Package: MSCMT (via r-universe)

October 22, 2026

Type Package

Encoding UTF-8

Version 1.4.4

Date 2026-06-15

Title Multivariate Synthetic Control Method Using Time Series

Depends R (>= 3.2.0)

Imports stats, utils, parallel, lpSolve, ggplot2, lpSolveAPI, Rglpk, Rdpack, rlang

Suggests Synth, DEoptim, rgenoud, DEoptimR, GenSA, GA, soma, cmaes, Rmalschains, NMOF, nloptr, pso, LowRankQP, kernlab, reshape, knitr, rmarkdown

Description Three generalizations of the synthetic control method (which has already an implementation in package 'Synth') are implemented: first, 'MSCMT' allows for using multiple outcome variables, second, time series can be supplied as economic predictors, and third, a well-defined cross-validation approach can be used. Much effort has been taken to make the implementation as stable as possible (including edge cases) without losing computational efficiency. A detailed description of the main algorithms is given in Becker and Klößner (2018) <doi:10.1016/j.ecosta.2017.08.002>.

License GPL

Copyright inst/COPYRIGHTS

VignetteBuilder knitr

RdMacros Rdpack

BuildVignettes yes

Config/roxygen2/version 8.0.0

URL https://github.com/mabe0033/MSCMT

BugReports https://github.com/mabe0033/MSCMT/issues

NeedsCompilation yes

Author Martin Becker [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2336-9751>), Stefan Klößner [aut],
Karline Soetaert [com], Jack Dongarra [cph], R.J. Hanson [cph],
K.H. Haskell [cph], Cleve Moler [cph], LAPACK authors [cph]

Maintainer Martin Becker <martin.becker@mx.uni-saarland.de>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-15 08:20:02 UTC

RemoteUrl https://github.com/cran/MSCMT

RemoteRef HEAD

RemoteSha 3d38c480000fd1f1f6985e6c063df588276255db

Contents

compare 2
did 3
ggplot.mscmt 4
improveSynth 8
listFromLong 10
mscmt 11
plot.mscmt 17
ppratio 19
print.mscmt 20
pvalue 21

Index **23**

`compare` *Compare MSCMT estimation results*

Description

`compare` collects estimation results from `mscmt` for comparison purposes.

Usage

```
compare(..., auto.name.prefix = "")
```

Arguments

`...`	Objects of class `"mscmt"` or (a) list(s) containing objects of class `"mscmt"`.
`auto.name.prefix`	A character string (default: `""`) internally used to facilitate automatic naming in nested lists of unnamed estimation results.

Details

compare collects (potentially many) estimation results from mscmt in a special object of class "mscmt", which includes a component "comparison" where the different estimation results are aggregated. This aggregated information is used by the ggplot.mscmt and print.mscmt methods to present summaries of the different results.

Value

An object of class "mscmt", which itself contains the individual estimation results as well as a component "comparison" with aggregated information.

---

did *Difference-in-difference estimator based on SCM*

Description

did calculates difference-in-difference estimators based on SCM.

Usage

```
did(
  x,
  what,
  range.pre,
  range.post,
  alternative = c("two.sided", "less", "greater"),
  exclude.ratio = Inf
)
```

Arguments

x	An object of class "mscmt", usually obtained as the result of a call to function mscmt.
what	A character vector. Name of the variable to be considered. If missing, the (first) dependent variable will be used.
range.pre	A vector of length 2 defining the range of the pre-treatment period with start and end time given as • annual dates, if the format of start/end time is "dddd", e.g. "2016", • quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1", • monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10", • weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23", • daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

corresponding to the format of the respective column of the `times.dep` argument of `mscmt`. If missing, the corresponding column of `times.dep` will be used.

`range.post` A vector of length 2 defining the range of the post-treatment period with start and end time given as

- annual dates, if the format of start/end time is "dddd", e.g. "2016",
- quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1",
- monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10",
- weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23",
- daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

corresponding to the format of the respective column of the `times.dep` argument of `mscmt`. Will be guessed if missing.

`alternative` A character string giving the alternative of the test. Either `"two.sided"` (default), `"less"`, or `"greater"`.

`exclude.ratio` A numerical scalar (default: `Inf`). When calculating the p-value, control units with an average pre-treatment gap of more then `exclude.ratio` times the average pre-treatment gap of the treated unit are excluded from the analysis.

Details

`did` calculates difference-in-difference estimators with corresponding p-values (if results of a placebo study are present) based on the Synthetic Control Method.

Value

A list with components `effect.size`, `average.pre` and `average.post`. If `x` contains the results of a placebo study, three components `p.value`, `rank`, and `excluded` (with the names of the excluded units) are included additionally.

