PSMatch

Features

psmatch is a package for implementing propensity score matching in Python 3

The following functionality is included in the package:

- Calculation of propensity scores based on a specified model.
- Matching of k controls to each treatment case with four different methods.
- Use a caliper to control each treatment case.
- Matching with or without replacement.
- Performing weight processing on matched data.
- Calculate the p value on the basis of the statistic.
- Evaluation of the matching process using statistical methods.

Technology

psmatch uses a number of open source projects to work properly:

- pandas
- numpy
- matplotlib
- scipy
- statsmodels

psmath itself is open source with a public repository on GitHub

Usage

```
#Instantiate PSMatch object
>>> m=PSMatch(path,control, covariants, dependents=None)


'''
Utilizes a logistic regression framework to calculate propensity scores based on
a specified model.
Results
-------
A Pandas DataFrame containing raw data and propensity scores.
'''
>>> m.calculate_proprnsity_scores()


'''
Performs propensity score matching.
Parameters
----------
df : Pandas DataFrame
    the attribute returned by the prepare_data() function
capliper: float
```

```
    value between 0 and 1, the maximum difference in scores allowed during
pairing, value between 0 and 1
replace: bool
    if not allow duplicate pairing
k:
    how many pairs of control group data are generated for each experimental
group
Returns
-------
matches : Pandas DataFrame
    the Match object attribute describing which control IDs are matched
    to a particular treatment case.
matched_data: Pandas DataFrame
    the Match object attribute containing the raw data for only treatment
    cases and their matched controls.
'''
>>> m.match(caliper=None, replace=False, k=1)


'''
Performs weight process
Parameters
----------
df : Pandas DataFrame
    the attribute returned by the match() function
method: str
    Weight processing method, including 'IPTW' 'SMRW-T' 'SMRW-C' 'OW' 'IPTW-P'
'''
>>> m.weighted_process(self,method=None)


'''
Conducts chi-square tests to verify statistically that the cases/controls
are well-matched on the variables of interest.
Parameters
----------
df : Pandas DataFrame
    df to evaluate p-value
method: str
    Weight processing method, including 'IPTW' 'SMRW-T' 'SMRW-C' 'OW' 'IPTW-P'
if_show: bool
    if show p-value condition
Return
----------
A dictionary contains the p-values of each variable and the number of variables
that failed the test
'''
>>> m.evaluate_p_value(self, df,if_show=True)


'''
Evaluate the mean, variance, SMD of coveriates and the results of dependent
variables
Parameters
----------
```

```
df : Pandas DataFrame
    The DataFrame to be evaluated
Return
----------
A Pandas DataFrame contains evaluate results
'''
>>> m.evaluate_dependent(df)


'''
Draw a line chart of smd and a histogram of the number of variables that failed
the test,
showing the matching effect of each method
Parameters
----------
if_weighed: bool
    if weight processed
'''
>>> m.plot_matching_efficiency(if_weighed=True)
```