Package 'SCtools'

October 12, 2022

Title Extensions for Synthetic Controls Analysis

Version 0.3.2.1

Date 2022-05-30

Description Extensions to the synthetic controls analyses
performed by the package 'Synth' as detailed in Abadie, Diamond, and Hain-
mueller (2011) <doi:10.18637/jss.v042.i13>. Includes generating
and plotting placebos, post/pre-
MSPE (Mean Squared Prediction Error) significance tests and plots,
and calculating average treatment effects for multiple treated units.
This package represents an implementation of those methods
suggested in
Abadie, Diamond,and Hainmueller (2010) <doi:10.1198/jasa.2009.ap08746> for
use in Synthetic Control Analysis.

BugReports https://github.com/bcastanho/SCtools/issues

Maintainer Bruno Castanho Silva <bcsilva@wiso.uni-koeln.de>

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5), future (>= 1.6.2)

Imports ggplot2, Synth, stringr, stats, cvTools, furrr, dplyr, purrr,
magrittr

Suggests testthat (>= 3.0.0), knitr, tidyr, rmarkdown

RoxygenNote 7.2.0

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Bruno Castanho Silva [aut, cre]
(<https://orcid.org/0000-0001-9363-4704>),
Michael DeWitt [aut] (<https://orcid.org/0000-0001-8940-1967>)

Repository CRAN

Date/Publication 2022-06-09 11:20:02 UTC

R topics documented:

SCtools-package   *SCTools: Tools for Synthetic Control Methods*

Description

A set of functions to extend the synthetic controls analyses performed by the package 'Synth'. Includes generating and plotting placebos, significance tests and plots, and calculating average treatment effects for multiple treated units.

Details

It has several goals:

- Allow easy generation of placebos
- Generate figures for inference on SCM outputs
- Extend the existing Synth package

Author(s)

Maintainer: Bruno Castanho Silva <bcsilva@wiso.uni-koeln.de> (ORCID)

Authors:

- Michael DeWitt <me.dewitt.jr@gmail.com> (ORCID)

See Also

Useful links:

- Report bugs at https://github.com/bcastanho/SCtools/issues

---

alcohol	*World Alcohol per Capita Consumption*

Description

This data set has been compiled from data from the World Health Organization (WHO) and the World Bank (WB). The primary purpose was to investigate the effects of policy changes in the Russian Federation enacted in 2003 around alcohol consumption. This is an excellent case study for SCM approaches to be used. You can read more about the policy changes at https://www.theguardian.com/world/2019/oct/01/russian-alcohol-consumption-down-40-since-2003-who

Usage

```
alcohol
```

Format

a data.frame with 5107 rows and 8 columns:

country_name The name of the country

year year

consumption Alcohol consumption per capita (liters/person); all types

country_code Three letter country code

labor_force_participation_rate Labor force participation rate, total (percent of total population ages 15+)

mobile_cellular_subscriptions Mobile cellular subscriptions (per 100 people)

inflation Inflation, consumer prices (annual percent)

manufacturing Manufacturing, value added (percent of GDP)

country_num The country number

Details

WHO data available at https://apps.who.int/gho/data/node.main.A1039?lang=en.

WB data available at https://data.worldbank.org/.

---

generate.placebos   *Function to generate placebo synthetic controls*

Description

Constructs a synthetic control unit for each unit in the donor pool of an implementation of the synthetic control method for a single treated unit. Used for placebo tests (see plot_placebos, mspe.test, mspe.plot) to assess the strength and significance of a causal inference based on the synthetic control method. On placebo tests, see Abadie and Gardeazabal (2003), and Abadie, Diamond, and Hainmueller (2010, 2011, 2014).