Examples

```
## Not run:
## for an example, see the main package vignette:
 vignette("WorkingWithMSCMT",package="MSCMT")

## End(Not run)
```

---

ggplot.mscmt *Plotting Results of mscmt with ggplot2*

Description

`ggplot.mscmt` plots results of `mscmt` based on `ggplot`.

Usage

```
## S3 method for class 'mscmt'
ggplot(
  data,
  mapping = aes(),
  what,
  type = c("gaps", "comparison", "placebo.gaps", "placebo.data", "p.value"),
  treatment.time,
  zero.line = TRUE,
  ylab,
  xlab = "Date",
  main,
  col,
  lty,
  lwd,
  legend = TRUE,
  bw = FALSE,
  date.format,
  unit.name,
  full.legend = TRUE,
  include.smooth = FALSE,
  include.mean = FALSE,
  include.synth = FALSE,
  draw.estwindow = TRUE,
  what.set,
  limits = NULL,
  alpha = 1,
  alpha.min = 0.1,
  exclude.units = NULL,
  exclude.ratio = Inf,
  ratio.type = c("rmspe", "mspe"),
  alternative = c("two.sided", "less", "greater"),
  draw.points = TRUE,
  control.name = "control units",
  size = 1,
  treated.name = "treated unit",
  labels = c("actual data", "synthesized data"),
  ...,
  environment = parent.frame()
)
```

Arguments

data	An object of class "mscmt", usually obtained as the result of a call to function mscmt.
mapping	An object necessary to match the definition of the ggplot generic (passed to ggplot as is). Defaults to aes().

what	A character vector. Name(s) of the variables to be plotted. If missing, the (first) dependent variable will be used.
type	A character scalar denoting the type of the plot containing either "gaps", "comparison", "placebo.gaps", "placebo.data", or "p.value". Partial matching allowed, defaults to "placebo.gaps", if results of a placebo study are present, and to "gaps", else.
treatment.time	An optional scalar (numeric, character, or Date) giving the treatment time. If treatment.time is numeric, Jan 01 of that year will be used. If treatment.time is a character string, it will be converted to a Date and must thus be in an unambiguous format. A vertical dotted line at the given point in time is included in the plot.
zero.line	A logical scalar. If TRUE (default), a horizontal dotted line (at zero level) is plotted for "gaps" and "placebo.gaps" plots.
ylab	Optional label for the y-axis, automatically generated if missing.
xlab	Optional label for the x-axis, defaults to "Date".
main	Optional main title for the plot, automatically generated if missing.
col	Optional character vector with length 1 (for gaps plots) or 2 (for all other plot types). For comparison plots, col contains the colours for the actual and synthesized data, for placebo.plots (with full.legend==FALSE), col contains the colours for the treated unit and the control units. Automatically generated if missing.
lty	Optional numerical vector with length 1 (for gaps plots) or 2 (for all other plot types). For comparison plots, lty contains the linetypes for the actual and synthesized data, for placebo.plots (with full.legend==FALSE), col contains the linetypes for the treated unit and the control units. Automatically generated if missing.
lwd	Optional numerical vector with length 1 (for gaps plots) or 2 (for all other plot types). For comparison plots, lty contains the linewidths for the actual and synthesized data, for placebo.plots (with full.legend==FALSE), col contains the linewidths for the treated unit and the control units. Automatically generated if missing.
legend	A logical scalar. If TRUE (default), a legend is included in the plot.
bw	A logical scalar. If FALSE (default), the automatically generated colours and line types are optimized for a colour plot, if TRUE, the automatic colours and line types are set for a black and white plot.
date.format	A character string giving the format for the tick labels of the x axis as documented in strptime. Defaults to "%b %y" or "%Y", depending on the granularity of the data.
unit.name	A character string with the title of the legend for comparison and placebo plots. Defaults to "Estimation" for comparison and "Unit" for placebo plots.
full.legend	A logical scalar. If TRUE (default), a full legend of all units (donors) is constructed. If FALSE, only the treated and the control units are distinguished.
include.smooth	A logical scalar. If TRUE, a geometric smoother based on the control units is added to placebo plots. Default: FALSE.

include.mean — A logical scalar. If TRUE, the arithmetic mean of all control units is added to placebo plots. Default: FALSE.

include.synth — A logical scalar. If TRUE, the synthesized data for the treated unit are added to plots of type "placebo.data". Defaults to FALSE.

draw.estwindow — A logical scalar. If TRUE (default), the time range containing all optimization periods is shaded in the corresponding plots.

what.set — An optional character string for a convenient selection of multiple variables. Accepted values are "dependents", "predictors", and "all", which collects all dependent, all predictor, or all variables of both types, respectively. Overrides parameter what (if the latter is present).

