**Documentation:** Package "synthpy"

**Version** 0.55

**Author:** Prachya Tantivatana (prachyaatan@gmail.com)

**Maintainer:** Prachya Tantivatana (prachyaatan@gmail.com)

**Description:** A package to conduct synthetic control analysis using Abadie, Diamond and Hainmueller (2014) framework. The code is developed with Hainmueller (2022) R package as guideline, but with additional features such as hypothesis testing and leave-one-out standard error. Synthetic control method is used for estimating an effect of a certain event to a single treated unit by constructing a synthetic control unit or a counterfactual that resemble the treated unit without the treatment from a group of control units, referred to as donor pool. The method creates synthetic control with a weight averaged of the control unit, where weight is sum up to one and is between 0 to one to avoid overfitting and allow researcher to interpret it as a mixture of control unit. Additionally, other characteristics can also be introduced into the process as predictors to improve the pretreatment fit of synthetic control. For optimization, cxvpy is implemented to compute weights for control units (W) whereas SciPy minimize function is used for the computation of predictors weight (v).

**GitHub link**: https://github.com/Prachyaatan/synthpy-by-Prachya

**Package imported**: cxvpy , matplotlib.pyplot , numpy, , pandas and scipy

**Notes**: This work is a part of UCLA Master of Quantitative Economics capstone project, Winter 2023

(1) **Function**: prep_data

**Description**: Transform a standard panel data into "prep_data" class that contains elements necessary for the computation of synthetic control with "synth" function.

**Argument**: (df = None, year = None, country = None, target = None, control = None,  begin = None, train = None,  validation = None, treatment = None,outcome = None, cov = None , method = "mean"):

   df: Panel data in form of pandas.dataframe.

   year: an integer that indicates period when the treatment occurs.

   country: a string that indicates the column name that identifies units in the data.

   target: a string that indicates the treatment unit.

   control: a list of control units.

   begin: an integer that indicates the first period to be included in the computation of predictors.

   train: an integer that indicates the first period to be included in the training dataset.

   validation: an integer that indicates the first period to be included in the validation dataset.

   treatment: an integer that indicates the period in which the treatment occurs.

   outcome: a string  that indicates the column name that contains the target variable.

   cov: a list of column name of predictors variables.

   method: a string that indicates the method used to compute predictor; default is mean.

**Output:**

   a variable with prep_data class

   prep_data.Y0: a numpy.array of outcome variable for control unit.

   prep_data.Y1: a numpy.array of outcome variable for treated units.

   prep_data.X0: a numpy.array of characteristic variables for control units.

   prep_data.X1: a numpy.array of characteristic variables for treated units.

   prep_data.X0_train: a numpy.array of predictors for control units for training set.

   prep_data.X1_train: a numpy.array of predictors for treated units for training set.

   prep_data.begin: an integer that indicates the first period to be included in the computation of predictors.

   prep_data.train: an integer that indicates the first period to be included in the training dataset.

prep_data.validation: an integer that indicates the first period to be included in the validation dataset.

prep_data.treatment: an integer that indicates the period in which the treatment occurs.

prep_data.lastperiod: an integer that indicates the last period in the dataset.

prep_data.year_norm: an integer used to normalize time period.

prep_data.predictor: a list of predictor variables.

prep_data.control_unit: a list of control units.


(2) **Function**: synth

**Description**: Compute and conduct synthetic control using prep_data class variables. This function will compute the synthetic controls and weight W and V.

**Argument**: (PD = None, custom_V = None , method = "NELDER-MEAD")

 PD: a prep_data class

 custom_V: a list of custom V or characteristic variable weight

 method: a string that indicates scipy optimization method, default is NELDER-MEAD.

**Output**:

synth.V_star: a pandas.dataframe containing the optimal V.

synth.Wstar: a pandas.dataframe containing the optimal W.

synth.synthetic: a numpy.array containing the synthetic control.

synth.actual: a numpy.array containing the actual outcome data.

synth.pretreat_rmse: a numpy.array containing RMSE in the pretreatment period.

synth.TE: a numpy.array containing the difference between actual and synthetic control.


