Package 'scpi'

February 11, 2022

Title Prediction Intervals for Synthetic Control Methods

Version 0.1

Author Matias Cattaneo, Yingjie Feng, Filippo Palomba, and Rocío Titiunik

Maintainer Filippo Palomba <fpalomba@princeton.edu>

Description The package computes point estimates and prediction intervals for Synthetic Control methods as proposed in Cattaneo, Feng, and Titiunik (2021).

Depends R (>= 4.1.0)

License GPL-2

Encoding UTF-8

Imports abind   (>= 1.4.5),
     CVXR   (>= 1.0.10),
     doRNG   (>= 1.8.2),
     doSNOW   (>= 1.0.19),
     dplyr   (>= 1.0.7),
     ECOSolveR   (>= 0.5.4),
     fastDummies (>= 1.6.3),
     foreach   (>= 1.5.1),
     ggplot2   (>= 3.3.3),
     magrittr   (>= 2.0.1),
     Matrix   (>= 1.3.3),
     nloptr   (>= 1.2.2.2),
     parallel   (>= 4.1.0),
     purrr   (>= 0.3.4),
     Qtools   (>= 1.5.4),
     rlang   (>= 0.4.11),
     stats   (>= 4.1.0),
     stringr   (>= 1.4.0),
     tibble   (>= 3.1.2),
     tidyr   (>= 1.1.3),
     utils   (>= 4.1.1)

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

---

scpi-package   *scpi: A package to compute Synthetic Control Prediction Intervals*

Description

The package computes point estimates and prediction intervals for Synthetic Control methods.

Author(s)

- Matias Cattaneo, Princeton University

- Yingjie Feng, Tsinghua University

- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.

- Rocio Titiunik, Princeton University

References

- Cattaneo, M. D., Feng, Y., & Titiunik, R. (2021). Prediction intervals for synthetic control methods. *Journal of the American Statistical Association*, 116(536), 1865-1880.

- Cattaneo, M. D., Feng, Y., Palomba F., and Titiunik, R. (2022)

---

print.scest   *Print Method for Synthetic Control Estimation*

Description

The print method for synthetic control estimation objects.

Usage

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

Arguments

x	Class "scest" object, obtained by calling [scest]().
...	Other arguments.

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

See Also

[scest]() for synthetic control estimation.

Supported methods: [print.scest](), [summary.scest]().

---

print.scpi	*Print Method for Synthetic Control Inference*

Description

The print method for synthetic control inference objects.

Usage

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

Arguments

x	Class "scpi" object, obtained by calling [scpi]().
...	Other arguments.

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

See Also

[scpi]() for synthetic control inference

Supported methods: [print.scpi](), [summary.scpi]().

---

scdata	*Data preparation to use before calling* scest *or* scpi *for point estimation and inference procedures using Synthetic Control.*

Description

The command prepares the data to be used by [scest](#) or [scpi](#). It allows the user to specify the outcome variable, the features of the treated unit to be matched, and covariate-adjustment feature by feature. The names of the output matrices follow the notation proposed in Cattaneo, Feng, and Titiunik (2021).

Usage

```
scdata(
  df,
  id.var,
  time.var,
  outcome.var,
  period.pre,
  period.post,
  unit.tr,
  unit.co,
  features = NULL,
  cov.adj = NULL,
  cointegrated.data = FALSE,
  anticipation = 0,
  constant = FALSE,
  report.missing = FALSE
)
```

Arguments

df	a dataframe object.
id.var	a character or numeric scalar with the name of the variable containing units' IDs. The ID variable can be numeric or character.
time.var	a character with the name of the time variable. The time variable has to be numeric, integer, or Date. In case time.var is Date it should be the output of [as.Date](#)() function. An integer or numeric time variable is suggested when working with yearly data, whereas for all other formats a Date type time variable is preferred.
outcome.var	a character with the name of the outcome variable. The outcome variable has to be numeric.
period.pre	a numeric vector that identifies the pre-treatment period in time.var.
period.post	a numeric vector that identifies the post-treatment period in time.var.
unit.tr	a character or numeric scalar that identifies the treated unit in id.var.
unit.co	a character or numeric vector that identifies the donor pool in id.var.
features	a character vector containing the name of the feature variables used for estimation. If features is specified, then outcome.var must be included in it. If this option is not specified the default is features = outcome.var.

cov.adj	a list specifying the names of the covariates to be used for adjustment for each feature. See the **Details** section for more.
cointegrated.data	
	a logical that indicates if there is a belief that the data is cointegrated or not. The default value is FALSE. See the **Details** section for more.
anticipation	a scalar that indicates the number of periods of potential anticipation effects. Default is 0.
constant	a logical which controls the inclusion of a constant term across features. The default value is FALSE.
report.missing	a logical which prints the location of missing values if present. The default value is FALSE.