Usage

```
generate.placebos(
  dataprep.out,
  synth.out,
  Sigf.ipop = 5,
  strategy = "sequential"
)

generate_placebos(
  dataprep.out,
  synth.out,
  Sigf.ipop = 5,
  strategy = "sequential"
)
```

Arguments

dataprep.out	A data.prep object produced by the dataprep command
synth.out	A synth.out object produced by the synth command
Sigf.ipop	The Precision setting for the ipop optimization routine. Default of 5.
strategy	The processing method you wish to use "sequential", "multicore" or "multisession". Use "multicore" or "multisession" to parallelize operations and reduce computing time. Default is sequential. Since SCtools >= 0.3.2 "multiprocess" is deprecated.

Value

df Data frame with outcome data for each control unit and their respective synthetic control and for the original treated and its control

mspe.placs Mean squared prediction error for the pretreatment period for each placebo

t0 First time unit in time.optimize.ssr

t1 First time unit after the highest value in time.optimize.ssr

tr Unit number of the treated unit

names.and.numbers Dataframe with two columns showing all unit numbers and names from control units

n Number of control units

treated.name Unit name of the treated unit

loss.v Pretreatment MSPE of the treated unit's synthetic control

Examples

```
## Not run: ## Example with toy data from Synth
library(Synth)
# Load the simulated data
data(synth.data)

# Execute dataprep to produce the necessary matrices for synth
dataprep.out<-
  dataprep(
    foo = synth.data,
    predictors = c("X1"),
    predictors.op = "mean",
    dependent = "Y",
    unit.variable = "unit.num",
    time.variable = "year",
    special.predictors = list(
      list("Y", 1991, "mean")
    ),
    treatment.identifier = 7,
    controls.identifier = c(29, 2, 13, 17),
    time.predictors.prior = c(1984:1989),
    time.optimize.ssr = c(1984:1990),
    unit.names.variable = "name",
    time.plot = 1984:1996
)

# run the synth command to create the synthetic control
synth.out <- synth(dataprep.out, Sigf.ipop=2)

## run the generate.placebos command to reassign treatment status
## to each unit listed as control, one at a time, and generate their
## synthetic versions. Sigf.ipop = 2 for faster computing time.
## Increase to the default of 5 for better estimates.
tdf <- generate.placebos(dataprep.out,synth.out, Sigf.ipop = 2)

## End(Not run)
```

---

is_tdf   *Test if the object is a tdf object*

Description

This function returns 'TRUE' for the object returned from the `generate.placebos` function. and 'FALSE' for all other objects, including regular data frames.

Usage

```
is_tdf(x)
```

Arguments

x   An object

Value

'TRUE' if the object inherits from the 'tdf' class.

---

is_tdf_multi	*Test if the object is a tdf_multi object*

Description

This function returns 'TRUE' for the object returned from the `multiple.synth` function. and 'FALSE' for all other objects, including regular data frames.

Usage

```
is_tdf_multi(x)
```

Arguments

x   An object

Value

'TRUE' if the object inherits from the 'tdf_multi' class.

---

mspe.plot *Plot the post/pre-treatment MSPE ratio*

Description

Plots the post/pre-treatment mean square prediction error ratio for the treated unit and placebos.

Usage

```
mspe.plot(
  tdf,
  discard.extreme = FALSE,
  mspe.limit = 20,
  plot.hist = FALSE,
  title = NULL,
  xlab = "Post/Pre MSPE ratio",
  ylab = NULL
)

mspe_plot(
  tdf,
  discard.extreme = FALSE,
  mspe.limit = 20,
  plot.hist = FALSE,
  title = NULL,
  xlab = "Post/Pre MSPE ratio",
  ylab = NULL
)
```

Arguments

tdf	An object constructed by [generate.placebos](#).
discard.extreme	
	Logical. Whether or not placebos with high pre-treatement MSPE should be excluded from the plot.
mspe.limit	Numerical. Used if discard.extreme is TRUE. It indicates how many times the pretreatment MSPE of a placebo should be higher than that of the treated unit to be considered extreme and discarded. Default is 20.
plot.hist	Logical. If FALSE, a dotplot with each unit name and its post/pre treatment MSPE ratio is produced. If TRUE, a histogram is produced, with the frequency of each ratio. Should be set to TRUE when there are many controls, to make visualization easier.
title	Character. Optional. Title of the plot.
xlab	Character. Optional. Label of the x axis.
ylab	Character. Optional. Label of the y axis.

