MicroDataCleaning

Desktop app and Python program to make do-files for data cleaning from layout tables as well as the resulting do-files

Overview

glmice automatically makes do-files to import fixed-length data using the layout tables from MHLW or MIC or possibly other ministries and government offices.

- glmice.exe: Desktop app based on main.py.
- main.py: Main program to make do-files for data cleaning from layout tables.
- do-file/: do-files used to import fixed-length data. Some files are generated by glmice while other files are provided by some contributers. See docs/contributers.txt for the full list of the contributers.

 demo

Requirement

- glmice.exe works without Python (Tested only under Windows 10).
- main.py works with Python3. You can download it from here. In addition, you need to install `python-Levenshtein`.

```
pip install python-Levenshtein
```

Usage

glmice

1. Download dist.zip from release page, and launch dist/glmice/glmice.exe after unzipping dist.zip.
2. Fill input information (i.e., 'Excel file', 'Excel sheet index', 'Output file', 'Data file')
3. Push 'Add' button. Then, the information is added to the list. You can check the added information by selecting it in the list and pushing 'Check' button.
4. Similarly, you can remove, if necessary, some inputs in the list by using 'Remove' button.
5. After adding all input information to the list, you may or may not fill 'Survey name', which could help to harmonize the variable names across data in different years.
6. Push 'Run' button.

You can skip Steps 1-3 by making a csv file with the necessary input information. Use docs/input_list_template.csv in such cases. Then, import that csv file (File > Import).

When launching the application, you might see the following error message:

> **Windows protected your PC**
> Windows Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

In such cases, please click `More info` (or `詳細情報`) and then `Run anyway` (or `実行`).

Main program

main.py is the main program to make do-files for data cleaning from layout tables.
(The program is not currently registered in PyPI, so please download Source code (zip/tar.gz) from release page)

```
Main(infile_list, index_list, outfile_list, data_list,
        xls=False, reservation=0.2, SurveyName=None)
"""
    infile_list: list of input files.
    (i.e., Excel files for the layout tables)

    index_list: list of Excel sheet indices (Count from 0).

    outfile_list: list of output file names.
    If outfile_list = ['File1', 'File2'], the resulting output files are
    'File1_const.do', 'File1_const.do', and so on.

    data_list: list of the raw data (in most cases, .txt or .dat files).

    xls: whether to make 'cleaned' layout sheet.
    (This option may be useful if you are an R user.)

    reservation: reservation distance which is used to make a do-file to
    harmonize several data. If the Levenshtein distance of two variables
    (from different survey years) is more than this reservation distance,
    those variables are judged to be different variables even if they are
    the closest pair. (You may not want to change this parameter, and rather
    making synonyms will be more useful.)

    SurveyName: Name of the survey (in Japanese). Specifying the survey name
    could improve variable matching process when harmonizing several data.

    Method defined here:

    run(): Run the program and make do-files.
"""
```

Output files

- FILENAME_const.do: import fixed-length data.
- FILENAME_var.do: put variable labels.
- FILENAME_val.do: put value labels.
- FILENAME_validate.do: check if the data seems correctly imported.

- rename.do: rename variable names to append several data.
- master.do: run all do-files above, append data and save the dta file. When using master.do, set global macro `DoFilePathTemp` and `DataFilePathTemp` appropriately. (rename.do and master.do are made in the root directory for `outfile_list`)
- rename.xls: summary of the variable matching result.
- FILENAME_layout.xls: clean layout table made when `xls=True`

Example code

```
from main import Main

infile_list = [
    'RootPath/Layout/XXX.xls',
    'RootPath/Layout/YYY.xls',
    'RootPath/Layout/ZZZ.xlsx'
]

index_list = [0, 0, 0]

outfile_list = [
    'RootPath/do-file/XXX/XXX',
    'RootPath/do-file/YYY/YYY',
    'RootPath/do-file/ZZZ/ZZZ'
]

data_list = [
    'RootPath/data/raw/dataXXX.txt',
    'RootPath/data/raw/dataYYY.txt',
    'RootPath/data/raw/dataZZZ.txt'
]

main = Main(
    infile_list, sheet_index_list, outfile_list, data_list
)
main.run()
```

Before running master.do ...

1. Add ado/CheckAppendValidity.ado and ado/DestringAll.ado to an appropriate directory. You can check the path by typing `adopath` on Stata.
2. Check if rename.do correctly works. To that end, rename.xls may be useful. In rename.xls, the cells are highlighted if the variable description is not identical to the previous or next year data.

Harmonize variable names across survey years

Although the program automatically implement fuzzy variable matching based on variable description and, if any, variable names, the matching is not very successful in some cases. You can improve the matching quality by making survey-specific synonym list such as this one, in which

each word in a list is replaced by the first element of the list when Levenshtein distance is measured.

The necessary procedure is as follows:

1. Make/modify a survey-specific thesaurus at 'python/module/Writers/VarNameThesaurus/Thesaurus/', and the variable name defined there should be `thesaurus_jargon`.
2. Then, modify `StrDistMeasureFactory` class in '/python/module/Writers/VarNameThesaurus/StrDistMeasure.py'. If the file name is 'NewThesaurus.py' and the survey name is 'xxxSurvey', you need to add

```
elif SurveyName == 'xxxSurvey':
    from .Thesaurus.NewThesaurus import thesaurus_jargon
```

to the if clause in `StrDistMeasureFactory` class. 3. Run the main program with the SurveyName option (i.e., `SurveyName='xxxSurvey'`).

This survey-specific synonym list will be quite useful for other users, so your contribution to provide it would be greatly appreciate.

Remarks

- All file names are recommended to be specified with full paths. (Windows users should not use `\` (backslash/yen sign) to specify the path. Use `/` instead.)
- If the specified directory does not exist, it is made automatically.
- The generated do-files do not necessarily correct. If you find something wrong regarding the program or the resulting do-file(s), please notify me via Issue.
- _validate.do checks if each variable has the values that it is supposed to have (only for categorical variables).
- rename.do is generated by finding a variable with a similar variable description and, if any, a similar variable name from the base data. So it is quite likely that some variables are renamed incorrectly. Please check and modify rename.do.
- Although value labels are put for each individual data, the labels are not put for the appended data, because the categories for each variable are very likely to be inconsistent across years.
- In addition, there is no file to make variable values consistent across different data.

License

Author

Takahiro-Toriyabe