limits — An optional vector of length 2 giving the range of the plot or NULL. If limits is numeric, Jan 01 of the corresponding years will be used. If limits is of type character, both strings will be converted to Dates (via as.Date) and must thus be in an unambiguous format.

alpha — Either a numerical scalar, a numerical vector of length corresponding to the number of units, or the character string "auto". If alpha is a numerical scalar (default with value 1), a fixed value for the alpha channel (transparency) is included for all units in placebo plots. If alpha is numeric and has length corresponding to the number of units, these values are assigned as alpha channel to the individual units. If "auto", the alpha channel information is obtained from the w weights of the control units.

alpha.min — A numerical scalar (default: 0.1). If alpha is set to "auto", the individual alpha channel information for control unit i is set to alpha.min + (1-alpha.min) * w[i].

exclude.units — An optional (default: NULL) character vector with names for control units which shall be excluded from placebo plots and p-value calculations.

exclude.ratio — A numeric scalar (default: Inf). Control units with a pre-treatment (r)mspe of more than exclude.ratio times the pre-treatment (r)mspe of the treated unit are excluded from placebo plots and p-value calculations.

ratio.type — A character string. Either rmspe (default) or mspe. Selects whether root mean squared errors or mean squared errors are considered for the exclusion of control units (see exclude.ratio).

alternative — A character string giving the alternative of the test for plots of type "p.value". Either "two.sided" (default), "less", or "greater".

draw.points — A logical scalar. If TRUE (default), points are added to the line plots to enhance visibility.

control.name — A character string for the naming of the non-treated units in placebo plots. Defaults to "control units".

size — A numerical scalar (default: 1). If draw.points is TRUE (default), size specifies the size of the points.

treated.name — A character string giving the label for the treated unit. Defaults to "treated unit".

labels — A character vector of length 2 giving the labels for the actual and synthesized data. Defaults to c("actual data","synthsized data").

...	Necessary to match the definition of the "ggplot" generic (passed to ggplot as is).
environment	An object necessary to match the definition of the "ggplot" generic (passed to ggplot as is). Defaults to parent.frame().

Details

A unified plot method for gaps plots, comparison of treated and synthetic values, as well as plots for placebo studies, based on ggplot. ggplot.mscmt is the preferred plot method and has more functionality than plot.mscmt.

Value

An object of class ggplot.

---

improveSynth *Check (and Improve) Results of Package Synth*

Description

improveSynth checks the results of synth for feasibility and optimality and tries to find a better solution.

Usage

```
improveSynth(
  synth.out,
  dataprep.out,
  lb = 1e-08,
  tol = 1e-05,
  verbose = TRUE,
  seed = 1,
  ...
)
```

Arguments

synth.out	A result of synth from package 'Synth'.
dataprep.out	The input of function synth which led to synth.out.
lb	A numerical scalar (default: 1e-8), corresponding to the lower bound for the outer optimization.
tol	A numerical scalar (default: 1e-5). If the relative *and* absolute improvement of loss.v and loss.w, respectively, exceed tol, this is reported (if verbose is TRUE). Better optima are always looked for (independent of the value of tol).
verbose	A logical scalar. Should the ouput be verbose (defaults to TRUE).

seed	A numerical vector or NULL. See the corresponding documentation for mscmt. Defaults to 1 in order to provide reproducibility of the results.
...	Further arguments to mscmt. Supported arguments are check.global, inner.optim, inner.opar, outer.optim, and outer.opar.

Details

Performing SCM means solving a nested optimization problem. Depending on the validity of the results of the inner optimization, SCM may produce

- invalid or infeasible results, if the vector w of donor weights reported as the result of the inner optimization is in fact not optimal, ie. produces too large loss.w,
- suboptimal results, if the vector v of predictor weights reported as the result of the outer optimization is in fact not optimal (which may be caused by shortcomings of the inner optimization).

improveSynth first checks synth.out for feasibility and then tries to find a feasible and optimal solution by applying the optimization methods of package MSCMT to dataprep.out (with default settings, more flexibility will probably be added in a future release).

Value

An updated version of synth.out, where solution.v, solution.w, loss.v, and loss.w are replaced by the optimum obtained by package 'MSCMT' and all other components of synth.out are removed.

