Package ‘mpitbR’

July 23, 2025

Type Package

Title Calculate Alkire-Foster Multidimensional Poverty Measures

Version 1.0.1

Maintainer Ignacio Girela <ignacio.girela@unc.edu.ar>

Description Estimate Multidimensional Poverty Indices disaggregated by population subgroups based on the Alkire and Foster method (2011) <doi:10.1016/j.jpubeco.2010.11.006>. This includes the calculation of standard errors and confidence intervals. Other partial indices such as incidence, intensity and indicator-specific measures as well as intertemporal changes analysis can also be estimated. The standard errors and confidence intervals are calculated considering the complex survey design.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports purrr, stats, survey

Depends R (>= 2.10)

Suggests gridExtra, knitr, parallel, rmarkdown, scales, stringi, tidyverse

URL https://github.com/girelaignacio/mpitbR

BugReports https://github.com/girelaignacio/mpitbR/issues

VignetteBuilder knitr

NeedsCompilation no

Author Ignacio Girela [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-3297-3854>), CONICET [fnd]

Repository CRAN

Date/Publication 2025-02-11 03:20:02 UTC

Contents

`ben_dhs06` *Preprocessed Benin DHS 06 dataset*

Description

The Benin DHS 06 dataset containing preprocessed data using the OPHI do-file from the Demographic and Health Survey (DHS) program.

Usage

```
ben_dhs06
```

Format

An object of class `data.frame` with 89282 rows and 18 columns.

Author(s)

OPHI

Source

- DHS Program: https://ohss.dhs.gov/about-our-data - OPHI do-file: https://ophi.org.uk/global-mpi/2024

References

OPHI and UNDP (2024). Global Multidimensional Poverty Index 2024: Poverty amid conflict. United Nations Development Programme (UNDP) and Oxford Poverty and Human Development Initiative (OPHI), University of Oxford.

`ben_dhs17_18` *Preprocessed Benin DHS 17-18 dataset*

Description

The Benin DHS 17-18 dataset containing preprocessed data using the OPHI do-file from the Demographic and Health Survey (DHS) program.

Usage

```
ben_dhs17_18
```

Format

An object of class `data.frame` with 73728 rows and 18 columns.

Author(s)

OPHI

Source

- DHS Program: https://ohss.dhs.gov/about-our-data - OPHI do-file: https://ophi.org.uk/global-mpi/2024

References

OPHI and UNDP (2024). Global Multidimensional Poverty Index 2024: Poverty amid conflict. United Nations Development Programme (UNDP) and Oxford Poverty and Human Development Initiative (OPHI), University of Oxford.

`coef.cotframe` *Extract the coefficients from the estimated changes over time measures*

Description

Extract the coefficients from the estimated changes over time measures

Usage

```
## S3 method for class 'cotframe'
coef(object, ...)
```

Arguments

`object`	a "cotframe"-class object
`...`	other arguments.

Details

The coef method for "cotframe"-class objects find the point estimates from the changes over time data frame. This method work for only one measure c("M0","H","A","hd","hdk"). Then, user should subset the data frame with the estimates by the chosen measure (including other preferred categories, i.e., poverty cut-off, subgroup, etc.)

Value

Coefficients extracted from the model cotframe object.

Author(s)

Ignacio Girela

See Also

confint, and summary methods, and mpitb.est function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

coef.lframe — *Extract the coefficients from the estimated cross-sectional measures*

Description

Extract the coefficients from the estimated cross-sectional measures

Usage

```
## S3 method for class 'lframe'
coef(object, ...)
```

Arguments

`object`	a "lframe"-class object
`...`	other arguments.

Details

The `coef` method for "lframe"-class objects find the point estimates from the different AF measures. This method work for only one measure `c("M0","H","A","hd","hdk")` (Note that contribution measure do no have confidence intervals). Then, user should subset the data frame with the estimates by the chosen measure (including other preferred categories, i.e., poverty cut-off, subgroup, etc.)

Value

Coefficients extracted from the model `lframe` object.

Author(s)

Ignacio Girela

See Also

`confint`, and `summary` methods, and `mpitb.est` function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                  d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

confint.cotframe *Extract the confidence intervals from the the estimated changes over time measures*

Description

Extract the confidence intervals from the the estimated changes over time measures

Usage

```
## S3 method for class 'cotframe'
confint(object, parm = "coefficient", level = 0.95, ...)
```

Arguments

`object`	a "cotframe"-class object
`parm`	"coefficient". Confidence intervals are only available for AF measure point estimates.
`level`	the confidence level required.
`...`	additional argument(s) for methods.

