Package 'saeTrafo'

June 10, 2024

Type Package

Title Transformations for Unit-Level Small Area Models

Version 1.0.4

Author Nora Würz [aut]

Maintainer Nora Würz <nora.wuerz@uni-bamberg.de>

Description The aim of this package is to offer new methodology for unit-level small area models under transformations and limited population auxiliary information. In addition to this new methodology, the widely used nested error regression model without transformations (see ``An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data'' by Battese, Harter and Fuller (1988) <doi:10.1080/01621459.1988.10478561>) and its well-known uncertainty estimate (see ``The estimation of the mean squared error of small-area estimators'' by Prasad and Rao (1990) <doi:10.1080/01621459.1995.10476570>) are provided. In this package, the log transformation and the data-driven log-shift transformation are provided. If a transformation is selected, an appropriate method is chosen depending on the respective input of the population data: Individual population data (see ``Empirical best prediction under a nested error model with log transformation'' by Molina and Martín (2018) <doi:10.1214/17-aos1608>) but also aggregated population data (see ``Estimating regional income indicators under transformations and access to limited population auxiliary information'' by Würz, Schmid and Tzavidis <unpublished>) can be entered. Especially under limited data access, new methodologies are provided in saeTrafo. Several options are available to assess the used model and to judge, present and export its results. For a detailed description of the package and the methods used see the corresponding vignette.

License GPL-2

URL https://github.com/NoraWuerz/saeTrafo

LazyData true

Encoding UTF-8

Imports nlme, emdi, stats, sfsmisc, parallelMap, ggplot2, moments, openxlsx, reshape2, HLMdiag, gridExtra, stringr, readODS, rlang

Suggests simFrame, sf, graphics, R.rsp

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

VignetteBuilder R.rsp

NeedsCompilation no

Copyright inst/COPYRIGHTS

Repository CRAN

Date/Publication 2024-06-10 06:00:02 UTC

Contents

compare_plot *Shows plots for the comparison of estimates*

Description

Function compare_plot is a generic function used to produce plots comparing point and existing MSE/CV estimates of direct and model-based estimation for the Mean.

Methods compare_plot.NER produce plots comparing point and existing MSE/CV estimates of direct and model-based estimation from NER_Trafo. The direct and model-based point estimates are compared by a scatter plot and a line plot. If the input arguments MSE and CV are set to TRUE, two extra plots are created, respectively: the MSE/CV estimates of the direct and model-based estimates are compared by boxplots and scatter plots.

Usage

```
compare_plot(
  model,
  direct,
  MSE = FALSE,
  CV = FALSE,
  label = "orig",
  color = c("blue", "lightblue3"),
  shape = c(16, 16),
  line_type = c("solid", "solid"),
  gg_theme = NULL,
  ...
)

## S3 method for class 'NER'
compare_plot(
  model = NULL,
  direct = NULL,
  MSE = FALSE,
  CV = FALSE,
  label = "orig",
  color = c("blue", "lightblue3"),
  shape = c(16, 16),
  line_type = c("solid", "solid"),
  gg_theme = NULL,
  ...
)
```

Arguments

model	a model object of type "NER", representing point and optional MSE estimates.
direct	an object of type "direct" from "emdi", representing point and MSE estimates. For more information on how to generate direct estimates, please see direct.

MSE	optional logical. If TRUE, the MSE estimates of the direct and model-based estimates are compared via boxplots and scatter plots.
CV	optional logical. If TRUE, the coefficient of variation estimates of the direct and model-based estimates are compared via boxplots and scatter plots.
label	argument that enables to customize title and axis labels. There are three options to label the evaluation plots: (i) original labels ("orig"), (ii) axis labels but no title ("no_title"), (iii) neither axis labels nor title ("blank").
color	a vector with two elements. The first color determines the color for the regression line in the scatter plot and the color for the direct estimates in the remaining plots. The second color specifies the color of the intersection line in the scatter plot and the color for the model-based estimates in the remaining plots. Defaults to c("blue", "lightblue3").
shape	a numeric vector with two elements. The first shape determines the shape of the points in the scatterplot and the shape of the points for the direct estimates in the remaining plots. The second shape determines the shape for the points for the model-based estimates. The options are numbered from 0 to 25. Defaults to c(16, 16).
line_type	a character vector with two elements. The first line type determines the line type for the regression line in the scatter plot and the line type for the direct estimates in the remaining plots. The second line type specifies the line type of the intersection line in the scatter plot and the line type for the model-based estimates in the remaining plots. The options are: "twodash", "solid", "longdash", "dotted", "dotdash", "dashed" and "blank". Defaults to c("solid", "solid").
gg_theme	theme list from package **ggplot2**. For using this argument, package **ggplot2** must be loaded via library(ggplot2). See also Example 2.
...	further arguments passed to or from other methods.

Details

Since all of the comparisons need a direct estimator, the plots are only created for in-sample domains.

Value

Plots comparing direct and model-based estimators for the Mean obtained by ggplot.

A scatter plot and a line plot comparing direct and model-based estimators for each selected indicator obtained by ggplot. If the input arguments MSE and CV are set to TRUE two extra plots are created, respectively: the MSE/CV estimates of the direct and model-based estimates are compared by boxplots and scatter plots.

