Package ‘JoSAE’

February 15, 2013

Type Package

Title Functions for unit-level small area estimators and their variances

Version 0.2

Date 2011-07-14

Author Johannes Breidenbach

Maintainer Johannes Breidenbach <job@skogog1andskap.no>

Description This package currently implements the unit level EBLUP and
GREG estimators as well as the estimate of their variances to
further document the publication of Breidenbach and Astrup
(2011). It also contains a vignette that explains the use of the implemented functions.

License GPL-2

LazyLoad yes

Depends nlme

Suggests rsae,lattice,xtable

Repository CRAN

Date/Publication 2011-10-11 15:49:59

NeedsCompilation no

R topics documented:

JoSAE-package .................................................. 2
eblup.mse.f .................................................. 6
JoSAE.domain.data ......................................... 8
JoSAE.sample.data .......................................... 9

Index 10

1
JoSAE-package

Provides functions for small area estimation and the estimate of variances (mean squared errors).

Description

This package currently implements the unit level EBLUP (Battese et al. 1988) and GREG (Sarndal 1984) estimators as well as their variance estimators. It also contains data and a vignette that explain its use.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>JoSAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>0.2</td>
</tr>
<tr>
<td>Date:</td>
<td>2011-07-14</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

The aim in the analysis of sample surveys is frequently to derive estimates of subpopulation characteristics. Often, the sample available for the subpopulation is, however, too small to allow a reliable estimate. If an auxiliary variable exists that is correlated with the variable of interest, small area estimation (SAE) provides methods to solve the problem (Rao 2003).

The purpose of this package is primarily to document the functions used in the publication of Breidenbach and Astrup (2011). The data used in that study are also provided.

You might wonder why this package is called JoSAE. Well, first of all, JoSAE sounds good. The other problem was that the packages SAE and SAE2 already exist (Gomez-Rubio, 2008). They are, however, not available on CRAN and unmaintained (as of July 2011). They also do not seem to implement the variance estimators that we needed. So I just combined SAE with the first part of my name.

Note

All the implemented functions/estimators are described in Rao (2003). This package merely makes the use of the estimators easier for the users that are not keen on programming. Especially the EBLUP variance estimator would require some effort.

After uploading the first JoSAE version (0.1), I realized there exists another SAE package that specializes on robust SAE (rsae; Schoch, 2011). It implements reampling variance estimators as opposed to analytical variance estimators implemented here. The rsaE package comes with a very good vignette as well as the Landsat data set used by Battese et al. (1988).

There are several points where the JoSAE could be improved:

Only univariate unit-level models with a simple block-diagonal variance structure are supported so far.
The computation is based on loops on the domain level. It would be more elegant to use blocked matrices.
... many more things that will hopefully improve once I have more experience with programming R-packages ...

Author(s)
Johannes Breidenbach
Maintainer: Johannes Breidenbach <job@skogoglandskap.no>

References
Gomez-Rubio (2008), Tutorial on small area estimation, UseR conference 2008, August 12-14, Technische Universitat Dortmund, Germany.

See Also
eblup.mse.f, JoSAE.sample.data, JoSAE.domain.data

Examples
#mean auxiliary variables for the populations in the domains
data(JoSAE.domain.data)
#data for the sampled elements
data(JoSAE.sample.data)
plot(biomass.ha~mean.canopy.ht,JoSAE.sample.data)

## the easy way: use the wrapper function to compute EBLUP and GREG estimates and variances
#lme model
summary(fit.lme <- lme(biomass.ha ~ mean.canopy.ht, data=JoSAE.sample.data
                          , random=~1|domain.ID))

  #domain data need to have the same column names as sample data or vice versa
d.data <- JoSAE.domain.data
names(d.data)[3] <- "mean.canopy.ht"
result <- eblup.mse.f.wrap(domain.data = d.data, lme.obj = fit.lme)
result
## END: the easy way

## the hard way: compute the EBLUP MSE components yourself

### get an overview of the domains

- Mean of the response and predictor variables from the sample. For the response this is the sample mean:
  ```r
tmp <- aggregate(JoSAE.sample.data[,c("biomass.ha", "mean.canopy.ht")],
                 by=list(domain.ID=JoSAE.sample.data$domain.ID), mean)
  names(tmp)[2:3] <- paste(names(tmp)[2:3], ".bar.sample", sep="")
```

- Number of samples within the domains:
  ```r
tmp1 <- aggregate(cbind(n.i=JoSAE.sample.data$biomass.ha),
                 by=list(domain.ID=JoSAE.sample.data$domain.ID), length)
```

- Combine it with the population information of the domains:
  ```r
overview.domains <- cbind(JoSAE.domain.data, tmp[,-1], n.i=tmp1[,-1])
```

### fit the models

- **lm** - the auxiliary variable explains forest biomass rather good
  ```r
summary(fit.lm <- lm(biomass.ha ~ mean.canopy.ht, data=JoSAE.sample.data))
```

- **lme**
  ```r
summary(fit.lme <- lme(biomass.ha ~ mean.canopy.ht, data=JoSAE.sample.data,
                         random=~1|domain.ID))
```