Details

The `confint` method for "cotframe"-class objects find the confidence intervals from the changes over time estimates data frame. This method work for only one measure `c("M0","H","A","hd","hdk")`. Then, user should subset the data frame with the estimates by the chosen measure (including other preferred categories, i.e., poverty cut-off, subgroup, etc.)

Value

Confidence intervals extracted from the model `cotframe` object.

Author(s)

Ignacio Girela

See Also

coef, and summary methods, and mpitb.est function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

---

confint.lframe *Extract the confidence intervals from the the estimated cross-sectional measures*

Description

Extract the confidence intervals from the the estimated cross-sectional measures

Usage

```
## S3 method for class 'lframe'
confint(object, parm = "coefficient", level = 0.95, ...)
```

Arguments

`object`	a "lframe"-class object
`parm`	"coefficient". Confidence intervals are only available for AF measure point estimates.
`level`	the confidence level required.
`...`	additional argument(s) for methods.

Details

The `confint` method for "lframe"-class objects find the confidence intervals from the different AF measures estimates data frame. This method work for only one measure `c("M0","H","A","hd","hdk")` (Note that contribution measure do no have confidence intervals). Then, user should subset the data frame with the estimates by the chosen measure (including other preferred categories, i.e., poverty cut-off, subgroup, etc.)

Value

Confidence intervals extracted from the model `lframe` object.

Author(s)

Ignacio Girela

See Also

`coef`, and `summary` methods, and `mpitb.est` function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
```

```
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

mpitb.est *Estimate multidimensional poverty indices based on the Alkire-Foster method*

Description

Estimate multidimensional poverty indices (MPI) based on the Alkire-Foster (AF) method including disaggregated cross-sectional and changes over time estimates as well as quantities such as standard errors and confidences intervals (accounting for the household survey design).

Usage

```
mpitb.est(set, ...)

## S3 method for class 'mpitb_set'
mpitb.est(
  set,
  klist = NULL,
  weights = "equal",
  measures = c("M0", "H", "A"),
  indmeasures = c("hd", "hdk", "actb", "pctb"),
  indklist = NULL,
  over = NULL,
  ...,
  cotyear = NULL,
  tvar = NULL,
  cotmeasures = c("M0", "H", "A", "hd", "hdk"),
  ann = FALSE,
  cotklist = NULL,
  cotoptions = "total",
  noraw = FALSE,
  nooverall = FALSE,
  level = 0.95,
  multicore = getOption("mpitb.multicore"),
  verbose = TRUE
)
```

Arguments

set	a "mpitb_set"-class object in which data, indicators, names and description have been specified.

`...` other arguments

`klist` a numeric vector representing the poverty cut-offs for calculating the MPI. Should be values between 1 and 100.

`weights` either a character value or a numeric vector. If "equal", it automatically calculated equal nested weights.

`measures` a character vector with the MPI and partial measures. Default include all the measures `c("M0","A","H")`. For more information, see Details section below.

`indmeasures` a character vector with the indicator-specific measures. Default include all the measures `c("hd", "hdk", "actb", "pctb")`. For more information, see Details section below.

`indklist` a numeric vector representing the poverty cut-offs for calculating indicator-specific measures. Should be values between 1 and 100. If `NULL`, it will be equal to `klist`.

`over` a character vector with columns names of the population subgroups in data.

`cotyear` a character value containing the column name of the years variable in the data. This argument is required if annualized changes over time measure are desired.

`tvar` a character value containing the column name of the time ID variable in the data. This argument determines if changes over time are calculated.

`cotmeasures` a character vector with the changes over time measures. Default include all the measures `c("M0","A","H","hd", "hdk")`. For more information, see Details section below.

`ann` logical. If `TRUE`, annualized changes over time measure are estimated. If `FALSE`, only non-annualized changes over time are calculated. Default value is `FALSE`.

`cotklist` a numeric vector representing the poverty cut-offs for calculating changes over time measures. Should be values between 1 and 100. If `NULL`, it will be equal to `klist`.

`cotoptions` a character vector. If "total", estimates change over the total period of observation, i.e. from the first year of observation to the last year of observation. If "insequence", then estimates all consecutive (i.e. year-to-year) changes. The default is "total"

`noraw` logical. If `TRUE`, non-annualized changes over time measure are not estimated. Default is `FALSE`.

