Package 'JoSAE'

July 2, 2014

Type Package

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

Version 0.2.2

Date 2013-12-14

Author Johannes Breidenbach

Maintainer Johannes Breidenbach <job@skogoglandskap.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 lattice,xtable

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-14 23:04:54

R topics documented:

JoSAE-package *Provides functions for some small area estimators and their 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

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 (2012). 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 (if pronounced like Josie). The other reason 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 specialzes on robust SAE (rsae; Schoch, 2011). It implements reampling variance estimators as opposed to analytical variance estimators implemented here. The `rsae` package included a very good vignette as well as the Landsat data set used by Battese et al. (1988). Unfortunately, `rsae` was archived as of R 3.0.2. the landsat data are therefore included in this package.

There are several points where the JoSAE package 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

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, 83, 28-36

Breidenbach, J. and Astrup, R. (2012), Small area estimation of forest attributes in the Norwegian National Forest Inventory. European Journal of Forest Research, 131:1255-1267.

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

Rao, J.N.K. (2003), Small area estimation. Wiley.

Sarndal, C. (1984), Design-consistent versus model-dependent estimation for small domains Journal of the American Statistical Association, JSTOR, 624-631

Schoch, T. (2011), rsae: Robust Small Area Estimation. R package version 0.1-3.

See Also

eblup.mse.f.wrap, 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 estimator.
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
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
overview.domains <- cbind(JoSAE.domain.data, tmp[,-1], n.i=tmp1[,-1])


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

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

    #mean lm residual -- needed for GREG
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
overview.domains$mean.resid.lme <- aggregate(resid(fit.lme, level=0)
                                        , by=list(domain.ID=JoSAE.sample.data$domain.ID)
                                             , mean)[,2]

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



    #GREG estimate
overview.domains$GREG <- overview.domains$synth +  overview.domains$mean.resid.lm

    #EBLUP estimate
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.ht"]
                                    ))
        )
}
                #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$domain.ID==domain.ID[i]
      , "mean.canopy.ht"]##change to other varnames if necessary
                            ))
      , X.bar.i =as.matrix(rbind(i=1##rbind!!
      , x=JoSAE.domain.data[JoSAE.domain.data$domain.ID==domain.ID[i]
      , "mean.canopy.ht.bar"]##change to other varnames if necessary
                             ))
                           , sum.A.i = sum.A.i.mats
                           )
    #third comp
    mse.c3.tmp <- eblup.mse.f.c3(n.i=overview.domains$n.i[
                      overview.domains$domain.ID==domain.ID[i]]
                           , lme.obj=fit.lme
```

```
                          , asympt.var.covar=asy.var.cov.mat)
    #third star comp
    mse.c3.star.tmp <- eblup.mse.f.c3.star( n.i=overview.domains$n.i[
                          overview.domains$domain.ID==domain.ID[i]]
      , lme.obj=fit.lme
      , mean.resid.i=overview.domains$mean.resid.lme[
          overview.domains$domain.ID==domain.ID[i]]
      , asympt.var.covar=asy.var.cov.mat)
    #save result
    result <- rbind(result, data.frame(kommune=domain.ID[i]
      , n.i=overview.domains$n.i[overview.domains$domain.ID==domain.ID[i]]
      , c1=as.numeric(mse.c1.tmp), c2=as.numeric(mse.c2.tmp)
      , c3=as.numeric(mse.c3.tmp), c3star=as.numeric(mse.c3.star.tmp)))
}

            #derive the actual EBLUP variances
overview.domains$EBLUP.var.1 <- result$c1 + result$c2 + 2* result$c3star
overview.domains$EBLUP.var.2 <- result$c1 + result$c2 + result$c3 + result$c3star


    #display the estimates and the sampling error (sqrt(var)) in two tables
round(data.frame(overview.domains[,c("domain.ID", "n.i", "N.i")],
             overview.domains[,c("biomass.ha.bar.sample", "GREG", "synth", "EBLUP")]),2)
       #the sampling errors of the eblup is mostly smaller than the one of the greg estimate.
         #both are always smaller than the sample mean variance.
round(data.frame(overview.domains[,c("domain.ID", "n.i", "N.i")],
                 sqrt(overview.domains[,c("sample.var", "GREG.var",
                        "EBLUP.var.1", "EBLUP.var.2")])),2)