Details

- cov.adj can be used in two ways. First, if only one feature is specified through the option features, cov.adj has to be a list with one (even unnamed) element (eg. cov.adj = list(c("constant","trend"))). Alternatively, if multiple features are specified, then the user has two possibilities:
  - provide a list with one element, then the same covariates are used for adjustment for each feature. For example, if there are two features specified and the user inputs cov.adj = list(c("constant","trend")), then a constant term and a linear trend are for adjustment for both features.
  - provide a list with as many elements as the number of features specified, then feature-specific covariate adjustment is implemented. For example, cov.adj = list('f1' = c("constant","trend"),'f2' = c("trend")). In this case the name of each element of the list should be one (and only one) of the features specified.

  This option allows the user to include feature-specific constant terms

- cointegrated.data allows the user to model the belief that $\mathbf{A}$ and $\mathbf{B}$ form a cointegrated system. In practice, this implies that when dealing with the pseudo-true residuals $\mathbf{u}$, the first-difference of $\mathbf{B}$ are used rather than the levels.

Value

A	a matrix containing pre-treatment features of the treated unit.
B	a matrix containing pre-treatment features of the control units.
C	a matrix containing covariates for adjustment.
P	a matrix whose rows are the vectors used to predict the out-of-sample series for the synthetic unit.
Y.pre	a matrix containing the pre-treatment outcome of the treated unit.
Y.post	a matrix containing the post-treatment outcome of the treated unit.
Y.donors	a matrix containing the pre-treatment outcome of the control units.
specs	a list containing some specifics of the data:

- J, the number of control units
- K, a numeric vector with the number of covariates used for adjustment for each feature
- KM, the total number of covariates used for adjustment
- M, number of features
- period.pre, a numeric vector with the pre-treatment period

- period.post, a numeric vector with the post-treatment period
- T0.features, a numeric vector with the number of periods used in estimation for each feature
- T1.outcome, the number of post-treatment periods
- glob.cons, for internal use only
- out.in.features, for internal use only

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). `<fpalomba@princeton.edu>`.
- Rocio Titiunik, Princeton University

References

- Abadie, A. (2021). Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2), 391-425.
- Cattaneo, M. D., Feng, Y., & Titiunik, R. (2021). Prediction intervals for synthetic control methods. *Journal of the American Statistical Association*, 116(536), 1865-1880.
- Cattaneo, M. D., Feng, Y., Palomba F., and Titiunik, R. (2022).. scpi - Uncertainty Quantification for Synthetic Control Estimators.

See Also

scest, scpi, scplot

Examples

```
data <- scpi_germany

df <- scdata(df = data, id.var = "country", time.var = "year",
             outcome.var = "gdp", period.pre = (1960:1990),
             period.post = (1991:2013), unit.tr = "West Germany",
             unit.co = unique(data$country)[-7], constant = T,
             cointegrated.data = T)
```

---

scest   *Estimation of Synthetic Control*

Description

The command constructs a counterfactual synthetic control unit as proposed in Cattaneo, Feng, and Titiunik (2021).

Usage

```
scest(
  data,
  w.constr = NULL,
  V = NULL,
  opt.list = NULL,
  plot = FALSE,
  plot.name = NULL,
  plot.path = NULL,
  save.data = NULL
)
```

Arguments

data   a class 'scpi_data' object, obtained by calling scdata.

w.constr   a list specifying the constraint set the estimated weights of the donors must
                belong to. w.constr can contain up to four objects:

- 'p', a string indicating the norm to be constrained (p should be one of "no norm", "L1", and "L2")
- 'dir', a string indicating whether the constraint on the norm is an equality ("==") or inequality ("<=")
- 'Q', a scalar defining the value of the constraint on the norm
- 'lb', a scalar defining the lower bound on the weights. It can be either 0 or -Inf.
- 'name', a character selecting one of the default proposals. See the **Details** section for more.