(3) **Function**: synth_plot

**Description**: Plot actual data and the synthetic control using matplotlib.pyplot package.

**Argument**: (result, xaxis = "year", yaxis = "variable", legend_loc = "lower left", plot_title = "plot")

 result: a synth class that is the result from synthetic control function "synth".

 xaxis: a string for the title of xlabel of the plot.

 yaxis: a string for the title of ylabel of the plot.

 legend_loc: a string that indicates the location of the legend box.

 plot_title: a string for the title of the plot.

**Output**:

 A plot of actual outcome data and the synthetic control on the same graph.

(4) **Function**: synth_diffplot

**Description**: Plot the difference between the actual data and the synthetic control using matplotlib.pyplot package.

**Argument**: (result, xaxis = "year", yaxis = "variable", plot_title = "plot")

       result: a synth class that is the result from synthetic control function "synth".

       xaxis: a string for the title of xlabel of the plot.

       yaxis: a string for the title of ylabel of the plot.

       plot_title: a string for the title of the plot.

       df: Panel data in form of pandas.dataframe.

**Output**:

       A plot of the difference between actual outcome data and synthetic control.


(5) **Function**: loo

**Description**: Compute the pretreatment period of RMSE of leave-one-out (loo)  and then plot the result. Use for robustness test and assessment of control unit.

**Argument**:  (df, year, country, target, control, begin, train, validation, treatment,

             outcome, cov, use_Vbaseline = True, opt_method = "NELDER-MEAD",

             legend_loc = "lower left", pred_method = "mean")

       year: an integer that indicates period when the treatment occurs.

       country: a string that indicates the column name that identifies units in the data.

       target: a string that indicates the treatment unit.

       control: a list of control units.

       begin: an integer that indicates the first period to be included in the computation of predictors.

       train: an integer that indicates the first period to be included in the training dataset.

       validation: an integer that indicates the first period to be included in the validation dataset.

       treatment: an integer that indicates the period in which the treatment occurs.

       outcome: a string  that indicates the column name that contains the target variable.

       cov: a list of column name of predictors variables.

       use_Vbaseline: a tubule indicates whether to optimize v for leave-one-out or use the baseline result v instead for faster computation.

       opt_method: a string that indicates the method used to optimize v, default is NELDER-MEAD.

       legend_loc: a string that indicates the location of the legend box.

pred_method: a string that indicates the method used to compute predictor; default is mean.

**Output:**

1) Return a pandas.dataframe of RMSE during the pretreatment period for each unit dropped.

2) Plot the baseline along with leave-one-out result.

(6) **Function**: placebo_permu

**Description**: Compute standard error for hypothesis test of synthetic control analysis using permutation between periods prior to treatment and control units.

**Argument**: (df, year, country, target, control, begin, train, validation, treatment,

outcome, cov,  placebo_start, use_Vbaseline = True, pb_pret_split = 0.5

opt_method = "NELDER-MEAD", pred_method = "mean", alpha = 0.05)

year: an integer that indicates period when the treatment occurs.

country: a string that indicates the column name that identifies units in the data.

target: a string that indicates the treatment unit.

control: a list of control units.

begin: an integer that indicates the first period to be included in the computation of predictors.

train: an integer that indicates the first period to be included in the training dataset.

validation: an integer that indicates the first period to be included in the validation dataset.

treatment: an integer that indicates the period in which the treatment occurs.

outcome: a string  that indicates the column name that contains the target variable.

cov: a list of column name of predictors variables.

use_Vbaseline: a tubule indicates whether to optimize v for leave-one-out or use the baseline result v instead for faster computation.

pb_pret_split: a float  that represents the ratio on splitting between training and validation period for the place. For example, 0.75 will split training:validation during pretreatment period to 25:75.

opt_method: a string that indicates the method used to optimize v, default is NELDER-MEAD.

pred_method: a string that indicates the method used to compute predictor; default is mean.

alpha: a float number for the confidence level.

**Output**:

1) Return a pandas.dataframe that contains the baseline result along with average, standard error and interval from the permutation of placebo assignment.

2) Plot baseline result with interval computed from the permutation of placebo assignment.

## Reference

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

Jens Hainmueller, Alexis Diamond (2022). Package "Synth". Link: http://cran.r-project.org/web/packages/Synth/index.html