`nooverall` logical. If `TRUE`, estimations over all the observations are omitted, e.g., national level calculations, and only measure for the specified subgroups are estimated. Default is `FALSE`.

`level` numeric value with the desired confidence level for the confidence interval calculations in decimal format. Default value is 0.95.

`multicore` logical. Use `multicore` package for parallel estimation by measure and poverty cut-off over multiple processors? It uses forking approach. See Details below.

`verbose` logical. If TRUE, print information messages to the console so the user can control the dimension and indicators weights and which measures are being estimated in the process. If FALSE, these messages are ommited.

Details

This functions is a S3 method for "mpitb_set" class. Hence, the project has to be correctly specified with `mpitb.set()` function previously.

The vector of poverty cut-offs (k) in percentage point, i.e., numbers between 1 and 100. Although the deprivation score ($c_i = \sum_{i=1}^{n} w_j g_{ij}^0$) is a real-valued function, given the weights, it will assume a limited number of values. The same occurs with the censored deprivation score. Therefore, despite accepting infinite number of values, results may not vary with close values of k. For this reason, it is recommended to use a very limited number of poverty cut-offs for the analysis.

If nothing is passed to `weights` argument, equal nested weights are calculated by dimension and indicator. In this case, it is preferred to pass indicators as a list in `mpitb.set()`. If the user wants to pass another weighting scheme, she should first pass the indicators as a character vector in `mpitb.set()` and then pass a numeric vector in `weights` such that the elements of this vector match with the vector of indicators and all the weights sum up to 1.

To specify the population subgroups (e.g., living area, sex, etc.) and estimate the disaggregated measures by each level of the subgroup, the user should pass the column names of the population subgroups in the data using `over` argument. If `over` is `NULL`, the measure are estimate using all the observations (e.g., national-level). If population subgroups are specified and `nooverall` is set to `TRUE`, aggregate (or national-level) estimates will not be produced.

Details on the AF measures estimation:

Available measures include the Adjusted Headcount Ratio (M_0), the Incidence (H) and the Intensity of poverty (A), as well as other indicator-specific measures such as the uncensored headcount ratio (h_j), the censored headcount ratio ($h_j(k)$) and the absolute and percentage contribution.

The three first partial measures are pass in `measures` argument. By default, `mpitb.est` calculates every measure `c("M0","H","A")`. The poverty cut-off (k) for these measures estimation is specified in `klist` argument.

The indicator-specific measure are passed in `indmeasures` argument. By default, `mpitb.est` calculates every measure `c("hd","hdk","actb","pctb")`. The poverty cut-off (k) for these measures estimation is specified in `indklist` argument. If `indklist` is `NULL`, poverty cut-offs in `klist` is used. The absolute contribution `c("actb")` cannot be estimated without also passing the censored headcount ratios of each indicator `c("hdk")` and the percentage contribution cannot be calculated without `c("hdk")` and `c("M0")` passed in `measures` argument.

If any of these arguments is `NULL`, `mpitb.est()` skips these measures. So it is useful for avoid calculating unnecessary estimations. For example, if `measures = c("H","A")` and `indmeasures = NULL`, only the Incidence and the Intensity will be estimated.

Details on changes over time measures:

The user can decide which AF measure changes over time she want to study. This is set in `cotmeasures`. By default it calculates all the measure, except contributions, i.e., `cotmeasure = c("M0","A","H","hd","hdk")`. It would be important to check this argument in order to save time. The poverty cut-off (k) for these measures estimation is specified in `cotklist` argument. If `cotklist` is `NULL`, poverty cut-offs in `klist` is used. The standard errors of the changes over time measures is estimated using Delta method.

For calculating any point estimate for each time period and any change over time measure, `tvar` should not be `NULL`. This argument should be a character with the column name that references the time period $t = 1, \ldots, T$.

Changes over time measure can also be annualized. For such measure, information about the years is needed. `cotyear` should be a character with the column name that have information about the years. Decimal digits are permitted. Argument `ann` is a logical value. If `TRUE`, annualized measures are calculated. If `cotyear` is passed, `ann` is automatically set to `TRUE`. If the former is not `NULL` and `ann` is `FALSE`, only non-annualized measures are estimated. If only annualized measure are under study, the user can switch `noraw` to `TRUE` to avoid estimating non-annualized changes.