See Also

saeTrafoObject, direct, NER_Trafo

Examples

```
# Examples for creating plots to compare the saeTrafo object with direct
# estimates (produced by the package emdi)

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp, MSE = TRUE)

# Get direct estimates from the R-package emdi
require(emdi)
library(emdi)
emdi_direct <- direct(y = "eqIncome", smp_data = eusilcA_smp,
                      smp_domains = "district", weights = "weight",
                      var = TRUE, na.rm = TRUE)

# Please detach emdi or use saeTrafo::compare_plot

# Example 1: Comparison plots with uncertainty assessment plots
# (for MSE and CV)
saeTrafo::compare_plot(model = NER_model, direct = emdi_direct, MSE = TRUE,
                       CV = TRUE)

# Example 2: Personalize comparison plots using the options provided with
# this function and ggplot themes
require(ggplot2)
library(ggplot2)
saeTrafo::compare_plot(model = NER_model, direct = emdi_direct, MSE = TRUE,
                       CV = TRUE, label = "no_title",
                       color = c("orange", "green"), shape = c(1,2),
                       line_type = c("dotted", "dashed"),
                       gg_theme = theme(
                          text = element_text(size = 20, color = "blue"),
                          panel.border = element_rect(linetype = "dashed",
                                                      fill = "NA")))
```

compare_pred *Compare predictions of model objects*

Description

Function compare_pred is a generic function used to compare predictions of two model objects.

Method compare_pred.saeTrafo compares predictions of two saeTrafo objects or a saeTrafo object and a emdi object.

Usage

```
compare_pred(object1, object2, MSE = FALSE, ...)

## S3 method for class 'saeTrafo'
compare_pred(object1, object2, MSE = FALSE, ...)
```

Arguments

object1	an object of type "saeTrafo".
object2	an object of type "saeTrafo" or "emdi" (emdiObject).
MSE	optional logical. If TRUE, MSE estimates are returned. Defaults to FALSE and than point estimates are returned.
...	further arguments passed to or from other methods.

Value

Data frame containing the point estimates or the MSE estimates (if MSE is set to TRUE) of both objects. If column names are duplicated, the suffixes "_1" and "_2" are added to their names. "_1" and "_2" standing for object1 and object2, respectively.

See Also

emdi, NER_Trafo, saeTrafoObject

Examples

```
# Example comparing two saeTrafo objects


# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model 1
NER_model_1 <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
```

```
                          self_empl + unempl_ben + age_ben + surv_ben +
                          sick_ben + dis_ben + rent + fam_allow +
                          house_allow + cap_inv + tax_adj,
                          smp_domains = "district",
                          pop_area_size = pop_area_size,
                          pop_mean = pop_mean, pop_cov = pop_cov,
                          smp_data = eusilcA_smp, MSE = TRUE)

# Nested error regression model 2
NER_model_2 <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                          self_empl + unempl_ben + age_ben + surv_ben,
                          smp_domains = "district",
                          pop_area_size = pop_area_size,
                          pop_mean = pop_mean, pop_cov = pop_cov,
                          smp_data = eusilcA_smp, MSE = TRUE)

# Generate a data frame for the comparison of point estimates
compare_pred(NER_model_1, NER_model_2)

# Generate a data frame for the comparison of MSE estimates
compare_pred(NER_model_1, NER_model_2, MSE = TRUE)
```

`estimators` *Presents point, MSE and CV estimates*

Description

Function `estimators` is a generic function used to present point and mean squared error (MSE) estimates. Furthermore, it calculates from both the coefficients of variation (CV).

Method `estimators.saeTrafo` presents point and MSE estimates. Coefficients of variation are calculated using these estimators. The returned object is suitable for printing with the method `print.estimators.saeTrafo`.

Usage

```
estimators(object, MSE, CV, ...)

## S3 method for class 'saeTrafo'
estimators(object, MSE = FALSE, CV = FALSE, ...)
```

Arguments

`object`	an object of type "saeTrafo", representing point and, if chosen, MSE estimates.
`MSE`	optional logical. If `TRUE`, MSE estimates are added to the data frame of point estimates. Defaults to `FALSE`.
`CV`	optional logical. If `TRUE`, coefficients of variation are added to the data frame of point estimates. Defaults to `FALSE`.
`...`	other parameters that can be passed to function `estimators`.

Details

Objects of class "estimators.saeTrafo" have methods for following generic functions: `head` and `tail` (for default documentation, see `head`, `tail`), `as.matrix` (for default documentation, see `matrix`), `as.data.frame` (for default documentation, see `as.data.frame`), `subset` (for default documentation, see `subset`).

Value

The return of `estimators.saeTrafo` is an object of type "estimators.saeTrafo" with point and/or MSE estimates and/or calculated CV's from `saeTrafoObject$ind` and, if chosen, `saeTrafoObject$MSE`. These objects contain two elements, one data frame `ind` and a character naming the indicator or indicator group `ind_name`.

See Also

`saeTrafoObject`, `NER_Trafo`

Examples

```
# Example for presenting point, MSE, and CV estimates for a saeTrafo object

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp, MSE = TRUE)

sae_mean <- estimators(NER_model, MSE = TRUE, CV = TRUE)
class(sae_mean)

# use generic functions for estimators.saeTrafo object
print(sae_mean)
head(sae_mean)
tail(sae_mean)
as.matrix(sae_mean)
as.data.frame(sae_mean)
subset(sae_mean)
```

`eusilcA_pop` *Simulated eusilc data - population data*

Description

The data set is synthetic EU-SILC data based on the data set `eusilcP` from package **simFrame**. The data set is reduced to 17 variables containing three regional variables for the states and districts.

Usage

```
eusilcA_pop
```

Format

A data frame with 25000 observations and 17 variables:

eqIncome numeric; a simplified version of the equivalized household income.

eqsize numeric; the equivalized household size according to the modified OECD scale.

gender factor; the person's gender (levels: male and female).

cash numeric; employee cash or near cash income (net).

self_empl numeric; cash benefits or losses from self-employment (net).

unempl_ben numeric; unemployment benefits (net).

age_ben numeric; old-age benefits (net).

surv_ben numeric; survivor's benefits (net).

sick_ben numeric; sickness benefits (net).

dis_ben numeric; disability benefits (net).

rent numeric; income from rental of a property or land (net).

fam_allow numeric; family/children related allowances (net).

house_allow numeric; housing allowances (net).

cap_inv numeric; interest, dividends, profit from capital investments in unincorporated business (net).

tax_adj numeric; repayments/receipts for tax adjustment (net).

state factor; state (nine levels).

district factor; districts (94 levels).