Details

Post/pre-treatement mean square prediction error ratio is the difference between the observed out-come of a unit and its synthetic control, before and after treatment. A higher ratio means a small pretreatment prediction error (a good synthetic control), and a high post-treatment MSPE, meaning a large difference between the unit and its synthetic control after the intervention. By calculating this ratio for all placebos, the test can be interpreted as looking at how likely the result obtained for a single treated case with a synthetic control analysis could have occurred by chance given no treatment. For more detailed description, see Abadie, Diamond, and Hainmueller (2011, 2014).

References

Abadie, A., Diamond, A., Hainmueller, J. (2014). Comparative Politics and the Synthetic Control Method. American Journal of Political Science Forthcoming 2014.

Synthetic : An R Package for Synthetic Control Methods in Comparative Case Studies. Journal of Statistical Software 42 (13) 1–17.

Abadie, A., Diamond, A., Hainmueller, J. (2011). Synth: An R Package for Synthetic Control Methods in Comparative Case Studies. Journal of Statistical Software 42 (13) 1–17.

Abadie A, Diamond A, Hainmueller J (2010). Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. Journal of the American Statistical Association 105 (490) 493–505.

See Also

generate.placebos, mspe.test, plot_placebos, synth

Examples

```
## Not run: ## Example with toy data from 'Synth'
library(Synth)
# Load the simulated data
data(synth.data)

# Execute dataprep to produce the necessary matrices for 'Synth'
dataprep.out<-
  dataprep(
    foo = synth.data,
    predictors = c("X1"),
    predictors.op = "mean",
    dependent = "Y",
    unit.variable = "unit.num",
    time.variable = "year",
    special.predictors = list(
      list("Y", 1991, "mean")
    ),
    treatment.identifier = 7,
    controls.identifier = c(29, 2, 13, 17),
    time.predictors.prior = c(1984:1989),
    time.optimize.ssr = c(1984:1990),
    unit.names.variable = "name",
```

```
      time.plot = 1984:1996
)

# run the synth command to create the synthetic control
synth.out <- synth(dataprep.out, Sigf.ipop=2)

## run the generate.placebos command to reassign treatment status
## to each unit listed as control, one at a time, and generate their
## synthetic versions. Sigf.ipop = 2 for faster computing time.
## Increase to the default of 5 for better estimates.
tdf <- generate.placebos(dataprep.out,synth.out, Sigf.ipop = 2)

## Test how extreme was the observed treatment effect given the placebos:
ratio <- mspe.test(tdf)
ratio$p.val

mspe.plot(tdf, discard.extreme = FALSE)

## End(Not run)
```

---

mspe.test   *Function to compute the post/pre treatment MSPE ratio for the treated*
                             unit and placebos

Description

Computes the post/pre treatement mean square prediction error ratio for a treated unit in a synthetic control analysis and all placebos produced with [generate.placebos](#). Returns a matrix with ratios and a p-value of how extreme the treated unit's ratio is in comparison with that of placebos. Equivalent to a significance testing of a synthetic controls result.

Usage

```
mspe.test(tdf, discard.extreme = FALSE, mspe.limit = 20)

mspe_test(tdf, discard.extreme = FALSE, mspe.limit = 20)
```

Arguments

tdf   An object constructed by [generate.placebos](#)

discard.extreme

                   Logical. Whether or not placebos with high pre-treatement MSPE should be
                   excluded from the count and significance testing.

mspe.limit   Numerical. Used if discard.extreme is TRUE. It indicates how many times the
                   pretreatment MSPE of a placebo should be higher than that of the treated unit to
                   be considered extreme and discarded. Default is 20.