- Mean lm residual -- needed for GREG
  ```r
overview.domains$mean.resid.lm <- aggregate(resid(fit.lm),
                                            by=list(domain.ID=JoSAE.sample.data$domain.ID),
                                            mean)[,2]
```

- Mean lme residual -- needed for EBLUP.var
  ```r
overview.domains$mean.resid.lme <- aggregate(resid(fit.lme, level=0),
                                             by=list(domain.ID=JoSAE.sample.data$domain.ID),
                                             mean)[,2]
```

### synthetic estimate

```r
overview.domains$synth <- predict(fit.lm,
                                  newdata=data.frame(mean.canopy.ht=JoSAE.domain.data$mean.canopy.ht.bar))
```

### GREG estimate

```r
overview.domains$GREG <- overview.domains$synth + overview.domains$mean.resid.lm
```

### EBLUP estimate

```r
overview.domains$EBLUP <- predict(fit.lme,
                                   newdata=data.frame(mean.canopy.ht=JoSAE.domain.data$mean.canopy.ht.bar,
                                                       domain.ID=JoSAE.domain.data$domain.ID),
                                   level=1)
```

### gamma
overview.domains$gamma.i <- eblup.mse.f.gamma.i(lme.obj=fit.lme,
  n.i=overview.domains$n.i)

#variance of the sample mean estimate
overview.domains$sample.var <-
  aggregate(JoSAE.sample.data$biomass.ha,
    by=list(domain.ID=JoSAE.sample.data$domain.ID), var)[-1]/overview.domains$n.i

#variance of the GREG estimate
overview.domains$GREG.var <-
  aggregate(resid(fit.lm),
    by=list(domain.ID=JoSAE.sample.data$domain.ID), var)[-1]/overview.domains$n.i

#variance of the EBLUP
#compute the A.i matrices for all domains (only needed once)
domain.ID <- JoSAE.domain.data$domain.ID
#initialize the result vector
a.i.mats <- vector(mode="list", length=length(domain.ID))
for(i in 1:length(domain.ID)){
  print(i)
  a.i.mats[[i]] <- eblup.mse.f.c2.ai(gamma.i=overview.domains$gamma.i[overview.domains$domain.ID==domain.ID[i]],
    n.i=overview.domains$n.i[overview.domains$domain.ID==domain.ID[i]],
    lme.obj=fit.lme
    X.i=as.matrix(cbind(i=1,
    x=JoSAE.sample.data[JoSAE.sample.data$domain.ID==domain.ID[i], "mean.canopy.h]
))
}

#add all the matrices
sum.A.i.mats <- Reduce("+", a.i.mats)

#the assymptotic var-cov matrix
asy.var.cov.mat <- eblup.mse.f.c3.asyvarcovarmat(n.i=overview.domains$n.i,
  lme.obj=fit.lme)

#put together the variance components
###### Some changes are required here, if you apply it to own data!
result <- NULL
for(i in 1:length(domain.ID)){
  print(i)
  #first comp
  mse.c1.tmp <- eblup.mse.f.c1(gamma.i=overview.domains$gamma.i[overview.domains$domain.ID==domain.ID[i]],
    n.i=overview.domains$n.i[overview.domains$domain.ID==domain.ID[i]],
    lme.obj=fit.lme)
  #second comp
  mse.c2.tmp <- eblup.mse.f.c2(gamma.i=overview.domains$gamma.i[overview.domains$domain.ID==domain.ID[i]],
    X.i=as.matrix(cbind(i=1!!cbind(),
    x=JoSAE.sample.data[JoSAE.sample.data$domain.ID==domain.ID[i], "mean.canopy.h"]!!cbind(),
    x=JoSAE.domain.data[JoSAE.domain.data$domain.ID==domain.ID[i], "mean.canopy.h"]!!cbind()))
    x.bar.i =as.matrix(rbind(i=1!!rbind(),
    x=JoSAE.domain.data[JoSAE.domain.data$domain.ID==domain.ID[i], "mean.canopy.h"]!!rbind(),
    x=JoSAE.domain.data[JoSAE.domain.data$domain.ID==domain.ID[i], "mean.canopy.h"]!!rbind()))

  result <- c(result, mse.c1.tmp, mse.c2.tmp)
  print(result)
}
eblup.mse.f

Functions to calculate the variance of an EBLUP estimate.

Description

Functions to calculate the EBLUP MSE (=variance). The wrap function calls all the other functions and calculates EBLUP, GREG, SRS, and Synthetic estimates for domain means as well as the variances of the EBLUP, GREG and SRS estimates.

Usage

eblup.mse.f(...)
eblup.mse.f.wrap(domain.data, lme.obj, debug=F, ...)
eblup.mse.f.gamma.i(lme.obj, n,i, ...)

eblup.mse.f.c1(lme.obj, n.i, gamma.i, ...) 
eblup.mse.f.c2.ai(lme.obj, n.i, gamma.i, X.i, ...) 
eblup.mse.f.c2(gamma.i, X.i, X.bar.i, sum.A.i, ...) 
eblup.mse.f.c2.asyvarcovarmat(lme.obj, n.i, ...) 
eblup.mse.f.c3.asyvarcovarmat(lme.obj, asympt.var.covar, n.i, ...) 
eblup.mse.f.c3(lme.obj, asympt.var.covar, n.i, ...) 
eblup.mse.f.c3.star(lme.obj, asympt.var.covar, n.i, mean.resid.i, ...)