`eusilcA_smp` *Simulated eusilc data - sample data*

Description

The data set is a simple random sample of data set `eusilcA_pop` which is based on `eusilcP` from package **simFrame**.

Usage

```
eusilcA_smp
```

Format

A data frame with 1000 observations and 18 variables:

eqIncome numeric; a simplified version of the equivalized household income.

eqsize numeric; the equivalized household size according to the modified OECD scale.

gender factor; the person's gender (levels: male and female).

cash numeric; employee cash or near cash income (net).

self_empl numeric; cash benefits or losses from self-employment (net).

unempl_ben numeric; unemployment benefits (net).

age_ben numeric; old-age benefits (net).

surv_ben numeric; survivor's benefits (net).

sick_ben numeric; sickness benefits (net).

dis_ben numeric; disability benefits (net).

rent numeric; income from rental of a property or land (net).

fam_allow numeric; family/children related allowances (net).

house_allow numeric; housing allowances (net).

cap_inv numeric; interest, dividends, profit from capital investments in unincorporated business (net).

tax_adj numeric; repayments/receipts for tax adjustment (net).

state factor; state (nine levels).

district factor; districts (94 levels).

weight numeric; constant weight.

fixef *Extract fixed effects from an saeTrafo object*

Description

Method `fixef.NER` extracts the fixed effects from an saeTrafo object of class "NER".

Usage

```
## S3 method for class 'NER'
fixef(object, ...)

## S3 method for class 'NER'
fixed.effects(object, ...)
```

Arguments

`object`	an object of type "NER".
`...`	additional arguments that are not used in this method.

Details

The alias `fixed.effects` can also be used instead of `fixef`. The generic function `fixef` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `fixed.effects`.

Value

A vector containing the fixed effects is returned.

See Also

`NER_Trafo`, `fixed.effects`

Examples

```
# Example to extract fixed effects

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
```

```
                        cap_inv + tax_adj,
                        smp_domains = "district",
                        pop_area_size = pop_area_size,
                        pop_mean = pop_mean, pop_cov = pop_cov,
                        smp_data = eusilcA_smp)

fixef(NER_model)
```

---

getData *Extract saeTrafo object data*

Description

Method `getData.NER` extracts the data frame used to fit the model.

Usage

```
## S3 method for class 'NER'
getData(object, ...)
```

Arguments

`object`	an object of type "NER".
`...`	additional arguments that are not used in this method.

Details

The generic function `getData` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `getData`.

Value

Data frame used to fit the model. For "NER" the (untransformed) sample data is returned.

See Also

`NER_Trafo`, `getData`

Examples

```
# Example to extract object data

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")
```

```
# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

getData(NER_model)
```

`getGroups` *Extract grouping factors from an saeTrafo object*

Description

Method `getGroups.NER` extracts grouping factors from a saeTrafo object.

Usage

```
## S3 method for class 'NER'
getGroups(object, ...)
```

Arguments

`object`	an object of type "NER".
`...`	additional arguments that are not used in this method.

Details

The generic function `getGroups` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `getGroups`.

Value

A vector containing the grouping factors.

See Also

`NER_Trafo`, `getGroups`

Examples

```
# Example to extract grouping factors

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

getGroups(NER_model)
```

---

getGroupsFormula *Extract grouping formula from a saeTrafo object*

Description

Method `getGroupsFormula.NER` extracts the grouping formula from an saeTrafo object.

Usage

```
## S3 method for class 'NER'
getGroupsFormula(object, ...)
```

Arguments

`object`	an object of type "NER".
`...`	additional arguments that are not used in this method.

Details

The generic function `getGroupsFormula` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `getGroupsFormula`.

Value

A one-sided formula.

See Also

NER_Trafo getGroupsFormula

Examples

```
# Example to extract grouping formula

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

getGroupsFormula(NER_model)
```

---

getResponse *Extract response variable from an saeTrafo object*

Description

Method getResponse.NER extracts the response variable from a saeTrafo object.

Usage

```
## S3 method for class 'NER'
getResponse(object, ...)
```

Arguments

object	an object of type "NER".
...	additional arguments that are not used in this method.

Details

The generic function getResponse is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see getResponse.

Value

Vector containing the response variable.

See Also

NER_Trafo, getResponse

Examples

```
# Example to extract the response variable

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

getResponse(NER_model)
```

---

getVarCov *Extract variance-covariance matrix from an saeTrafo object*

Description

Method getVarCov.NER extracts the variance-covariance matrix from a fitted model of class "NER".

Usage

```
## S3 method for class 'NER'
getVarCov(obj, individuals = 1, type = "random.effects", ...)
```

Arguments

obj	an object of type "NER".
individuals	vector of levels of the in-sample domains can be specified for the types "conditional" or "marginal".

type	a character that determines the type of variance-covariance matrix. Types that can be chosen (i) random-effects variance-covariance matrix ("random.effects"), (ii) conditional variance-covariance matrix ("conditional"), (iii) marginal variance-covariance matrix ("marginal"). Defaults to "random.effects".
...	additional arguments that are not used in this method.

Details

The generic function getVarCov is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see getVarCov.

Value

A variance-covariance matrix or a list of variance-covariance matrices, if more than one individual is selected. For method getVarCov.NER, the dimensions of the matrices are 1 x 1 for type "random.effects" and number of in-sample domains x number of in-sample domains for types "conditional" and "marginal".

See Also

NER_Trafo, getVarCov

Examples

```
# Example to extract variance-covariance matrix

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

getVarCov(NER_model)
```

`intervals` *Confidence intervals on coefficients of an saeTrafo object*

Description

Method `intervals.NER` provides the approximate confidence intervals on the coefficients (fixed effects) of an saeTrafo object.