Finally, if there are more than two years survey rounds, the user can decide if estimate the change over the total period of observation, i.e. from the first year of observation to the last year of observation or year-to-year changes. To do the former, `cotoptions = "total"` whereas for the latter case, `cotoptions = "insequence"`. By default, `cotoptions = "total"` to avoid unnecessary estimations.

Some details on other arguments and estimations:

The package includes the possibility to do parallel calculations over all the measures and poverty cut-offs. If `multicore` is `TRUE`, the package proceeds with parallel estimations. Caveat: this package uses Forking method for parallelization which is only available on Unix-like systems (including Linux), i.e., Windows users cannot benefit from parallelization.

For every measure the standard errors and confidence intervals are estimated. The former are estimated taking into account the survey structure whereas the latter are estimated considering measures as proportions using `svyciprop()` function from "survey" R package (it uses the "logit" method which consists of fitting a logistic regression model and computes a Wald-type interval on the log-odds scale, which is then transformed to the probability scale).

Value

An object with S3 class "mpitb_est" containing two data frames with the estimates of the cross-sectional measures ("lframe"-class) and changes over time ("cotframe"-class).

Author(s)

Ignacio Girela

References

Alkire, S., Foster, J. E., Seth, S., Santos, M. E., Roche, J., & Ballon, P. (2015). Multidimensional poverty measurement and analysis. Oxford University Press.

Alkire, S., Roche, J. M., & Vaz, A. (2017). Changes over time in multidimensional poverty: Methodology and results for 34 countries. World Development, 94, 232-249. doi:10.1016/j.worlddev.2017.01.011

Suppa, N. (2023). mpitb: A toolbox for multidimensional poverty indices. The Stata Journal, 23(3), 625-657. doi:10.1177/1536867X231195286

See Also

`coef`, `confint`, and `summary` methods, and `mpitb.set` function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

---

mpitb.set *Set the specification of the Multidimensional Poverty Measurement and Analysis project*

Description

Set the specification of the Multidimensional Poverty Measurement and Analysis project

Usage

```
mpitb.set(data, ...)

## S3 method for class 'survey.design2'
mpitb.set(data, indicators, ..., name = "unnamed", desc = "desc.")

## S3 method for class 'data.frame'
mpitb.set(data, indicators, ..., name = "unnamed", desc = "desc.")
```

Arguments

`data` a "survey.design2"-class object where a complex survey design was previously specified. Can be a "data.frame" but it is coerced to "survey.design2" class assuming equal probabilities.

`...` other arguments

`indicators` a list or character vector containing the names of the indicators. If it is a list, the element represents the dimension which character vector with their corresponding indicators. At most 10 dimensions are allowed. The indicators should belong to columns names of ‘data‘. See Details below.

`name` a character containing a desired specification of the project name. It also serves as an ID and it is recommended to use short names (at most 10 characters are permitted).

`desc` a character containing a desired specification of the project description.

Details

The data passed to `data` argument assumes that the indicators columns is the deprivation matrix $\mathbf{g}^0 = [g^0_{ij}]$, where $g^0_{ij} = 1$ if the i-th person is deprived in the j-th indicator and $g^0_{ij} = 0$ otherwise, for $i = 1, \ldots, n$ and $j = 1, \ldots, d$. This argument should be a "survey.design2"-class object in which the complex survey design structure was previously specified using `svydesign` of survey package. If `data` is a "data.frame", it is coerced to a "survey.design2"-class object assuming equal probabilities, which is rarely used in household surveys.

These columns should not contain any missing value. For estimating the multidimensional poverty measures, the R survey package supports missing values for calculating the point estimation but it would not be able to calculate the standard error and, therefore, the confidence intervals.

The `indicators` argument should contain the names of indicators corresponding to the columns names in `data`. It is advisable to pass a list object where each element is the dimension and contain the character string with the indicators name because the package can calculate the nested equal weights automatically in the subsequent estimations. At most 10 dimensions are allowed. It can also be a character string. In this later case, if nested weights across dimensions is used, the user should be careful and specified later the corresponding weights by hand.

Finally, `name` and `desc` arguments are useful for identifying each MPI setting while working in a multidimensional poverty measurement and analysis project. Names with more than 10 characters are not allowed for tidiness purposes.

Value

"mpitb_set"-class object

Author(s)

Ignacio Girela

References

Alkire, S., Foster, J. E., Seth, S., Santos, M. E., Roche, J., & Ballon, P. (2015). Multidimensional poverty measurement and analysis. Oxford University Press.