Examples

```
## Not run:
## check whether package 'Synth' is available
if (require("Synth")) {

## process first example of function "synth" in package 'Synth'
## (comments are removed):

  data(synth.data)
  dataprep.out<-
    dataprep(
     foo = synth.data,
     predictors = c("X1", "X2", "X3"),
     predictors.op = "mean",
     dependent = "Y",
     unit.variable = "unit.num",
     time.variable = "year",
     special.predictors = list(
        list("Y", 1991, "mean"),
        list("Y", 1985, "mean"),
        list("Y", 1980, "mean")
                               ),
     treatment.identifier = 7,
     controls.identifier = c(29, 2, 13, 17, 32, 38),
```

```
    time.predictors.prior = c(1984:1989),
    time.optimize.ssr = c(1984:1990),
    unit.names.variable = "name",
    time.plot = 1984:1996
    )

  synth.out <- synth(dataprep.out)

## check and (try to) improve these results:
  synth2.out <- improveSynth(synth.out,dataprep.out)
}

## End(Not run)
```

`listFromLong` *Convert Long Format to List Format*

Description

`listFromLong` converts long to list format.

Usage

```
listFromLong(
  foo,
  unit.variable,
  time.variable,
  unit.names.variable = NULL,
  exclude.columns = NULL
)
```

Arguments

`foo`	A `data.frame` containing the data in "long" format.
`unit.variable`	Either a numeric scalar with the column number (in `foo`) containing the units or a character scalar with the corresponding column name in `foo`.
`time.variable`	Either a numeric scalar with the column number (in `foo`) containing the times or a character scalar with the corresponding column name in `foo`.
`unit.names.variable`	Optional. If not `NULL`, either a numeric scalar with the column number (in `foo`) containing the unit names or a character scalar with the corresponding column name in `foo`. Must match with the units defined by `unit.variable` (if not `NULL`).
`exclude.columns`	Optional (defaults to `NULL`). Numeric vector with column numbers of `foo` to be excluded from the conversion.

Details

listFromLong is a convenience function to convert long format (in a data.frame, as used by package 'Synth') to list format, where data is stored as a list of matrices.

Most parameter names are named after their equivalents in the dataprep function of package 'Synth'.

Value

A list of matrices with rows corresponding to the times and columns corresponding to the unit (or unit names, respectively) for all columns of foo which are neither excluded nor have a special role as time, unit, or unit names variable.

Examples

```
if (require("Synth")) {
  data(basque)
  Basque <- listFromLong(basque, unit.variable="regionno",
                         time.variable="year",
                         unit.names.variable="regionname")
  names(Basque)
  head(Basque$gdpcap)
}
```

---

mscmt *Multivariate SCM Using Time Series*

Description

mscmt performs the Multivariate Synthetic Control Method Using Time Series.

Usage

```
mscmt(
  data,
  treatment.identifier = NULL,
  controls.identifier = NULL,
  times.dep = NULL,
  times.pred = NULL,
  agg.fns = NULL,
  placebo = FALSE,
  placebo.with.treated = FALSE,
  univariate = FALSE,
  univariate.with.dependent = FALSE,
  check.global = TRUE,
  inner.optim = "wnnlsOpt",
  inner.opar = list(),
  outer.optim = "DEoptC",
```

```
  outer.par = list(),
  outer.opar = list(),
  std.v = c("sum", "mean", "min", "max"),
  alpha = NULL,
  beta = NULL,
  gamma = NULL,
  return.ts = TRUE,
  single.v = FALSE,
  verbose = TRUE,
  debug = FALSE,
  seed = NULL,
  cl = NULL,
  times.pred.training = NULL,
  times.dep.validation = NULL,
  v.special = integer(),
  cv.alpha = 0,
  spec.search.treated = FALSE,
  spec.search.placebos = FALSE
)
```

Arguments

`data` Typically, a list of matrices with rows corresponding to times and columns corresponding to units for all relevant features (dependent as well as predictor variables, identified by the list elements' names). This might be the result of converting from a `data.frame` by using function `listFromLong`.

For convenience, `data` may alternatively be the result of function `dataprep` of package `'Synth'`. In this case, the parameters `treatment.identifier`, `controls.identifier`, `times.dep`, `times.pred`, and `agg.fns` are ignored, as these input parameters are generated automatically from `data`. The parameters `univariate`, `alpha`, `beta`, and `gamma` are ignored by fixing them to their defaults. Using results of `dataprep` is experimental, because the automatic generation of input parameters may fail due to lack of information contained in results of `dataprep`.

`treatment.identifier`

A character scalar containing the name of the treated unit. Must be contained in the column names of the matrices in `data`.

`controls.identifier`

A character vector containing the names of at least two control units. Entries must be contained in the column names of the matrices in `data`.

`times.dep` A matrix with two rows (containing start times in the first and end times in the second row) and one column for each dependent variable, where the column names must exactly match the names of the corresponding dependent variables. A sequence of dates with the given start and end times of

- annual dates, if the format of start/end time is "dddd", e.g. "2016",
- quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1",
- monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10",

- weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23",
- daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

will be constructed; these dates are looked for in the row names of the respective matrices in `data`. In applications with cross-validation, `times.dep` belongs to the main period.