Usage

```
## S3 method for class 'NER'
intervals(object, level = 0.95, parm = NULL, ...)
```

Arguments

`object`	an object of type "NER".
`level`	an optional numeric value with the confidence level for the intervals. Defaults to 0.95.
`parm`	vector of names to specify which parameters are to be given confidence intervals. If `NULL`, all parameters are taken into account. Defaults to `NULL`.
`...`	additional arguments that are not used in this method.

Details

The generic function `intervals` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `intervals`.

Value

A matrix with rows corresponding to the parameters and columns containing the lower confidence limits (lower), the estimated values (est.), and upper confidence limits (upper).

See Also

`NER_Trafo`, `intervals`

Examples

```
# Example to extract confidence intervals on coefficients

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
```

```
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

intervals(NER_model)
```

---

load_shapeaustria *Loading the shape file for Austrian districts*

Description

The function simplifies to load the shape file for Austrian districts.

Usage

```
load_shapeaustria()
```

Details

The shape file contains the borders of Austrian districts. Thus, it can be used for the visualization of estimation results for Austrian districts.

Value

A shape file of class `sf`.

---

map_plot *Visualizes regional disaggregated estimates on a map*

Description

Function `map_plot` creates spatial visualizations of the estimates obtained by small area estimation methods.

Usage

```
map_plot(
  object,
  MSE = FALSE,
  CV = FALSE,
  map_obj = NULL,
  map_dom_id = NULL,
  map_tab = NULL,
  color = c("white", "red4"),
  scale_points = NULL,
  guide = "colourbar",
  return_data = FALSE
)
```

Arguments

object	an object of type saeTrafo, containing the estimates to be visualized.
MSE	optional logical. If TRUE, the MSE is also visualized. Defaults to FALSE.
CV	optional logical. If TRUE, the CV is also visualized. Defaults to FALSE.
map_obj	an sf object (map object) as defined by the **sf** package on which the data should be visualized.
map_dom_id	a character string containing the name of a variable in map_obj that indicates the domains.
map_tab	a data.frame object with two columns that match the domain variable from the census data set (first column) with the domain variable in the map_obj (second column). This should only be used if the IDs in both objects differ.
color	a vector of length 2 defining the lowest and highest color in the plots.
scale_points	a numeric vector of length two. This scale will be used for every plot.
guide	character passed to scale_fill_gradient from **ggplot2**. Possible values are "none", "colourbar", and "legend". Defaults to "colourbar".
return_data	if set to TRUE, a fortified data frame including the map data as well as the chosen indicators is returned. Customized maps can easily be obtained from this data frame via the package **ggplot2**. Defaults to FALSE.

Value

Creates the plots demanded and, if selected, a fortified data.frame containing the mapdata and chosen indicators.

See Also

sf, NER_Trafo, saeTrafoObject

Examples

```
# Examples for creating maps to visualize the saeTrafo estimates

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp, MSE = TRUE)

# Load shape file
load_shapeaustria()

# Example 1: Map plots with uncertainty plots (for MSE and CV)
map_plot(NER_model, MSE = TRUE, CV = TRUE, map_obj = shape_austria_dis,
         map_dom_id = "PB")

# Example 2: Personalize map plot for point estimates
map_plot(NER_model, map_obj = shape_austria_dis, map_dom_id = "PB",
         color = c("white", "darkblue"),
         scale_points = c(0, max(NER_model$ind$Mean)))

# Example 3: More options to personalize map plot by using return_data = TRUE
# and ggplot2
require(ggplot2)
library(ggplot2)
data_plot <- map_plot(NER_model, map_obj = shape_austria_dis, map_dom_id = "PB",
                      return_data = TRUE)
ggplot(data_plot, aes(long = NULL, lat = NULL,
                      group = "PB", fill = Mean))+
       geom_sf(color = "black") +
       theme_void() +
       ggtitle("Personalized map") +
       scale_fill_gradient2(low = "red", mid = "white", high = "darkgreen",
                            midpoint = 20000)

# Example 4: Create a suitable mapping table to use numerical identifiers of
# the shape file

# First find the right order
dom_ord <- match(shape_austria_dis$PB, NER_model$ind$Domain)
```

```
#Create the mapping table based on the order obtained above
map_tab <- data.frame(pop_data_id = NER_model$ind$Domain[dom_ord],
                      shape_id = shape_austria_dis$BKZ)

# Create map plot for mean indicator - point and CV estimates but no MSE
# using the numerical domain identifiers of the shape file
map_plot(object = NER_model, MSE = FALSE, CV = TRUE,
         map_obj = shape_austria_dis,
         map_dom_id = "BKZ", map_tab = map_tab)
```

NER_Trafo *Nested error regression Model under transformations*

Description

Function NER_Trafo estimates small area means based on the (transformed) nested error regression (NER) model (*Battese et al., 1988*). In contrast to the empirical best predictor of *Molina and Rao (2010)*, which is implemented in the package **emdi** (ebp), no unit-level population data are required.

NER_Trafo supports the log as well as the data-driven log-shift transformation. Especially for skewed variables, (data-driven) transformations are useful to meet the model assumptions for the error terms. If a transformation is chosen and aggregates (means and covariance) are simultaneously provided for the population, point estimates are produced by the method of *Wuerz et al. (2022)*, which uses kernel density estimation to resolve the issue of not having access to population micro-data. In the case that population data are available at unit-level and the log or log-shift transformation is selected, the bias-correction of *Berg and Chandra (2014)* and *Molina and Martín (2018)* is applied. For this data situation, more methods and options are provided in the package **emdi**. If only population means are available and the log or log-shift transformation is selected, a bias-correction due to the transformation is added but for the lack of access to population data no correction is available. Therefore, a part of the bias is disregarded.

