be present in `universe`.

Value

`mind.unit` returns an object of class "`mind`". The object is a list contains 13 objects:

`EBLUP` a data frame containing for the domain of interest the EBLUP estimates [Rao, Molina 2015] for the m-modalities of the responce variable.

`PROJ` a data frame containing for the domain of interest the Projection estimates [Kim, Rao 2011] for the m-modalities of the responce variables.

`SYNTH` a data frame containing for the domain of interest the Synthetic estimates [Rao, Molina 2015] for the m-modalities of the responce variable.

`mse_EBLUP` a data frame containing for the domain of interest the MSE, along with the single components G1, G2 and G3, for the EBLUP estimator for the m-modalities of the variables of interest.

`cv_EBLUP` a data frame containing the coefficent of variation for the EBLUP estimator for the m-modalities of the variable of interest.

`Nd` a data frame with the total population of the domain of interest.

`nd` a data frame with the sample size of the sampled domain of interest.

`r_effect` a list containing the random effects for each modes of the `responses` and for each `broadarea` (if any).

`beta` a data frame with named columns of coefficents.

`mod_performance` a list containing fit indices, absolute error metrics, tests of overall model significance (taking into account only the `fixed.terms`) for each modes of the `responses` and for each `broadarea` (if any).

`sigma_e` a data frame with the residuals standard deviation σ_e for each modes of the `responses` and for each `broadarea` (if any).

`sigma_u` a data frame with the random effects standard deviation σ_u for each modes of the `responses`, for each random effect and for each `broadarea` (if any).

`ICC` a data frame with the Intraclass Coefficent Correlation for each modes of the `responses`, for each random effect and and for each `broadarea` (if any). The population ICC in this framework is:

$$ICC = \frac{\sigma_u^2}{(\sigma_u^2 + \sigma_e^2)}$$

This ICC can be generalized to allow for covariate effects, in which case the ICC is interpreted as capturing the within-class similarity of the covariate-adjusted data values.

Author(s)

Developed by Michele D'Alo', Stefano Falorsi, Andrea Fasulo

References

Battese, G., E., Harter, R., M., Fuller, W., A., (1988). 'An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data', Journal of the American Statistical Association Vol. 83, No. 401 (Mar., 1988), pp. 28-36.

Datta, G., S., Day, B., Basawa, I., (1999). 'Empirical best linear unbiased and empirical Bayes prediction in multivariate small area estimation', Journal of Statistical Planning and Inference, Volume 75, Issue 2, 1 January 1999, Pages 269-279

D'Alo', M., Falorsi, S., Fasulo, A., (2021). 'MIND, an R package for multivariate small area estimation with multiple random effects', SAE2021 BIG4small. Book of short papers, Pages 43-48

ESSnet on SAE, (2012). 'Guidelines for the application of the small area estimation methods in NSI sample surveys'

FAO (2021). 'Guidelines on data disaggregation for SDG Indicators using survey data', pp. 105. http://www.fao.org/publications/card/en/c/CB3253EN/

Harmening, S., Kreutzmann, A.K., Pannier, S., Salvati, N., Schmid, T., (2021). 'A Framework for Producing Small Area Estimates Based on Area-Level Models in R, The R package emdi vignette'

Kim, J. K., Rao, J. N., (2011). 'Combining data from two independent surveys: a model-assisted approach', Biometrika 99(1), 85100.

Rao, J.N., Molina, I., (2015). 'Small Area Estimation', John Wiley & Sons

Saei, A., Chambers, R., (2003). 'Small Area Estimation Under Linear and Generalized Linear Mixed Models With Time and Area Effects', S3RI Methodology Working Paper M03/15

See Also

predict.mind has examples of fitting multivariate variables.

Examples

```
# Load example data
data(data_s);data(univ)

# The sample units cover 104 over 333 domains in the population data frame
length(unique(data_s$dom));length(unique(univ$dom))

## Example 1
# One random effect at domain level
# Double possible formulations
formula<-as.formula(occ_stat~(1|mun)+
factor(sexage)+factor(edu)+factor(fore))
#or
formula<-as.formula(cbind(emp,unemp,inact)~(1|mun)+
factor(sexage)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea
univ_1<-univ[,-6]

example.1<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_1)
summary(example.1$EBLUP)
rm(univ_1)
```

```
## Example 2
# One random effect for a marginal domain
formula<-as.formula(occ_stat~(1|pro)+factor(sexage)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea
univ_2<-univ[,-5]

example.2<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_2)
summary(example.2$EBLUP)
rm(univ_2)

## Example 3
# Two random effects both at domain level and marginal level
formula<-as.formula(occ_stat~(1|mun)+(1|pro)+
factor(sexage)+factor(edu)+factor(fore))

example.3<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ)
summary(example.3$EBLUP)

## Example 4
# One random effect at domain level and with broadarea
formula<-as.formula(occ_stat~(1|mun)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea
univ_4<-univ[,-2]

example.4<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_4,broadarea="pro")
summary(example.4$EBLUP)
rm(univ_4)
```