`times.pred` A matrix with two rows (containing start times in the first and end times in the second row) and one column for each predictor variable, where the column names must exactly match the names of the corresponding predictor variables. A sequence of dates with the given start and end times of

- annual dates, if the format of start/end time is "dddd", e.g. "2016",
- quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1",
- monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10",
- weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23",
- daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

will be constructed; these dates are looked for in the row names of the respective matrices in `data`. In applications with cross-validation, `times.pred` belongs to the main period.

`agg.fns` Either `NULL` (default) or a character vector containing one name of an aggregation function for each predictor variable (i.e., each column of `times.pred`). The character string "id" may be used as a "no-op" aggregation. Each aggregation function must accept a numeric vector and return either a numeric scalar ("classical" MSCM) or a numeric vector (leading to MSCM*T* if length of vector is at least two).

`placebo` A logical scalar. If `TRUE`, a placebo study is performed where, apart from the treated unit, each control unit is considered as treated unit in separate optimizations. Defaults to `FALSE`. Depending on the number of control units and the complexity of the problem, placebo studies may take a long time to finish.

`placebo.with.treated`
A logical scalar. If `TRUE`, the treated unit is included as control unit (for other treated units in placebo studies). Defaults to `FALSE`.

`univariate` A logical scalar. If `TRUE`, a series of univariate SCMT optimizations is done (instead of one MSCMT optimization) even if there is more than one dependent variable. Defaults to `FALSE`.

`univariate.with.dependent`
A logical scalar. If `TRUE` (and if `univariate` is also `TRUE`), all dependent variables (contained in the column names of `times.dep`) apart from the current (real) dependent variable are included as predictors in the series of univariate SCMT optimizations. Defaults to `FALSE`.

`check.global` A logical scalar. If `TRUE` (default), a check for the feasibility of the unrestricted outer optimum (where actually no restrictions are imposed by the predictor variables) is made before starting the actual optimization procedure.

`inner.optim` A character scalar containing the name of the optimization method for the inner optimization. Defaults to `"wnnlsOpt"`, which (currently) is the only supported implementation, because it outperforms all other inner optimizers we are aware of. `"ipopOpt"`, which uses `ipop`, and `LowRankQPOpt`, which uses `LowRankQP` as inner optimizer, have experimental support for benchmark purposes.

`inner.opar` A list containing further parameters for the inner optimizer. Defaults to the empty list. (For `"wnnlsOpt"`, there are no meaningful further parameters.)

`outer.optim` A character vector containing the name(s) of the optimization method(s) for the outer optimization. Defaults to `"DEoptC"`, which (currently) is the recommended global optimizer. The optimizers currently supported can be found in the documentation of parameter `outer.opar`, where the default control parameters for the various optimizers are listed. If `outer.optim` has length greater than 1, one optimization is invoked for each outer optimizer (and, potentially, each random seed, see below), and the best result is used.

`outer.par` A list containing further parameters for the outer optimization procedure. Defaults to the empty list. Entries in this list may override the following hard-coded general defaults:

- `lb=1e-8`, corresponding to the lower bound for the ratio of predictor weights,
- `opt.separate=TRUE`, corresponding to an improved outer optimization where each predictor is treated as the (potentially) most important predictor (i.e. with maximal weight) in separate optimizations (one for each predictor), see [1].

`outer.opar` A list (or a list of lists, if `outer.optim` has length greater than 1) containing further parameters for the outer optimizer(s). Defaults to the empty list. Entries in this list may override the following hard-coded defaults for the individual optimizers, which are quite modest concerning the computing time. `dim` is a variable holding the problem dimension, typically the number of predictors minus one.

Optimizer	**Package**	**Default parameters**
DEoptC	MSCMT	nG=500, nP=20*dim, waitgen=100, minimpr=1e-14, F=0.5, CR=0.9
cma_es	cmaes	maxit=2500
crs	nloptr	maxeval=2.5e4, xtol_rel=1e-14, population=20*dim, algorithm="NLOPT_GN_CRS2_LM"
DEopt	NMOF	nG=100, nP=20*dim
DEoptim	DEoptim	nP=20*dim
ga	GA	maxiter=50, monitor=FALSE, popSize=20*dim
genoud	rgenoud	print.level=0, max.generations=70, solution.tolerance=1e-12, pop.size=20*dim, wait.generations=dim, boundary.enforcement=2, gradient.check=FALSE, MemoryMatrix=FALSE
GenSA	GenSA	max.call=1e7, max.time=25/dim, trace.mat=FALSE
isres	nloptr	maxeval=2e4, xtol_rel=1e-14, population=20*dim, algorithm="NLOPT_GN_ISRES"
malschains	Rmalschains	popsize=20*dim, maxEvals=25000

nlminbOpt	MSCMT/stats	nrandom=30
optimOpt	MSCMT/stats	nrandom=25
PSopt	NMOF	nG=100, nP=20*dim
psoptim	pso	maxit=700
soma	soma	nMigrations=100

If `outer.opar` is a list of lists, its names must correspond to (a subset of) the outer optimizers chosen in `outer.optim`.