Additionally, analytically mean squared errors (MSE) are calculated in the case of no transformation following *Prasad and Rao (1990)*. For the log and log-shift transformation, a parametric bootstrap procedure proposed by *Wuerz et al. (2022)* following *Gonzalez-Manteiga et al. (2008)* is applied. Please note that this can only be determined if covariance data are also provided. If population data is available on unit-level a bootstrap procedure as described in *Molina and Martín (2018)* is applied.

Usage

```
NER_Trafo(
  fixed,
  pop_area_size = NULL,
  pop_mean = NULL,
  pop_cov = NULL,
  pop_data = NULL,
```

```
  pop_domains = NULL,
  smp_data,
  smp_domains,
  threshold = 30,
  B = 50,
  transformation = "log.shift",
  interval = "default",
  MSE = FALSE,
 parallel_mode = ifelse(grepl("windows", .Platform$OS.type), "socket", "multicore"),
  cpus = 1,
  seed = 123
)
```

Arguments

fixed	a two-sided linear formula object describing the fixed-effects part of the nested error linear regression model with the dependent variable on the left of a ~ operator and the explanatory variables on the right, separated by + operators. The argument corresponds to the argument fixed in function lme.
pop_area_size	a named numeric vector containing the number of individuals within each domain. This numeric vector is named with the domain names.
pop_mean	a named list. Each element of the list contains the population means for the p covariates for a specicfic domain. The list is named with the respective domain name. The numeric vector within the list is named with the covariate names. The covariates right of the ~ operator in fixed need to comprise.
pop_cov	a named list. Each element of the list contains the domain-specific covariance matrice for p covariates for a specicfic domain. The list is named with the respective domain name. The matrix within the list has row and column names with the respective covariate names. The covariates right of the ~ operator in fixed need to comprise. If pop_cov is not available only a bias-correction due to the transformation is added but for the lack of access to population data a correction is not possible. Additionally, no MSE could be determined.
pop_data	a data frame that needs to comprise the variables named on the right of the ~ operator in fixed, i.e. the explanatory variables, and pop_domains. Please note, if population data is available other methods using unit-level population data, like ebp, could be applied.
pop_domains	a character string containing the name of a variable that indicates domains in the population data. The variable can be numeric or a factor but needs to be of the same class as the variable named in smp_domains. Only needed if population data are given.
smp_data	a data frame that needs to comprise all variables named in fixed and smp_domains.
smp_domains	a character string containing the name of a variable that indicates domains in the sample data. The variable can be numeric or a factor but needs to be of the same class as the variable named in pop_domains.
threshold	a numeric value indicating the threshold for using pooled domain data (for domains with sample sizes below the threshold) or non pooled domain data (for

	domains with sample sizes above the threshold) for the density estimation within the approach of *Wuerz et al. (2022)*. Defaults to 30.
B	a number determining the number of bootstrap replications in the parametric bootstrap approach. The number must be greater than 1. Defaults to 50. For practical applications, values larger than 200 are recommended.
transformation	a character string. Three different transformation types for the dependent variable can be chosen (i) no transformation ("no"); (ii) log transformation ("log"); (iii) Log-Shift transformation ("log.shift"). Defaults to "log.shift".
interval	a string equal to 'default' or a numeric vector containing a lower and upper limit determining an interval for the estimation of the optimal parameter for the log-shift transformation. The interval is passed to function optimize for the optimization. Defaults to an interval based on the range of y. If the convergence fails, it is often advisable to choose a smaller more suitable interval. For right skewed distributions, the negative values may be excluded.
MSE	optional logical. If TRUE, MSE estimates are provided. Defaults to FALSE.
parallel_mode	modus of parallelization, defaults to an automatic selection of a suitable mode, depending on the operating system, if the number of cpus is chosen higher than 1. For details, see parallelStart.
cpus	number determining the kernels that are used for the parallelization. Defaults to 1. For details, see parallelStart.
seed	an integer to set the seed for the random number generator. For the usage of random number generation, see Details. If seed is set to NULL, seed is chosen randomly. Defaults to 123.

Details

For the parametric bootstrap and the density estimation approach random number generation is used. Thus, a seed is set by the argument seed.

Value

An object of class "NER", "saeTrafo" that provides estimators for regional means optionally corresponding MSE estimates. Several generic functions have methods for the returned object. For a full list and descriptions of the components of objects of class "saeTrafo", see saeTrafoObject.

References

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

Berg, E. and Chandra, H. (2014). Small area prediction for a unit-level lognormal model. Computational Statistics & Data Analysis, Vol.78, 159–175.

González-Manteiga, W., Lombardía, M. J., Molina, I., Morales, D. and Santamaría, L. (2008). Analytic and bootstrap approximations of prediction errors under a multivariate Fay–Herriot model. Computational Statistics & Data Analysis, Vol. 52, No. 12, 5242-5252.

Molina, I. and Martín, N. (2018). Empirical best prediction under a nested error model with log transformation. The Annals of Statistics, Vol.46, No. 5, 1961–1993.

Molina, I. and Rao, J.N.K. (2010). Small area estimation of poverty indicators. The Canadian Journal of Statistics, Vol. 38, No.3, 369-385.

Prasad, N.N., Rao, J.N.K. (1990). The estimation of the mean squared error of small-area estimators. Journal of the American statistical association, Vol. 85, No. 409, 163-171.

Wuerz, N., Schmid, T., and Tzavidis, N. (2022) Estimating regional income indicators under transformations and access to limited population auxiliary information. Journal of the Royal Statistical Society: Series A (Statistics in Society), Vol. 185, No. 4, 1679-1706.