V   a weighting matrix to be used when minimizing the sum of squared residuals

$$(\mathbf{A} - \mathbf{Bw} - \mathbf{Cr})'\mathbf{V}(\mathbf{A} - \mathbf{Bw} - \mathbf{Cr})$$

The default is the identity matrix, so equal weight is given to all observations.

opt.list   a list specifying the stopping criteria and the algorithm for the underlying opti-
                mizer nloptr for point estimation. See the **Details** section for more.

plot   a logical specifying whether scplot should be called and a plot saved in the
                current working directory. For more options see scplot.

plot.name   a string containing the name of the plot (the format is by default .png). For more
                options see scplot.

plot.path   a string containing the path at which the plot should be saved (default is output
                of getwd().)

save.data   a character specifying the name and the path of the saved dataframe containing
                the processed data used to produce the plot.

Details

- **Estimation of Weights.** w.constr specifies the constraint set on the weights. First, the element p allows the user to choose between imposing a constraint on either the L1 (p = "L1") or the L2 (p = "L2") norm of the weights and imposing no constraint on the norm (p = "no norm"). Second, Q specifies the value of the constraint on the norm of the weights. Third,

lb sets the lower bound of each component of the vector of weights. Fourth, `dir` sets the direction of the constraint on the norm in case `p = "L1"` or `p = "L2"`. If `dir = "=="`, then

$$||\mathbf{w}||_p = Q, \quad w_j \geq lb, \ j = 1, \ldots, J$$

If instead `dir = "<="`, then

$$||\mathbf{w}||_p \leq Q, \quad w_j \geq lb, \ j = 1, \ldots, J$$

If instead `dir = "NULL"` no constraint on the norm of the weights is imposed.

An alternative to specifying an ad-hoc constraint set on the weights would be choosing among some popular types of constraints. This can be done by including the element 'name' in the list `w.constr`. The following are available options:

– If `name == "simplex"` (the default), then

$$||\mathbf{w}||_1 = 1, \quad w_j \geq 0, \ j = 1, \ldots, J.$$

– If `name == "lasso"`, then
$$||\mathbf{w}||_1 \leq Q,$$

where `Q` is by default equal to 1 but it can be provided as an element of the list (eg. `w.constr = list(name = "lasso",Q = 2)`).

– If `name == "ridge"`, then
$$||\mathbf{w}||_2 \leq Q,$$

where `Q` is a tuning parameter that is by default computed as

$$(J + KM)\widehat{\sigma}_u^2 / ||\widehat{\mathbf{w}}_{OLS}||_2^2$$

where J is the number of donors and KM is the total number of covariates used for adjustment. The user can provide `Q` as an element of the list (eg. `w.constr = list(name = "ridge",Q = 1)`).

– If `name == "ols"`, then the problem is unconstrained and the vector of weights is estimated via ordinary least squares.

- **Algorithm Options.** The default is a sequential quadratic programming (SQP) algorithm for nonlinearly constrained gradient-based optimization (`algorithm = 'NLOPTR_LD_SLSQP'`) for all cases not involving the L1 norm. For a complete list of algorithms see the official nloptr documentation. The other default values are `maxeval = 5000`, `ftol_res = 1.0e-8`, `ftol_abs = 1.0e-8`, `xtol_res = 1.0e-8`, `xtol_abs = 1.0e-8`, `tol_constraints_eq = 1.0e-8`, and `tol_constraints_ine = 1.0e-8`. More information on the stopping criteria can be obtained running `nloptr.print.options()` or `nloptr.get.default.options()`. f the optimization involves the L1 norm then CVXR is used for optimization. More information on the stopping criteria can be obtained reading the official documentation.

Value

data	object containing used data and some other values.
est.results	object containing all the estimation results.

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

References

- Abadie, A. (2021). Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2), 391-425.

- Cattaneo, M. D., Feng, Y., & Titiunik, R. (2021). Prediction intervals for synthetic control methods. *Journal of the American Statistical Association*, 116(536), 1865-1880.