Details

Post/pre-treatment mean square prediction error ratio is the difference between the observed out-
come of a unit and its synthetic control, before and after treatment. A higher ratio means a small
pre-treatment prediction error (a good synthetic control), and a high post-treatment MSPE, meaning
a large difference between the unit and its synthetic control after the intervention. By calculating
this ratio for all placebos, the test can be interpreted as looking at how likely the result obtained
for a single treated case with a synthetic control analysis could have occurred by chance given no
treatement. For more detailed description, see Abadie, Diamond, and Hainmueller (2011, 2014).

Value

p.val The p-value of the treated unit post/pre MSPE ratio. It is the proportion of units (placebos
and treated) that have a ratio equal or higher that of the treated unit

test Dataframe with two columns. The first is the post/pre MSPE ratio for each unit. The second
indicates unit names

See Also

generate.placebos, mspe.plot, synth

Examples

```
## Not run: ## Example with toy data from 'Synth'
library(Synth)
# Load the simulated data
data(synth.data)

# Execute dataprep to produce the necessary matrices for 'Synth'
dataprep.out<-
  dataprep(
    foo = synth.data,
    predictors = c("X1"),
    predictors.op = "mean",
    dependent = "Y",
    unit.variable = "unit.num",
    time.variable = "year",
    special.predictors = list(
      list("Y", 1991, "mean")
    ),
    treatment.identifier = 7,
    controls.identifier = c(29, 2, 13, 17),
    time.predictors.prior = c(1984:1989),
    time.optimize.ssr = c(1984:1990),
    unit.names.variable = "name",
    time.plot = 1984:1996
)

# run the synth command to create the synthetic control
synth.out <- synth(dataprep.out, Sigf.ipop=2)
```

```
## run the generate.placebos command to reassign treatment status
## to each unit listed as control, one at a time, and generate their
## synthetic versions. Sigf.ipop = 2 for faster computing time.
## Increase to the default of 5 for better estimates.
tdf <- generate.placebos(dataprep.out,synth.out, Sigf.ipop = 2)

## Test how extreme was the observed treatment effect given the placebos:
ratio <- mspe.test(tdf)
ratio$p.val

mspe.plot(tdf, discard.extreme = FALSE)

## End(Not run)
```

---

multiple.synth   *Function to Apply Synthetic Controls to Multiple Treated Units*

Description

Generates one synthetic control for each treated unit and calculates the difference between the
treated and the synthetic control for each. Returns a vector with outcome values for the synthetic
controls, a plot of average treatment effects, and if required generates placebos out of the donor
pool to be used in conjunction with plac.dist. All arguments are the same used for dataprep in
the Synth package, except for treated.units, treatment.time, and generate.placebos.

Usage

```
multiple.synth(
  foo,
  predictors,
  predictors.op,
  dependent,
  unit.variable,
  time.variable,
  special.predictors,
  treated.units,
  control.units,
  time.predictors.prior,
  time.optimize.ssr,
  unit.names.variable,
  time.plot,
  treatment.time,
  gen.placebos = FALSE,
  strategy = "sequential",
  Sigf.ipop = 5
)

multiple_synth(
```

```
  foo,
  predictors,
  predictors.op,
  dependent,
  unit.variable,
  time.variable,
  special.predictors,
  treated.units,
  control.units,
  time.predictors.prior,
  time.optimize.ssr,
  unit.names.variable,
  time.plot,
  treatment.time,
  gen.placebos = FALSE,
  strategy = "sequential",
  Sigf.ipop = 5
)
```