See Also

saeTrafoObject, lme, estimators.saeTrafo, plot.saeTrafo, summaries.saeTrafo

Examples

```
# Examples for (transformed) nested error regression model

# Load Data
data("eusilcA_pop")
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# formula object for all examples
formula <- eqIncome ~ gender + eqsize + cash + self_empl + unempl_ben +
                      age_ben + surv_ben + sick_ben + dis_ben + rent +
                      fam_allow + house_allow + cap_inv + tax_adj

# For all four examples, no MSEs/variances are determined in order to avoid
# long run times. These can be obtained with MSE = TRUE.

# Example 1: No transformation - classical NER
NER_model_1 <- NER_Trafo(fixed = formula, transformation = "no",
                         smp_domains = "district", smp_data = eusilcA_smp,
                         pop_area_size = pop_area_size, pop_mean = pop_mean)

# Example 2: Log-shift transformation and population aggregates
# (means and covariances) with changed threshold
NER_model_2 <- NER_Trafo(fixed = formula,
                         smp_domains = "district", smp_data = eusilcA_smp,
                         pop_area_size = pop_area_size, pop_mean = pop_mean,
                         pop_cov = pop_cov, threshold = 50)
```

```
# Example 3: Log-shift transformation and population data
# A bias-corrections which need unit-level population data are applied
NER_model_3 <- NER_Trafo(fixed = formula,
                         smp_domains = "district", smp_data = eusilcA_smp,
                         pop_data = eusilcA_pop, pop_domains = "district")


# Example 4: Log-shift transformation and population aggregates
# (only means (!) - Therefore, no MSE estimation is available, bias is
# disregarded)
NER_model_4 <- NER_Trafo(fixed = formula,
                         smp_domains = "district", smp_data = eusilcA_smp,
                         pop_area_size = pop_area_size, pop_mean = pop_mean)
```

`plot.saeTrafo` *Plots for an saeTrafo object*

Description

Diagnostic plots of the nested error regression model (see also NER_Trafo) are obtained. These include Q-Q plots and density plots of residuals and random effects, a Cook's distance plot for detecting outliers and the log-likelihood of the estimation of the optimal parameter in log-shift transformations. The return depends on the transformation, such that a plot for the optimal parameter is only returned in case if a transformation with transformation parameter is chosen. The range of the x-axis is optional but necessary to change if there are convergence problems. All plots are obtained by ggplot.

Usage

```
## S3 method for class 'saeTrafo'
plot(
  x,
  label = "orig",
  color = c("blue", "lightblue3"),
  gg_theme = NULL,
  cooks = TRUE,
  range = NULL,
  ...
)
```

Arguments

x	an object of type "NER", representing point and, if chosen, MSE estimates obtained by the (transformed) nested error regression model (see also NER_Trafo).

label	argument that enables to customize title and axis labels. There are three instant options to label the diagnostic plot: (i) original labels ("orig"), (ii) axis lables but no title ("no_title"), (iii) neither axis labels nor title ("blank"), (iv) individual labels by a list that needs to have below structure. Six elements can be defined called qq_res, qq_ran, d_res, d_ran, cooks and opt_lambda for the six different plots and these list elements need to have three elements each called title, y_lab and x_lab. Only the labels for the plots that should be different to the original need to be specified. Please see the details section for an example with the default labels.
color	a character vector with two elements. The first element defines the color for the line in the QQ-plots, for the Cook's Distance plot and for the optimal parameter plot. The second element defines the color for the densities.
gg_theme	theme list from package **ggplot2**. For using this argument, package **ggplot2** must be loaded via library(ggplot2).
cooks	optional logical. If TRUE, a Cook's distance plot is returned. The used method mdffits.default from the package **HLMdiag** struggles when data sets get large. In these cases, cooks should be set to FALSE. It defaults to TRUE.
range	optional sequence determining the range of the x-axis for plots of the optimal transformation parameter that defaults to NULL. In that case a range of the default interval is used for the plots of the optimal parameter. This leads in some cases to convergence problems such that it should be changed to e.g. the selected interval. The default value depends on the chosen data driven transformation and equals the default interval for the estimation of the optimal parameter.
...	optional arguments passed to generic function.

Details

The default settings of the label argument are as follows:

list(

qq_res = c(title="Error term", y_lab="Quantiles of pearson residuals", x_lab="Theoretical quantiles"),

qq_ran = c(title="Random effect", y_lab="Quantiles of random effects", x_lab="Theoretical quantiles"),

d_res = c(title="Density - Pearson residuals", y_lab="Density", x_lab="Pearson residuals"),

d_ran = c(title="Density - Standardized random effects", y_lab="Density", x_lab="Standardized random effects"),

cooks = c(title="Cook's Distance Plot", y_lab="Cook's Distance", x_lab="Index"),

opt_lambda = c(title="Log-Shift - REML", y_lab="Log-Likelihood", x_lab="expression(lambda)"))

Value

Two Q-Q plots in one grid, two density plots, a Cook's distance plot and a likelihood plot for the optimal parameter of transformations with transformation parameter obtained by ggplot.

See Also

saeTrafoObject, NER_Trafo

Examples

```
# Examples for diagnostic plots

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

# Example 1: Default diagnostic plot
plot(NER_model)

# Example 2: Creation of diagnostic plots without labels and titles,
# different colors and without Cook's distance plot.

plot(NER_model, label = "no_title", color = c("red", "yellow"),
     cooks = FALSE)
```

---

pop_area_size *Aggregates from simulated eusilc population data: domain sizes*

Description

This data contains aggregates from eusilcA_pop which is based on eusilcP from package **simFrame**.

Usage

```
pop_area_size
```

Format

A named numeric vector containing the number of individuals within each domain. This numeric vector is named with the domain names.

`pop_cov` *Aggregates from simulated eusilc population data: domain-specific covariances*

Description

This data contains aggregates from `eusilcA_pop` which is based on `eusilcP` from package **simFrame**.