`std.v` A character scalar containing one of the function names "sum", "mean", "min", or "max" for the standardization of the predictor weights (weights are divided by `std.v(weights)` before reporting). Defaults to "sum", partial matching allowed.

`alpha` A numerical vector with weights for the dependent variables in an MSCMT optimization or `NULL` (default). If not `NULL`, the length of `alpha` must agree with the number of dependent variables, `NULL` is equivalent to weight 1 for all dependent variables.

`beta` Either `NULL` (default), a numerical vector, or a list. If `beta` is a numerical vector or a list, its length must agree with the number of dependent variables.

- If `beta` is a numerical vector, the `i`th dependent variable is discounted with discount factor `beta[i]` (the observations of the dependent variables must thus be in chronological order!).
- If `beta` is a list, the components of `beta` must be numerical vectors with lengths corresponding to the numbers of observations for the individual dependent variables. These observations are then multiplied with the corresponding component of `beta`.

`gamma` Either `NULL` (default), a numerical vector, or a list. If `gamma` is a numerical vector or a list, its length must agree with the number of predictor variables.

- If `gamma` is a numerical vector, the output of `agg.fns[i]` applied to the `i`th predictor variable is discounted with discount factor `gamma[i]` (the output of `agg.fns[i]` must therefore be in chronological order!).
- If `gamma` is a list, the components of `gamma` must be numerical vectors with lengths corresponding to the lengths of the output of `agg.fns` for the individual predictor variables. The output of `agg.fns` is then multiplied with the corresponding component of `gamma`.

`return.ts` A logical scalar. If `TRUE` (default), most results are converted to time series.

`single.v` A logical scalar. If `FALSE` (default), a selection of feasible (optimal!) predictor weight vectors is generated. If `TRUE`, the one optimal weight vector which has maximal order statistics is generated to facilitate cross validation studies.

`verbose` A logical scalar. If `TRUE` (default), output is verbose.

`debug` A logical scalar. If `TRUE`, output is very verbose. Defaults to `FALSE`.

`seed` A numerical vector or `NULL`. If not `NULL`, the random number generator is initialized with the elements of `seed` via `set.seed(seed)` (see Random) before calling the optimizer, performing repeated optimizations (and staying with the best) if `seed` has length greater than 1. Defaults to `NULL`. If not `NULL`, the seeds `int.seed` (default: 53058) and `unif.seed` (default: 812821) for genoud are

also initialized to the corresponding element of seed, but this can be overridden with the list elements int.seed and unif.seed of (the corresponding element of) outer.opar.

cl NULL (default) or an object of class cluster obtained by makeCluster of package parallel. Repeated estimations (see outer.optim and seed) and placebo studies will make use of the cluster cl (if not NULL).

times.pred.training
A matrix with two rows (containing start times in the first and end times in the second row) and one column for each predictor variable, where the column names must exactly match the names of the corresponding predictor variables (or NULL by default). If not NULL, times.pred.training defines training periods for cross-validation applications. For the format of the start and end times, see the documentation of parameter times.pred.

times.dep.validation
A matrix with two rows (containing start times in the first and end times in the second row) and one column for each dependent variable, where the column names must exactly match the names of the corresponding dependent variables (or NULL by default). If not NULL, times.dep.validation defines validation period(s) for cross-validation applications. For the format of the start and end times, see the documentation of parameter times.dep.

v.special integer vector containing indices of important predictors with special treatment (see below). Defaults to the empty set.

cv.alpha numeric scalar containing the minimal proportion (of the maximal feasible weight) for the weights of the predictors selected by v.special. Defaults to 0.

spec.search.treated
A logical scalar. If TRUE, a specification search (for the optimal set of included predictors) is done for the treated unit. Defaults to FALSE.

spec.search.placebos
A logical scalar. If TRUE, a specification search (for the optimal set of included predictors) is done for the control unit. Defaults to FALSE.

Details

mscmt combines, if necessary, the preparation of the raw data (which is expected to be in "list" format, possibly after conversion from a data.frame with function listFromLong) and the call to the appropriate MSCMT optimization procedures (depending on the input parameters). For details on the input parameters alpha, beta, and gamma, see [1]. For details on cross-validation, see [2].