Alkire, S., Roche, J. M., & Vaz, A. (2017). Changes over time in multidimensional poverty: Methodology and results for 34 countries. World Development, 94, 232-249. doi:10.1016/j.worlddev.2017.01.011

Suppa, N. (2023). mpitb: A toolbox for multidimensional poverty indices. The Stata Journal, 23(3), 625-657. doi:10.1177/1536867X231195286

See Also

`mpitb.est` function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta, t==1)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")
```

---

print.summary.cotframe *Print Summary function of summary.cotframe object*

Description

Print Summary function of summary.cotframe object

Usage

```
## S3 method for class 'summary.cotframe'
print(
  x,
  digits = max(4, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

`x`	the "summary.cotframe"-class object
`digits`	controls number of digits printed in output.
`signif.stars`	should significance stars be printed alongside output.
`...`	optional arguments

Value

summarized object with nice format

print.summary.lframe *Print Summary function of summary.lframe object*

Description

Print Summary function of summary.lframe object

Usage

```
## S3 method for class 'summary.lframe'
print(
  x,
  digits = max(4, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

`x`	the "summary.lframe"-class object
`digits`	controls number of digits printed in output.
`signif.stars`	should significance stars be printed alongside output.
`...`	optional arguments

Value

summarized object with nice format

summary.cotframe *Summary function of the estimates of the changes over time measures*

Description

Summary function of the estimates of the changes over time measures

Usage

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

Arguments

object	the "cotframe"-class object
...	other arguments

Details

The summary method for "cotframe"-class objects with coefficients, standard errors and the corresponding t-values and the p-value for a two sided test, $H_0 : \beta = 0$ being β any measure, with the 'significance stars'. This method work for only one measure c("M0","H","A","hd","hdk") and either relative or absolute measure, either non-annualized and annualized measure and only one poverty cut-off. Then, user should subset the data frame to obtain the desired summary.

Value

summary.cotframe a list of summary information for the estimated AF measure

Author(s)

Ignacio Girela

See Also

coef, and confint methods, and mpitb.est function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
```

```
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

summary.lframe *Summary function of the estimates of the cross-sectional AF measures*

Description

Summary function of the estimates of the cross-sectional AF measures

Usage

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

Arguments

object	the "lframe"-class object
...	other arguments

Details

The summary method for "lframe"-class objects with coefficients, standard errors and the corresponding t-values and the p-value for a two sided test, $H_0 : \beta = 0$ being β any measure, with the 'significance stars'. This method work for only one measure c("M0","H","A","hd","hdk"). Then, user should subset the data frame to obtain the desired summary.

Value

summary.lframe a list of summary information for the estimated AF measure

Author(s)

Ignacio Girela

See Also

coef, and confint methods, and mpitb.est function.

Examples

```
library(mpitbR)

data <- subset(syn_cdta)
data <- na.omit(data)

svydata <- survey::svydesign(id=~psu, weights = ~weight, strata = ~stratum, data = data)

indicators <- list(d1 = c("d_nutr","d_cm"),
                   d2 = c("d_satt","d_educ"),
                   d3 = c("d_elct","d_sani","d_wtr","d_hsg","d_ckfl","d_asst"))

# Specify mpitb project
set <- mpitb.set(svydata, indicators = indicators, name = "myname", desc = "pref. desc")

# Estimate the cross-sectional MPI and compare non-annualized changes over time
est <- mpitb.est(set, klist = c(33), measures = "M0", indmeasures = NULL,
                 tvar = "t", cotmeasures = "M0",
                 weights = "equal", over = c("area"))

coef(subset(est$lframe, measure == "M0" & t == 1))
confint(subset(est$lframe, measure == "M0" & t == 1))
summary(subset(est$lframe, measure == "M0" & t == 1))

coef(subset(est$cotframe, measure == "M0"))
confint(subset(est$cotframe, measure == "M0"))
summary(subset(est$cotframe, measure == "M0" & ctype == "abs" & ann == 0 & k == 33))
```

syn_cdta — *Synthetic dataset with a typical household survey design*

Description

This is a synthetic dataset drawn from the Stata mpitb package. This is the data used in the provided examples.

Usage

```
syn_cdta
```

Format

An object of class `data.frame` with 15000 rows and 17 columns.

Author(s)

Nicolai Suppa

References

doi:10.1177/1536867X231195286

Index