`predict.mind` *Predict method for Multivariate Linear Mixed Model*

Description

Predicted values based on Multivariate Linear Mixed Model object

Usage

```
## S3 method for class 'mind'
predict(object,data,type= "proj",dir_s=NULL,dir_cov=NULL,...)
```

Arguments

`object`	an object of class "`mind`"
`data`	an object in which to look with which to predict.

type	the type of prediction required. ; the predictors type are the "eblup", "proj" and "synth".The default to "proj".
dir_s	optionally, if type is equal to eblup a data frame with the count for each modalities of the responce variable for the covariate patterns must be provided.
dir_cov	optionally, if type is equal to eblup a data frame with the sample units count for the covariate patterns must be provided.
...	arguments based from or to other methods

Details

predict.mind produces predicted values, obtained by means the regression parameters on the frame data. In the actual version of predict.mind only unit level predictions are provided. If the type is equal to proj or synth a data.frame with individual predictions for all the modalities of the responce variable will be produced. When the eblup predictor is chosen two more input data.frame must be provided and the function will produce predictions for all the profiles identified by the cross-classification of the domains and covariate patterns.

Value

When the predictor type is set equal to proj or synth the predict.mind produces a data.frame of predictions with the columns name equal to the multivariate responses. If the predictor type is eblup a data frame with predictions for all the profile (cross-clafficiation of domain of interest and coavariate patterns) is provided.

Author(s)

Developed by Andrea Fasulo

Examples

```
# Load example data
data(data_s);data(univ)

# The sample units cover 104 over 333 domains in the population data frame
length(unique(data_s$dom));length(unique(univ$dom))


# One random effect at domain level
formula<-as.formula(cbind(emp,unemp,inact)~(1|mun)+
factor(sexage)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea
univ_1<-univ[,-6]

example.1<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_1)

## Example 1
#Projection predictions
example.1.predict<-predict.mind(object=example.1,data=univ,type= "proj")
```

```
# Check if the sum of the unit level predictions at
# domain level are equal to the mind.unit Projection predictions
ck<-cbind(univ,example.1.predict)
ck<-aggregate(cbind(emp,unemp,inact)~dom,ck,sum)
head(ck);head(example.1$PROJ)

## Example 2
#Synthetic predictions
example.1.synth<-predict.mind(object=example.1,data=univ,type="synth")

# Check if the sum of the unit level predictions at
# domain level are equal to the mind.unit Synthetic predictions
ck<-cbind(univ,example.1.synth)
ck<-aggregate(cbind(emp,unemp,inact)~dom,ck,sum)
head(ck);head(example.1$SYNTH)

## Example 3
#EBLUP predictions
inp_1<-aggregate(cbind(emp,unemp,inact)~dom+mun+sexage + edu + fore,data_s,sum)

inp_2<-aggregate(emp+unemp+inact~dom+mun+sexage+edu +fore,data_s,sum)

example.1.eblup<-predict.mind(object=example.1,data=univ_1,type="eblup",dir_s=inp_1,dir_cov=inp_2)

# Check if the sum of the predictions at
# profile level are equal to the mind.unit Eblup predictions
ck<-aggregate(cbind(emp,unemp,inact)~dom,example.1.eblup,sum)
head(ck);head(example.1$EBLUP)
```

`univ` *Synthetic population dataset for Multivatiate Linear Mixed Model*

Description

Synthetic population data frame containing the complete list of the units belonging to the target population along with the corresponding values of the auxiliary variables.

Usage

```
data(univ)
```

Format

A data frame with 514320 observations on 7 variables:

`dom` domain of interest codes

`sexage` cross classification of age and sex

`edu` educational level

`fore` bynary variable, 2 for foreigner 1 otherwise

`mun` municipal codes

`pro` provincial codes

`tot` column of 1

Details

The informations on the population are the same collected in the syntethic sample data_s appart from the information on the occupational status that are present only for the sample units.

`mind.unit` allows to use a data frame of known population totals based on the marginal distribution of the profile identified by the auxiliary variables (See 'Examples').

Examples

```
library(dplyr)

# Load example data
data(data_s);data(univ)
summary(univ)


formula<-as.formula(occ_stat~(1|pro)+factor(sexage)+factor(edu)+factor(fore))

# Drop from the universe data frame variables not referenced in the formula or in the broadarea
univ_1<-univ[,-5]

# 1) Estimation using the complete list of the unit beloging the target population:
example.1<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_1)
rm(univ_1)

# Creation of the know population totals object:
univ_ag<-aggregate(tot~-1+factor(dom)+factor(pro)+
factor(sexage)+factor(edu)+factor(fore),univ,sum)
colnames(univ_ag)<-c("dom","pro","sexage","edu","fore","tot")

# Set all variables as numeric.
#Remember that only the domains codes and the random terms must to be numeric variables.
univ_ag <- mutate_all(univ_ag, function(x) as.numeric(as.character(x)))

# 2) Estimation using the know population totals (totals in univ_ag) :
example.2<-mind.unit(formula=formula,dom="dom",data=data_s,universe=univ_ag)
```

Index