Arguments

`foo`	Dataframe with the panel data.
`predictors`	Vector of column numbers or column-name character strings that identifies the predictors' columns. All predictors have to be numeric.
`predictors.op`	A character string identifying the method (operator) to be used on the predictors. Default is mean.
`dependent`	The column number or a string with the column name that corresponds to the dependent variable.
`unit.variable`	The column number or a string with the column name that identifies unit numbers. The variable must be numeric.
`time.variable`	The column number or a string with the column name that identifies the period (time) data. The variable must be numeric.
`special.predictors`	
	A list object identifying additional predictors and their pre-treatment years and operators.
`treated.units`	A vector identifying the `unit.variable` numbers of the treated units.
`control.units`	A vector identifying the `unit.variable` numbers of the control units.
`time.predictors.prior`	
	A numeric vector identifying the pretreatment periods over which the values for the outcome predictors should be averaged.
`time.optimize.ssr`	
	A numeric vector identifying the periods of the dependent variable over which the loss function should be minimized between each treated unit and its synthetic control.
`unit.names.variable`	
	The column number or string with column name identifying the variable with units' names. The variable must be a character.

time.plot	A vector identifying the periods over which results are to be plotted with [path.plot](#)
treatment.time	A numeric value with the value in time.variable that marks the intervention.
gen.placebos	Logical. Whether a placebo (a synthetic control) for each unit in the donor pool should be constructed. Will increase computation time.
strategy	The processing method you wish to use "sequential", "multicore" or "multisession" . Use "multicore" or "multisession" to parallelize operations and reduce computing time. Default is sequential. Since SCtools >= 0.3.2 "multiprocess" is deprecated.
Sigf.ipop	The Precision setting for the ipop optimization routine. Default of 5.

Details

The function runs [dataprep](#) and [synth](#) for each unit identified in treated.units. It saves the vector with predicted values for each synthetic control, to be used in estimating average treatment effects in applications of Synthetic Controls for multiple treated units.

For further details on the arguments, see the documentation of [Synth](#).

Value

Data frame. Each column contains the outcome values for every time-point for one unit or its synthetic control. The last column contains the time-points.

Examples

```
## Using the toy data from 'Synth':

library(Synth)
data(synth.data)
set.seed(42)

multi <- multiple.synth(foo = synth.data,
                        predictors = c("X1"),
                        predictors.op = "mean",
                        dependent = "Y",
                        unit.variable = "unit.num",
                        time.variable = "year",
                        treatment.time = 1990,
                        special.predictors = list(
                          list("Y", 1991, "mean")
                        ),
                        treated.units = c(2,7),
                        control.units = c(29, 13, 17),
                        time.predictors.prior = c(1984:1989),
                        time.optimize.ssr = c(1984:1990),
                        unit.names.variable = "name",
                        time.plot = 1984:1996, gen.placebos =  FALSE,
                        Sigf.ipop = 2)
## Plot with the average path of the treated units and the average of their
## respective synthetic controls:
```

```
multi$p
```

---

plac.dist   *Plot the distribution of placebo samples for synthetic control analysis*
                       with multiple treated units.

Description

Takes the output object of `multiple.synth` creates a distribution of placebo average treatment effects, to test the significance of the observed ATE. Does so by sampling k placebos (where k = the number of treated units) nboots times, and calculating the average treatment effect of the k placebos each time.

Usage

```
plac.dist(multiple.synth, nboots = 500)

plac_dist(multiple.synth, nboots = 500)
```

Arguments

`multiple.synth`	multiple.synthAn object returned by the function `multiple.synth`
`nboots`	Number of bootstrapped samples of placebos to take. Default is `500`. It should be higher for more reliable inference.

Value

p The plot.

att.t The observed average treatment effect.

df Dataframe where each row is the ATT for one bootstrapped placebo sample, used to build the distribution plot.

p.value Proportion of bootstrapped placebo samples ATTs which are more extreme than the observed average treatment effect. Equivalent to a p-value in a two-tailed test.