Arguments

- **domain.data**: data set with the mean of the auxiliary variables for every domain including the domain ID. Names of the variables must be the same as in the unit-level sample data that were used to fit the lme model.
- **lme.obj**: a linear mixed-effects model generated with `lme`
- **n.i**: the number of samples within domain i
- **gamma.i**: the gamma_i value resulting from the gamma.i method of this function
- **X.i**: the design matrix of sampled elements in domain i
- **X.bar.i**: mean of the populations elements design matrix in domain i
- **sum.A.i**: sum of the domains A_i matrices resulting from `eblup.mse.f.c2.ai`
- **asympt.var.covar**: the asymptotic variance-covariance matrix of the mixed-effects model resulting from `eblup.mse.f.c2.asyvarcovarmat`
- **mean.resid.i**: the mean residual of the fixed-part of the linear mixed-effects model in domain i (i.e., use `level=0` in `predict.lme`)
- **...**: forward attributes to other functions. Not used so far.
- **debug**: details are printed if true - Only used by the wrapper function

Details

Most users will probably only use the convenient wrap function. Nonetheless, all components of the EBLUP variance can also be calculated separately.

Value

A component of the EBLUP variance (aka mean squared error). Which component depends on the function used.

The wrap function returns a data frame with many entries for every domain: The domain-level predictor variables obtained from the domain.data, the mean of the predictor variables and response (aka the sample mean or SRS estimate) observed at the samples, the number of samples (n.i.sample), the mean residuals of a lm (fitted using the fixed part of the lme) and the lme (mean.resid.lm, mean.resid.lme), the synthetic (Synth), EBLUP, and GREG estimates of the mean of the variable of interest, the gamma_i value, the variance of the means for the sample and GREG estimates (.var.mean), the components of the EBLUP variance (c1-c3star), the results of the first and the second method (cf. Rao 2003) to derive the EBLUP variance (EBLUP.var.1, EBLUP.var.2), and the standard errors derived from the variances (.se).
Note
Currently, only single level mixed-effects model with homogeneous variance structure are supported.

Author(s)
Johannes Breidenbach

References
Breidenbach and Astrup (2011)

See Also
JoSAE-package for more examples

Examples
library(nlme)
data(JoSAE.sample.data)
# fit a lme
summary(fit.lme <- lme(biomass.ha ~ mean.canopy.ht, data=JoSAE.sample.data,
    random=-1|domain.ID))
# calculate the first component of the EBLUP variance for a domain with 5 samples
eblup.mse.f.c1(fit.lme, 5, 0.2)

---

**JoSAE.domain.data**  
Dataframe containing the domains population and number of elements and mean of the auxiliary variable

Description
Auxiliary variable: Mean canopy height derived from a photogrammetric canopy height model

Usage
data(JoSAE.domain.data)

Format
A data frame with 14 observations on the following 3 variables.

  domain.ID a numeric vector
  N.i a numeric vector - number of population elements
  mean.canopy.ht.bar a numeric vector - mean of the auxiliary variable
JoSAE.sample.data

Source
Breidenbach and Astrup (2011)

See Also
JoSAE-package for more examples

Examples
data(JoSAE.domain.data)
str(JoSAE.domain.data)

---

JoSAE.sample.data

Sample plots of the Norwegian National Forest Inventory (NNFI) with a variable of interest and an auxiliary variable

Description
Above ground forest biomass over all tree species is the variable of interest. Mean canopy height derived from a photogrammetric canopy height model of 20~cm geometric and 10~cm radiometric resolution is the auxiliary variable.

Usage
data(JoSAE.sample.data)

Format
A data frame with 145 observations on the following 4 variables.

- sample.ID: a numeric vector
- domain.ID: a numeric vector
- biomass.ha: a numeric vector of the variable of interest
- mean.canopy.ht: a numeric vector of the auxiliary variable

Source
Breidenbach, J. and Astrup, R. (2011)

See Also
JoSAE-package for more examples

Examples
data(JoSAE.sample.data)
## maybe str(JoSAE.sample.data); plot(JoSAE.sample.data) ...
plot(biomass.ha~mean.canopy.ht,JoSAE.sample.data)
Index

*Topic datasets
  JoSAE.domain.data, 8
  JoSAE.sample.data, 9

*Topic package
  JoSAE-package, 2

eblup.mse.f, 3, 6
eblup.mse.f.c2.ai, 7
eblup.mse.f.c3.asyvarcovmat, 7

JoSAE (JoSAE-package), 2
JoSAE-package, 2
JoSAE.domain.data, 3, 8
JoSAE.sample.data, 3, 9

lme, 7

predict.lme, 7

rsae, 2