Value

An object of class "mscmt", which is essentially a list containing the results of the estimation and, if applicable, the placebo study. The most important list elements are

- the weight vector w for the control units,
- a matrix v with weight vectors for the predictors in its columns,
- scalars loss.v and rmspe with the dependent loss and its square root,

- a vector `loss.w` with the predictor losses corresponding to the various weight vectors in the columns of `v`,
- a matrix `predictor.table` containing aggregated statistics of predictor values (similar to list element `tab.pred` of function `synth.tab` of package `'Synth'`),
- a list of multivariate time series `combined` containing, for each dependent and predictor variable, a multivariate time series with elements `treated` for the actual values of the treated unit, `synth` for the synthesized values, and `gaps` for the differences.

Placebo studies produce a list containing individual results for each unit (as treated unit), starting with the original treated unit, as well as a list element named `placebo` with aggregated results for each dependent and predictor variable.

If `times.pred.training` and `times.dep.validation` are not `NULL`, a cross-validation is done and a list of elements `cv` with the results of the cross-validation period and `main` with the results of the main period is returned.

References

[1] Becker M, Klößner S (2018). "Fast and Reliable Computation of Generalized Synthetic Controls." *Econometrics and Statistics*, **5**, 1–19. https://doi.org/10.1016/j.ecosta.2017.08.002.

[2] Becker M, Klößner S, Pfeifer G (2018). "Cross-Validating Synthetic Controls." *Economics Bulletin*, **38**, 603-609. Working Paper, http://www.accessecon.com/Pubs/EB/2018/Volume38/EB-18-V38-I1-P58.pdf.

Examples

```
## Not run:
## for examples, see the package vignettes:
browseVignettes(package="MSCMT")

## End(Not run)
```

`plot.mscmt` *Plotting Results of MSCMT*

Description

`plot.mscmt` plots results of `mscmt`.

Usage

```
## S3 method for class 'mscmt'
plot(
  x,
  what,
  type = c("gaps", "comparison", "placebo.gaps", "placebo.data"),
  treatment.time,
```

```
  zero.line = TRUE,
  ylab,
  xlab = "Date",
  main,
  sub,
  col,
  lty,
  lwd,
  legend = TRUE,
  bw = FALSE,
  ...
)
```

Arguments

x	An object of class "mscmt", usually obtained as the result of a call to function mscmt.
what	A character scalar. Name of the variable to be plotted. If missing, the (first) dependent variable will be used.
type	A character scalar denoting the type of the plot containing either "gaps", "comparison", "placebo.gaps", or "placebo.data". Partial matching allowed, defaults to "placebo.gaps", if results of a placebo study are present, and to "gaps", else.
treatment.time	An optional numerical scalar. If not missing, a vertical dotted line at the given point in time is included in the plot. treatment.time is measured in years, but may as well be a decimal number to reflect treatment times different from January 1st.
zero.line	A logical scalar. If TRUE (default), a horizontal dotted line (at zero level) is plotted for "gaps" and "placebo" plots.
ylab	Optional label for the y-axis, automatically generated if missing.
xlab	Optional label for the x-axis, defaults to "Date".
main	Optional main title for the plot, automatically generated if missing.
sub	Optional subtitle for the plot. If missing, the subtitle is generated automatically for "comparison" and "gaps" plots.
col	Optional character vector with length corresponding to the number of units. Contains the colours for the different units, automatically generated if missing.
lty	Optional numerical vector with length corresponding to the number of units. Contains the line types for the different units, automatically generated if missing.
lwd	Optional numerical vector with length corresponding to the number of units. Contains the line widths for the different units, automatically generated if missing.
legend	A logical scalar. If TRUE (default), a legend is included in the plot.
bw	A logical scalar. If FALSE (default), the automatically generated colours and line types are optimized for a colour plot, if TRUE, the automatic colours and line types are set for a black and white plot.
...	Further optional parameters for the underlying plot function.

Details

A unified basic plot function for gaps plots, comparison of treated and synthetic values, as well as plots for placebo studies. Consider using `ggplot.mscmt` instead, which is the preferred plot method and has more functionality than `plot.mscmt`.

Value

Nothing useful (function is called for its side effects).

`ppratio` *Post-pre-(r)mspe-ratios for placebo studies*

Description

`ppratio` calculates post-to-pre-(r)mspe-ratios for placebo studies.

Usage

```
ppratio(
  x,
  what,
  range.pre,
  range.post,
  type = c("rmspe", "mspe"),
  return.all = FALSE
)
```

Arguments

`x`	An object of class `"mscmt"`, usually obtained as the result of a call to function `mscmt`.
`what`	A character vector. Name of the variable to be considered. If missing, the (first) dependent variable will be used.
`range.pre`	A vector of length 2 defining the range of the pre-treatment period with start and end time given as • annual dates, if the format of start/end time is "dddd", e.g. "2016", • quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1", • monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10", • weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23", • daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18", corresponding to the format of the respective column of the `times.dep` argument of `mscmt`. If missing, the corresponding column of `times.dep` will be used.