Examples

```
## Not run:
## Using the toy data from Synth:
library(Synth)
data(synth.data)
set.seed(42)
## Run the function similar to the dataprep() setup:
multi <- multiple.synth(foo = synth.data,
                        predictors = c("X1", "X2", "X3"),
                        predictors.op = "mean",
                        dependent = "Y",
```

```
                          unit.variable = "unit.num",
                          time.variable = "year",
                          treatment.time = 1990,
                          special.predictors = list(
                            list("Y", 1991, "mean"),
                            list("Y", 1985, "mean"),
                            list("Y", 1980, "mean")
                          ),
                          treated.units = c(2,7),
                          control.units = c(29, 13, 17, 32),
                          time.predictors.prior = c(1984:1989),
                          time.optimize.ssr = c(1984:1990),
                          unit.names.variable = "name",
                          time.plot = 1984:1996, gen.placebos = TRUE, Sigf.ipop = 2,
                          strategy = 'multicore' )

## Plot with the average path of the treated units and the average of their
## respective synthetic controls:

multi$p

## Bootstrap the placebo units to get a distribution of placebo average
## treatment effects, and plot the distribution with a vertical line
## indicating the actual ATT:

att.test <- plac.dist(multi)
att.test$p

## End(Not run)
```

---

plot_placebos	*Function to plot placebos of a synthetic control analysis*

Description

Creates plots with the difference between observed units and synthetic controls for the treated and control units. See Abadie, Diamond, and Hainmueller (2011).

Usage

```
plot_placebos(
  tdf = tdf,
  discard.extreme = FALSE,
  mspe.limit = 20,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  alpha.placebos = 1,
  ...
)
```

Arguments

`tdf`	An object with a list of outcome values for placebos, constructed by [generate.placebos]().
`discard.extreme`	
	Logical. Whether or not units with high pre-treatement MSPE should be excluded from the plot. Takes a default of `FALSE`.
`mspe.limit`	Numerical. Used if `discard.extreme` is `TRUE`. It indicates how many times the pre-treatment MSPE of a placebo should be higher than that of the treated unit to be considered extreme and discarded. Default is `20`.
`xlab`	Character. Optional. Label of the x axis.
`ylab`	Character. Optional. Label of the y axis.
`title`	Character. Optional. Title of the plot.
`alpha.placebos`	the transparency setting, default of 1
`...`	optional arguments (currently not used)

See Also

[generate.placebos](), [gaps.plot](), [synth](), [dataprep]()

Examples

```
## Not run: ## Example with toy data from Synth
library(Synth)
# Load the simulated data
data(synth.data)

# Execute dataprep to produce the necessary matrices for synth
dataprep.out<-
  dataprep(
    foo = synth.data,
    predictors = c("X1"),
    predictors.op = "mean",
    dependent = "Y",
    unit.variable = "unit.num",
    time.variable = "year",
    special.predictors = list(
      list("Y", 1991, "mean")
    ),
    treatment.identifier = 7,
    controls.identifier = c(29, 2, 13, 17),
    time.predictors.prior = c(1984:1989),
    time.optimize.ssr = c(1984:1990),
    unit.names.variable = "name",
    time.plot = 1984:1996
)

# run the synth command to create the synthetic control
synth.out <- synth(dataprep.out, Sigf.ipop=2)
```

```
## run the generate.placebos command to reassign treatment status
## to each unit listed as control, one at a time, and generate their
## synthetic versions. Sigf.ipop = 2 for faster computing time.
## Increase to the default of 5 for better estimates.
tdf <- generate.placebos(dataprep.out,synth.out, Sigf.ipop = 2, strategy='multicore')

## Plot the gaps in outcome values over time of each unit --
## treated and placebos -- to their synthetic controls

p <- plot_placebos(tdf,discard.extreme=TRUE, mspe.limit=10, xlab='Year')
p

## End(Not run)
```

| synth.data | *Synth Data Synthetic data that can be used to explore SCtools.* |

Description

Synth Data Synthetic data that can be used to explore SCtools.

Usage

```
synth.data
```

Format

a data.frame with 168 rows and 7 columns:

unit.num The experimental unit number

year year

name name of the experimental unit

Y outcome of interest

X1 Covariate 1

X2 Covariate 2

X3 Covariate 3

Index