- Cattaneo, M. D., Feng, Y., Palomba F., and Titiunik, R. (2022).. scpi - Uncertainty Quantification for Synthetic Control Estimators.

See Also

scdata, scpi, scplot

Examples

```
data <- scpi_germany

df <- scdata(df = data, id.var = "country", time.var = "year",
             outcome.var = "gdp", period.pre = (1960:1990),
             period.post = (1991:2013), unit.tr = "West Germany",
             unit.co = unique(data$country)[-7], constant = T,
             cointegrated.data = T)

result <- scest(df, w.constr = list(name = "simplex", Q = 1))
result <- scest(df, w.constr = list(lb = 0, dir = "==", p = "L1", Q = 1))
```

---

scpi	*Prediction Intervals for Synthetic Control Methods*

Description

The command constructs a counterfactual synthetic control unit as proposed in Cattaneo, Feng, and Titiunik (2021). scpi returns the estimated post-treatment series for the synthetic unit through the command scest and quantifies in-sample and out-of-sample uncertainty to provide confidence intervals for each point estimate.

Usage

```
scpi(
  data,
  w.constr = NULL,
  V = NULL,
  P = NULL,
  u.missp = TRUE,
  u.sigma = "HC1",
  u.order = 1,
  u.lags = 0,
  u.design = NULL,
  u.alpha = 0.05,
  e.method = "all",
```

```
    e.order = 1,
    e.lags = 0,
    e.design = NULL,
    e.alpha = 0.05,
    sims = 200,
    rho = NULL,
    rho.max = NULL,
    cores = NULL,
    plot = FALSE,
    plot.name = NULL,
    w.bounds = NULL,
    e.bounds = NULL,
    opt.list.est = NULL,
    opt.list.inf = NULL,
    save.data = NULL
)
```

Arguments

data	a class 'scpi_data' object, obtained by calling [scdata](#).
w.constr	a list specifying the constraint set the estimated weights of the donors must belong to. w.constr can contain up to five elements:

- 'p', a scalar indicating the norm to be used (p should be one of "no norm", "L1", and "L2")
- 'dir', a string indicating whether the constraint on the norm is an equality ("==") or inequality ("<=")
- 'Q', a scalar defining the value of the constraint on the norm
- 'lb', a scalar defining the lower bound on the weights. It can be either 0 or -Inf.
- 'name', a character selecting one of the default proposals See the **Details** section for more.

V	a weighting matrix to be used when minimizing

$$(\mathbf{A} - \mathbf{Bw} - \mathbf{Cr})'\mathbf{V}(\mathbf{A} - \mathbf{Bw} - \mathbf{Cr})$$

P	a $T_1 \times (J + K_1)$ matrix containing the design matrix to be used to obtain the predicted. post-intervention outcome of the synthetic control unit. T_1 is the number of post-treatment periods, J is the size of the donor pool, and K_1 is the number of covariates used for adjustment in the outcome equation.
u.missp	a logical indicating if misspecification should be taken into account when dealing with $\mathbf{u}$.
u.sigma	a string specifying the type of variance-covariance estimator to be used when estimating the conditional variance of $\mathbf{u}$.
u.order	a scalar that sets the order of the polynomial in $\mathbf{B}$ when predicting moments of $\mathbf{u}$.
u.lags	a scalar that sets the number of lags of $\mathbf{B}$ when predicting moments of $\mathbf{u}$.
u.design	a matrix with the same number of rows of $\mathbf{A}$ and $\mathbf{B}$ and whose columns specify the design matrix to be used when modeling the estimated pseudo-true residuals $\mathbf{u}$.
u.alpha	a scalar specifying the confidence level for in-sample uncertainty.