##END: the hard way
```

`eblup.mse.f.wrap` *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.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.c3.asyvarcovarmat(lme.obj, 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.c3.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 compontent of the EBLUP variance (aka mean squared error). Which component depends on the function used.

The wrap function returns a data frame with many entrys 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 random intercept mixed-effects models with homogeneous variance structure are supported.

Author(s)

Johannes Breidenbach

References

Breidenbach and Astrup (2012), Small area estimation of forest attributes in the Norwegian National Forest Inventory. European Journal of Forest Research, 131:1255-1267.

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

Source

Breidenbach, J. and Astrup, R. (2012), Small area estimation of forest attributes in the Norwegian National Forest Inventory. European Journal of Forest Research, 131:1255-1267.

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. (2012), Small area estimation of forest attributes in the Norwegian National Forest Inventory. European Journal of Forest Research, 131:1255-1267.

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)
```

`landsat` *LANDSAT data: Prediction of County Crop Areas Using Survey and Satellite Data*

Description

The `landsat` data.frame is a compilation (by Battese et al., 1988) of survey and satellite data. It consists of data on segments (primary sampling unit; 1 segement =approx= 250 hectares) under corn and soybeans for 12 counties in north-central Iowa; see Details, below.

The landsat data.frame was made available by Tobias Schoch with the R package rsae. Since rsae was archived as of R 3.0.2, the data and this description was copied from rsae 0.1-4 in the archives.

Usage

```
data(landsat)
```

Format

A data frame with 37 observations on the following 10 variables.

`SegmentsInCounty` a numeric vector; no. of segments per county

`SegementID` a numeric vector; sample segment identifier (per county)

`HACorn` a numeric vector; hectares of corn for each sample segment (as reported in the June 1978 Enumerative Survey)

`HASoybeans` a numeric vector; hectares of soybeans for each sample segment (as reported in the June 1978 Enumerative Survey)

`PixelsCorn` a numeric vector; no. of pixels classified as corn for each sample segment (LANDSAT readings)

`PixelsSoybeans` a numeric vector; no. of pixels classified as soybeans for each sample segment (LANDSAT readings)

`MeanPixelsCorn` a numeric vector; county mean number of pixels classified as corn

`MeanPixelsSoybeans` a numeric vector; county mean number of pixels classified as soybeans

`outlier` a logical vector; flags observation no. 33 as outlier

`CountyName` a factor with levels (i.e., county names) `Cerro Gordo` `Hamilton` `Worth` `Humboldt` `Franklin` `Pocahontas` `Winnebago` `Wright` `Webster` `Hancock` `Kossuth` `Hardin`

Details

The `landsat` data is a compilation (by Battese et al., 1988) of the LANDSAT satellite data from the U.S. Department of Agriculture (USDA) and the 1978 June Enumerative Survey.

Survey data: The survey data on the areas under corn and soybeans (reported in hectares) in the 37 segments of the 12 counties (north-central Iowa) have been determined by USDA Statistical Reporting Service staff, who interviewed farm operators. A segment is about 250 hectares.

Satellite data: For the LANDSAT satellite data, information is recorded as "pixels". The USDA has been engaged in research toward transforming satellite information into good estimates of crop areas at the individual pixel and segments level. A pixel is about 0.45 hectares. The satellite (LANDSAT) readings were obtained during August and September 1978.

Data for more than one sample segment are available for several counties (i.e, unbalanced data).

Observations No. 33 has been flaged as outlier (cf., Battese et al. (1988, p. 28).

Source

The data `landsat` is from Table 1 of Battese et al. (1988, p. 29).

Schoch (2011) rsae: Robust Small Area Estimation, R package version 0.1-4: http://cran.r-project.org/src/contrib/Archive/rsa
4.tar.gz

References

Battese, G.E, R.M. Harter, and W.A. Fuller (1988): An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data, *Journal of the American Statistical Association* 83, pp. 28–36.

Examples

```
data(landsat)
```

Index