Usage

```
pop_cov
```

Format

A named list. Each element of the list contains the domain-specific covariance matrice for p covariates for a specicfic domain. The list is named with the respective domain name. The matrix within the list has row and column names with the respective covariate names. The covariates right of the ~ operator in `fixed` need to comprise.

`pop_mean` *Aggregates from simulated eusilc population data: domain-specific means*

Description

This data contains aggregates from `eusilcA_pop` which is based on `eusilcP` from package **simFrame**.

Usage

```
pop_mean
```

Format

A named list. Each element of the list contains the population means for the p covariates for a specicfic domain. The list is named with the respective domain name. The numeric vector within the list is named with the covariate names. The covariates right of the ~ operator in `fixed` need to comprise.

predict.NER *Predictions from saeTrafo objects*

Description

Method predict.NER extracts the direct estimates, the empirical best linear unbiased or empirical best predictors for all domains from an saeTrafo object.

Usage

```
## S3 method for class 'NER'
predict(object, ...)
```

Arguments

object	an object of type "saeTrafo".
...	additional arguments that are not used in this method.

Value

Data frame with domain predictors.

See Also

saeTrafoObject, NER_Trafo

Examples

```
# Examples for Predictions from saeTrafo objects

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

predict(NER_model)
```

qqnorm.saeTrafo *Quantile-quantile plots for an saeTrafo object*

Description

Normal quantile-quantile plots of the underlying model (see NER_Trafo) are obtained. The plots are obtained by ggplot.

Usage

```
## S3 method for class 'saeTrafo'
qqnorm(y, color = c("blue", "lightblue3"), gg_theme = NULL, ...)
```

Arguments

y	a model object of type "saeTrafo" (see NER_Trafo).
color	a character vector with two elements. The first element defines the color for the line in the Q-Q plots, for the Cook's Distance plot and for the Box-Cox plot. The second element defines the color for the densities.
gg_theme	theme list from package **ggplot2**. For using this argument, package **ggplot2** must be loaded via `library(ggplot2)`. See also Example 2.
...	optional arguments passed to generic function.

Value

Two Q-Q plots in one grid obtained by ggplot.

See Also

saeTrafoObject, NER_Trafo

Examples

```
# Examples for Quantile-quantile plots

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
```

```
                      pop_mean = pop_mean, pop_cov = pop_cov,
                      smp_data = eusilcA_smp)

# Example 1: Default Quantile-quantile plots
qqnorm(NER_model)

# Example 2: Personalized plot using theme
require("ggplot2")
library(ggplot2)
qqnorm(NER_model,
       color = c("red", "darkgreen"),
       gg_theme = theme(panel.background = element_rect(fill = NA),
                        panel.grid.major = element_line(colour = "grey50"),
                        panel.ontop = TRUE)
)
```

`ranef` *Extract random effects of saeTrafo object*

Description

Method `ranef.NER` extracts the random effects from an saeTrafo object.

Usage

```
## S3 method for class 'NER'
ranef(object, ...)

## S3 method for class 'NER'
random.effects(object, ...)
```

Arguments

`object`	an object of type "NER".
`...`	additional arguments that are not used in this method.

Details

The alias `random.effects` can also be used instead of `ranef`. The generic function `ranef` is imported from package **nlme** and re-exported to make the S3-methods available, even though the **nlme** package itself is not loaded or attached. For default documentation, see `random.effects`.

Value

A vector containing the estimated random effects at domain level is returned.

See Also

NER_Trafo, random.effects

Examples

```
# Example to extract random effects

# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp)

ranef(NER_model)
```

---

saeTrafo *The R Package saeTrafo for Estimating unit-level Small Area Models under Transformations*

Description

The package **saeTrafo** supports estimating regional means based on the nested error regression model (*Battese et al., 1988*). Therefore, point estimation and mean squared error estimation (*Prasad and Rao, 1990*) for the classical model is offered. In addition to the classical model, the logarithmic and the data-driven log-shift transformation are provided. The core function NER_Trafo allows several options to enter population data: Either individual population data or only aggregates can be entered. If full population data is given, the method of *Molina and Martín (2018)* is applied. Compared to other small area packages, these transformations are accessible in the absence of population micro-data. Only population aggregates (mean values, population sizes and preferably also covariances) need to be supplied. The methodology for point and mean squared error estimates is described in *Wuerz et al. (2022)* and is made available in a user-friendly way within **saeTrafo**.

Details

The estimation function is called NER_Trafo. For this function, several methods are available such as estimators.saeTrafo and summaries.saeTrafo. For a full list, please see saeTrafoObject. Furthermore, functions map_plot and write.excel help to visualize and export results. An overview of all currently provided functions can be requested by `library(help=saeTrafo)`.

References

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

Molina, I. and Martín, N. (2018). Empirical best prediction under a nested error model with log transformation. The Annals of Statistics, Vol.46, No. 5, 1961–1993.

Prasad, N.N., Rao, J.N.K. (1990). The estimation of the mean squared error of small-area estimators. Journal of the American statistical association, Vol.85, No. 409, 163-171.

Wuerz, N., Schmid, T., Tzavidis, N. (2022). Estimating regional income indicators under transformations and access to limited population auxiliary information. Unpublished. _Package

---

`saeTrafoObject` *Fitted saeTrafoObject*

Description

An object of class saeTrafo that represents point predictions of domain-specific means. Optionally, it also contains corresponding MSE estimates. Objects of these classes have methods for various generic functions. See Details for more information.