e.method	a string selecting the method to be used in quantifying out-of-sample uncertainty among: "gaussian" which uses conditional subgaussian bounds; "ls" which specifies a location-scale model for $\mathbf{u}$; "qreg" which employs a quantile regressions to get the conditional bounds; "all" uses each one of the previous methods.
e.order	a scalar that sets the order of the polynomial in $\mathbf{B}$ when predicting moments of $\mathbf{e}$.
e.lags	a scalar that sets the number of lags of $\mathbf{B}$ when predicting moments of $\mathbf{e}$.
e.design	a matrix with the same number of rows of $\mathbf{A}$ and $\mathbf{B}$ and whose columns specify the design matrix to be used when modeling the estimated out-of-sample residuals $\mathbf{e}$.
e.alpha	a scalar specifying the confidence level for out-of-sample uncertainty.
sims	a scalar providing the number of simulations to be used in quantifying in-sample uncertainty.
rho	a string specifying the regularizing parameter that imposes sparsity on the estimated vector of weights. If rho = 'type-1' (the default), then the tuning parameter is computed based on optimization inequalities. Users can provide a scalar with their own value for rho. Other options are described in the **Details** section.
rho.max	a scalar indicating the maximum value attainable by the tuning parameter rho.
cores	number of cores to be used by the command. The default is the number of cores available minus one.
plot	a logical specifying whether scplot should be called and a plot saved in the current working directory. For more options see scplot.
plot.name	a string containing the name of the plot (the format is by default .png). For more options see scplot.
w.bounds	a $T_1 \times 2$ matrix with the user-provided bounds on β. If w.bounds is provided, then the quantification of in-sample uncertainty is skipped. It is possible to provide only the lower bound or the upper bound by filling the other column with NAs.
e.bounds	a $T_1 \times 2$ matrix with the user-provided bounds on $\mathbf{e}$. If e.bounds is provided, then the quantification of out-of-sample uncertainty is skipped. It is possible to provide only the lower bound or the upper bound by filling the other column with NAs.
opt.list.est	a list specifying the stopping criteria and the algorithm for the underlying optimizer nloptr for point estimation. See the **Details** section for more.
opt.list.inf	similar to the previous one but for the optimizer used for inference purposes. See the **Details** section for more.
save.data	a character specifying the name and the path of the saved dataframe containing the processed data used to produce the plot.

Details

- **Estimation of Weights.** w.constr specifies the constraint set on the weights. First, the element p allows the user to choose between imposing a constraint on either the L1 (p = "L1") or the L2 (p = "L2") norm of the weights and imposing no constraint on the norm (p = "no norm"). Second, Q specifies the value of the constraint on the norm of the weights. Third, lb sets the lower bound of each component of the vector of weights. Fourth, dir sets the direction of the constraint on the norm in case p = "L1" or p = "L2". If dir = "==", then

$$||\mathbf{w}||_p = Q, \quad w_j \geq lb, \ j = 1, \dots, J$$

If instead dir = ”<=”, then

$$||\mathbf{w}||_p \leq Q, \quad w_j \geq lb, \quad j = 1, \ldots, J$$

If instead dir = ”NULL” no constraint on the norm of the weights is imposed.

An alternative to specifying an ad-hoc constraint set on the weights would be choosing among some popular types of constraints. This can be done by including the element 'name' in the list w.constr. The following are available options:

  – If name == ”simplex” (the default), then

$$||\mathbf{w}||_1 = 1, \quad w_j \geq 0, \quad j = 1, \ldots, J.$$

  – If name == ”lasso”, then
$$||\mathbf{w}||_1 \leq Q,$$

  where Q is by default equal to 1 but it can be provided as an element of the list (eg. w.constr = list(name = ”lasso”, Q = 2)).

  – If name == ”ridge”, then
$$||\mathbf{w}||_2 \leq Q,$$

  where Q is a tuning parameter that is by default computed as

$$(J + KM)\widehat{\sigma}_u^2 / ||\widehat{\mathbf{w}}_{OLS}||_2^2$$

  where J is the number of donors and KM is the total number of covariates used for adjustment. The user can provide Q as an element of the list (eg. w.constr = list(name = ”ridge”, Q = 1)).

  – If name == ”ols”, then the problem is unconstrained and the vector of weights is estimated via ordinary least squares.