`range.post` A vector of length 2 defining the range of the post-treatment period with start and end time given as

- annual dates, if the format of start/end time is "dddd", e.g. "2016",
- quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1",
- monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10",
- weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23",
- daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

corresponding to the format of the respective column of the `times.dep` argument of `mscmt`. Will be guessed if missing.

`type` A character string. Either `rmspe` (default) or `mspe`. Selects whether root mean squared errors or mean squared errors are calculated.

`return.all` A logical scalar. If `FALSE` (default), only the (named) vector of post-pre-(r)mspe-ratios is returned, if `TRUE`, a three-column matrix with pre- and post-treatment (r)mspe's as well as the post-pre-ratios will be returned.

Details

`ppratio` calculates post-to-pre-(r)mspe-ratios for placebo studies based on Synthetic Control Methods.

Value

If `return.all` is `FALSE`, a (named) vector of post-pre-(r)mspe-ratios. If `return.all` is `TRUE`, a matrix with three columns containing the pre-treatment (r)mspe, the post-treatment (r)mspe, and the post-pre-ratio.

`print.mscmt` *Printing Results of MSCMT*

Description

`print.mscmt` prints results of `mscmt`.

Usage

```
## S3 method for class 'mscmt'
print(x, ...)
```

Arguments

`x` An object of class `"mscmt"`, usually obtained as the result of a call to function `mscmt`.

`...` Further arguments to be passed to or from other methods. They are ignored in this function.

Details

A human-readable summary of mscmt's results.

Value

Nothing useful (function is called for its side effects).

---

pvalue *P-values for placebo studies*

Description

pvalue calculates p-values for placebo studies.

Usage

```
pvalue(
  x,
  what,
  range.pre,
  range.post,
  alternative = c("two.sided", "less", "greater"),
  exclude.ratio = Inf,
  ratio.type = c("rmspe", "mspe")
)
```

Arguments

x	An object of class "mscmt", usually obtained as the result of a call to function mscmt.
what	A character vector. Name of the variable to be considered. If missing, the (first) dependent variable will be used.
range.pre	A vector of length 2 defining the range of the pre-treatment period with start and end time given as • annual dates, if the format of start/end time is "dddd", e.g. "2016", • quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1", • monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10", • weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23", • daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18", corresponding to the format of the respective column of the times.dep argument of mscmt. If missing, the corresponding column of times.dep will be used.

`range.post` A vector of length 2 defining the range of the post-treatment period with start and end time given as

- annual dates, if the format of start/end time is "dddd", e.g. "2016",
- quarterly dates, if the format of start/end time is "ddddQd", e.g. "2016Q1",
- monthly dates, if the format of start/end time is "dddd?dd" with "?" different from "W" (see below), e.g. "2016/03" or "2016-10",
- weekly dates, if the format of start/end time is "ddddWdd", e.g. "2016W23",
- daily dates, if the format of start/end time is "dddd-dd-dd", e.g. "2016-08-18",

corresponding to the format of the respective column of the `times.dep` argument of `mscmt`. Will be guessed if missing.

`alternative` A character string giving the alternative of the test. Either `"two.sided"` (default), `"less"`, or `"greater"`.

`exclude.ratio` A numerical scalar (default: `Inf`). Control units with a pre-treatment-(r)mspe of more than `exclude.ratio` times the pre-treatment-(r)mspe of the treated unit are excluded from the calculations of the p-value.

`ratio.type` A character string. Either `rmspe` (default) or `mspe`. Selects whether root mean squared errors or mean squared errors are calculated.

Details

`pvalue` calculates p-values for placebo studies based on Synthetic Control Methods.

Value

A time series containing the p-values for the post-treatment periods.

Examples

```
## Not run:
## for an example, see the main package vignette:
 vignette("WorkingWithMSCMT",package="MSCMT")

## End(Not run)
```

Index

as.Date, *7*

compare, 2

data.frame, *11, 12*, *16*
dataprep, *11, 12*
Date, *6*
did, 3

genoud, *15*
ggplot, *4*, *8*
ggplot.mscmt, *3*, 4, *8*, *19*

improveSynth, 8
ipop, *14*

listFromLong, 10, *12*, *16*
LowRankQP, *14*

makeCluster, *16*
mscmt, *2–5*, *9*, 11, *18–22*

plot, *18*
plot.mscmt, *8*, 17, *19*
ppratio, 19
print.mscmt, *3*, 20
pvalue, 21

Random, *15*

strptime, *6*
synth, *8*
synth.tab, *17*