Details

Objects of class "saeTrafo" and subclass "NER_Trafo" have the following methods: `compare_pred`, `estimators`, `plot.saeTrafo`, `predict.NER`, `qqnorm.saeTrafo`, `compare_plot`, `getData`, `getGroups`, `getGroupsFormula`, `getResponse`, `plot` (for documentation, see `plot.saeTrafo`), `print`, `qqnorm` (for documentation, see `qqnorm.saeTrafo`) and `summary` (for documentation, see `summaries.saeTrafo`), `coef` (for default documentation, see `coef`), `confint` (for default documentation, see `confint`), `family` (for default documentation, see `family`), `fitted` (for default documentation, see `fitted.values`), `fixef`, `formula` (for default documentation, see `formula`), `getVarCov`, `intervals`, `logLik` (for default documentation, see `logLik`), `nobs` (for default documentation, see `nobs`), `ranef`, `residuals` (for default documentation, see `residuals`), `terms` (for default documentation, see `terms`), `vcov` (for default documentation, see `vcov`) `sigma` (for default documentation, see `sigma`)

Value

The following components are always included in an saeTrafo object but not always filled:

`call`	the function call that produced the object.
`fixed`	for details, see `fixed` in `NER_Trafo`.
`framework`	a list with components that describe the data setup, e.g., number of domains in the sample.
`ind`	data frame containing estimates for the mean per domain.

method	character returning the method for the estimation approach used to fit the linear mixed model and for the the optimal lambda, in our case "reml".
model	list containing a selection of model components.
MSE	data frame containing MSE estimates corresponding to the mean predictions in ind per domain if MSE is selected in function call. If MSE, MSE is NULL.
transformation	character or list containing information about applied transformation.
transform_param	a list with two elements, optimal_lambda and shift_par, where the first contains the optimal parameter for a transformation with transformation parameter or NULL for no and log transformation and the second the potential shift parameter for the log transformation and NULL for no transformation.
successful_bootstraps	a numeric returning the number of successful bootstraps. If MSE = FALSE in the function call or transformation = "no", successful_bootstraps is NULL.

See Also

NER_Trafo, lme, lmeObject

---

summaries.saeTrafo *Summarizes an saeTrafo object*

Description

Additional information about the data and model in small area estimation methods and components of an saeTrafo object are extracted. The returned object is suitable for printing with the print function.

Usage

```
## S3 method for class 'NER'
summary(object, ...)
```

Arguments

object	an object of type "NER", representing point and MSE estimates.
...	additional arguments that are not used in this method.

Value

an object of type "summary.NER" with information about the sample and population data, the usage of transformation, normality tests and information of the model fit.

See Also

saeTrafoObject, NER_Trafo, skewness, kurtosis, shapiro.test

`write.excel` *Exports an saeTrafo Object to an Excel file or OpenDocument Spreadsheet*

Description

Function `write.excel` enables the user to export point and MSE estimates as well as diagnostics from the `summary` to an Excel file. The user can choose if the results should be reported in one or several Excel sheets. Furthermore, a selection of indicators can be specified. Respectively the function `write.ods` enables the export to OpenDocument Spreadsheets. Note that while `write.exel` will create a single document `write.ods` will create a group of files.

Usage

```
write.excel(
  object,
  file = "excel_output.xlsx",
  MSE = FALSE,
  CV = FALSE,
  split = FALSE
)

write.ods(
  object,
  file = "ods_output.ods",
  MSE = FALSE,
  CV = FALSE,
  split = FALSE
)
```

Arguments

`object`	an object of type "saeTrafo", representing point and MSE estimates.
`file`	path and filename of the spreadsheet to create. It should end on .xlsx or .ods respectively.
`MSE`	logical. If `TRUE`, the MSE of the saeTrafoObject is exported. Defaults to `FALSE`.
`CV`	logical. If `TRUE`, the CV of the saeTrafoObject is exported. Defaults to `FALSE`.
`split`	logical. If `TRUE`, point estimates, MSE and CV are written to different sheets in the Excel file. In `write.ods` `TRUE` will result in different files for point estimates and their precision. Defaults to `FALSE`.

Details

These functions create an Excel file via the package **openxlsx** and ODS files via the package **readODS**. Both packages require a zip application to be available to R. If this is not the case the authors of **openxlsx** suggest the first of the following two ways.

- Install Rtools from: http://cran.r-project.org/bin/windows/Rtools/ and modify the system PATH during installation.
- If Rtools is installed, but no system path variable is set. One can set such a variable temporarily to R by a command like: `Sys.setenv("R_ZIPCMD" = "PathToTheRToolsFolder/bin/zip.exe")`.

To check if a zip application is available they recommend the command `shell("zip")`.

Value

An Excel file is created in your working directory, or at the given path. Alternatively multiple ODS files are created at the given path.

See Also

saeTrafoObject, NER_Trafo

Examples

```
# Examples for exporting saeTrafoObject to an Excel file or OpenDocument
# Spreadsheet

## Not run:
# Load Data
data("eusilcA_smp")
data("pop_area_size")
data("pop_mean")
data("pop_cov")

# Nested error regression model
NER_model <- NER_Trafo(fixed = eqIncome ~ gender + eqsize + cash +
                       self_empl + unempl_ben + age_ben + surv_ben +
                       sick_ben + dis_ben + rent + fam_allow + house_allow +
                       cap_inv + tax_adj,
                       smp_domains = "district",
                       pop_area_size = pop_area_size,
                       pop_mean = pop_mean, pop_cov = pop_cov,
                       smp_data = eusilcA_smp, MSE = TRUE)

# Example 1: Export estimates for all indicators and uncertainty measures and
# diagnostics to Excel
write.excel(NER_model, file = "excel_output.xlsx", MSE = TRUE, CV = TRUE)

# Example 2: Single Excel sheets for point, MSE and CV estimates
write.excel(NER_model, file = "excel_output_split.xlsx", MSE = TRUE,
            CV = TRUE, split = TRUE)

# Example 3: Same as example 1 but for an ODS output
write.ods(NER_model, file = "ods_output_all.ods", MSE = TRUE, CV = TRUE)

## End(Not run)
```

Index