• **Regularization.** rho is estimated through the formula

$$\varrho = \mathcal{C}\frac{\log(T_0)^c}{T_0^{1/2}}$$

where $\mathcal{C} = \widehat{\sigma}_u / \min_j \widehat{\sigma}_{b_j}$ if rho = 'type-1', $\mathcal{C} = \max_j \widehat{\sigma}_{b_j}\widehat{\sigma}_u / \min_j \widehat{\sigma}_{b_j}^2$ if rho = 'type-2', and $\mathcal{C} = \max_j \widehat{\sigma}_{b_j u} / \min_j \widehat{\sigma}_{b_j}^2$ if rho = 'type-3',
rho defines a new sparse weight vector as

$$\widehat{w}_j^\star = \mathbf{1}(\widehat{w}_j \geq \varrho)$$

• **In-sample uncertainty.** To quantify in-sample uncertainty it is necessary to model the pseudo-residuals $\mathbf{u}$. First of all, estimation of the first moment of $\mathbf{u}$ can be controlled through the option u.missp. When u.missp = FALSE, then $\mathbf{E}[u \,|\, H] = 0$. If instead u.missp = TRUE, then $\mathbf{E}[u \,|\, H]$ is estimated using a linear regression of $\widehat{\mathbf{u}}$ on H. The default set of variables in H is composed of $\mathbf{B}$ and $\mathbf{C}$ and, if required, with lags (u.lags) and polynomials (u.order) of $\mathbf{B}$. The option u.design allows the user to provide an ad-hoc set of variables to form H. Regarding the second moment of $\mathbf{u}$, different estimators can be chosen: HC0, HC1, HC2, HC3, and HC4 using the option u.sigma.

• **Out-of-sample uncertainty.** To quantify out-of-sample uncertainty it is necessary to model the out-of-sample residuals $\mathbf{e}$ and estimate relevant moments. By default, the design matrix used during estimation H is composed of $\mathbf{B}$ and $\mathbf{C}$ and, if required, with lags (e.lags) and polynomials (e.order) of $\mathbf{B}$. The option e.design allows the user to provide an ad-hoc set of variables to form H. Finally, the option e.method allows the user to select one of three estimation methods: "gaussian" relies on conditional subgaussian bounds; "ls" estimates conditional bounds using a location-scale model; "qreg" uses conditional quantile regression of the residuals $\mathbf{e}$ on H.

- **Algorithm Options.** The default is a sequential quadratic programming (SQP) algorithm for nonlinearly constrained gradient-based optimization (algorithm = 'NLOPTR_LD_SLSQP') for all cases not involving the L1 norm. For a complete list of algorithms see the official nloptr documentation. The other default values are maxeval = 5000, ftol_res = 1.0e-8, ftol_abs = 1.0e-8, xtol_res = 1.0e-8, xtol_abs = 1.0e-8, tol_constraints_eq = 1.0e-8, and tol_constraints_ine = 1.0e-8. More information on the stopping criteria can be obtained running nloptr.print.options() or nloptr.get.default.options(). If the optimization involves the L1 norm then CVXR is used for optimization. More information on the stopping criteria can be obtained reading the official documentation.

Value

data	object containing used data as returned by scdata and some other values.
est.results	object containing all the results from scest.
inference.results	
	object containing all the inference-related results.

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

References

- Abadie, A. (2021). Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2), 391-425.
- Cattaneo, M. D., Feng, Y., & Titiunik, R. (2021). Prediction intervals for synthetic control methods. *Journal of the American Statistical Association*, 116(536), 1865-1880.
- Cattaneo, M. D., Feng, Y., Palomba F., and Titiunik, R. (2022).. scpi - Uncertainty Quantification for Synthetic Control Estimators.

See Also

scdata, scest, scplot

Examples

```
data <- scpi_germany

df <- scdata(df = data, id.var = "country", time.var = "year",
             outcome.var = "gdp", period.pre = (1960:1990),
             period.post = (1991:2013), unit.tr = "West Germany",
             unit.co = unique(data$country)[-7], constant = T,
             cointegrated.data = T)

result <- scpi(df, w.constr = list(name = "simplex", Q = 1))
result <- scpi(df, w.constr = list(lb = 0, dir = "==", p = "L1", Q = 1))
```

---

scpi_germany   *Economic indicators for OECD countries*

Description

A dataset containing some economic indicators of 17 OECD countries from 1960 to 2003.

Usage

```
scpi_germany
```

Format

A data frame with 748 rows and 11 variables:

index country index.

country name of the country.

year time index, in years.

gdp GDP per Capita (PPP, 2002 USD).

infrate annual percentage change in consumer prices (base year 1995).

trade trade openness measured as export plus imports as percentage of GDP.

schooling percentage of secondary school attained in the total population aged 25 and older.

industry industry share of value added.

Source

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/24714>

---

scplot   *Plot Synthetic Control Estimate*

Description

The command plots the actual pre-treatment and post-treatment series of the treated unit and the estimated counterfactual synthetic control unit with corresponding confidence intervals. Confidence intervals can take into account either in-sample uncertainty only or in-sample and out-of-sample uncertainty as developed in Cattaneo, Feng, and Titiunik (2021). The input object should come from the command scest or from the command scpi.

Usage

```
scplot(
  result,
  fig.path = NULL,
  fig.name = NULL,
  fig.format = "png",
  e.out = T,
  col.treated = "grey46",
  col.synth = "mediumblue",
  label.xy = NULL,
  plot.range = NULL,
  x.ticks = NULL,
  event.label = NULL,
  plot.specs = NULL,
  save.data = NULL
)
```

Arguments

`result`	a class scpi_est' object, obtained by calling \code{\link{scest}}, or a class scpi_pi' object, obtained by calling [scpi](#)
`fig.path`	a string indicating the path where the plot(s) should be saved.
`fig.name`	a string indicating the name of the plot(s). If multiple plots will be saved the command automatically generates a numeric suffix to avoid overwriting them.
`fig.format`	a string indicating the format in which the plot(s) should be saved.
`e.out`	a logical specifying whether out-of-sample uncertainty should be included in the plot(s).
`col.treated`	a string specifying the color for the treated unit series. Find the full list at http://sape.inf.usi.ch/quick-reference/ggplot2/colour.
`col.synth`	a string specifying the color for the synthetic unit series. Find the full list at http://sape.inf.usi.ch/quick-reference/ggplot2/colour.
`label.xy`	a character list with two elements indicating the name of the axes (eg. label.xy = list(x.lab = "Year", y.lab = "GDP growth (%)")).
`plot.range`	a numeric array indicating the time range of the plot(s).
`x.ticks`	a numeric list containing the location of the ticks on the x axis.
`event.label`	a list containing a character object ('lab') indicating the label of the event and a numeric object indicating the height of the label in the plot.
`plot.specs`	a list containing some specifics to be passed to ggsave (eg. img.width, img.height, dpi)
`save.data`	a character specifying the name and the path of the saved dataframe containing the processed data used to produce the plot.

Value

`plots`	a list containing standard ggplot object(s) that can be used for further customization.

Author(s)

- Matias Cattaneo, Princeton University

- Yingjie Feng, Tsinghua University

- Filippo Palomba, Princeton University (maintainer). `<fpalomba@princeton.edu>`.

- Rocio Titiunik, Princeton University

References

- Cattaneo, M. D., Feng, Y., & Titiunik, R. (2021). Prediction intervals for synthetic control methods. *Journal of the American Statistical Association*, 116(536), 1865-1880.

- Cattaneo, M. D., Feng, Y., Palomba F., and Titiunik, R. (2022).. scpi - Uncertainty Quantification for Synthetic Control Estimators.

See Also

scdata, scest, scpi

Examples

```
data <- scpi_germany

df <- scdata(df = data, id.var = "country", time.var = "year",
             outcome.var = "gdp", period.pre = (1960:1990),
             period.post = (1991:2013), unit.tr = "West Germany",
             unit.co = unique(data$country)[-7], constant = T,
             cointegrated.data = T)

result <- scest(df, w.constr = list(name = "simplex", Q = 1))

scplot(result)
```

---

summary.scest   *Summary Method for Synthetic Control Estimation*

Description

The summary method for synthetic control estimation objects.

Usage

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

Arguments

object	Class "scest" object, obtained by calling scest.
...	Additional arguments

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

See Also

scest

Supported methods: print.scest, summary.scest.

---

summary.scpi	*Summary Method for Synthetic Control Inference*

Description

The summary method for synthetic control inference objects.

Usage

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

Arguments

object	Class "scpi" object, obtained by calling scpi.
...	Additional arguments

Author(s)

- Matias Cattaneo, Princeton University
- Yingjie Feng, Tsinghua University
- Filippo Palomba, Princeton University (maintainer). <fpalomba@princeton.edu>.
- Rocio Titiunik, Princeton University

See Also

scpi

Supported methods: print.scpi, summary.scpi.

Index