Assignment lab 1 - Development environment setup and dataset statistics

Version: 31.08.2020

The purpose of this assignment

The goal of this lab is to check that the student has knowledge in the following topics:

- Development environment setup.
- Working with datasets based on pandas and csv.
- Statistics refreshment.
- Dataset statistics and visualisation with pandas, numpy and matplotlib.

Task

1. Install Python Anaconda distribution (or Python with required modules).
2. Create software project in GitLab. Use one of to https://gitlab.cs.ttu.ee or https://gitlab.com. See class 1 material for details.
3. Print out python and available modules versions.
4. Read dataset in project template data directory to pandas data frame and print out data overview.
5. Print out all possible `County` column values and counts.
6. Print out all `Type` column unique values.
7. Create new pandas data frame and extract from previous data frame only SME (Small and medium-sized enterprises) data by limiting number on employees.
8. Extract column (pandas series) `Number of employees` from previously created data frame.
9. Calculate number of employees mean, median, mode, standard deviation and quartiles and print out the results.
10. Plot SME Number of employees histogram including lines showing mean, median and mode.
11. Plot SME Number of employees box plot.
12. Find correlation between `Number of employees` and `Labour taxes and payments` and print out correlation matrix.
13. Plot SME Number of employees and Labour taxes and payments correlation scatter plots showing correlation.

Guidelines

Project repository structure and files

Project shall consist of following files (excluding directories `.git` and also `builds` if local gitlab-runner is used).

```
.
├── .gitignore
├── .gitlab-ci.yml
├── .pylintrc
├── common
|   ├── describe_data.py
|   └── test_env.py
├── data
|   └── tasutud_maksud_2020_ii_kvartal_eng.csv
├── lab1.py
└── results
    └── .placeholder
```

`lab1.py` shall be created by student.

For `.gitignore`, `.gitlab-ci.yml`, `pylintrc`, `data` and `common` files from lab1 template shall be used.

`lab1.py` template is available via snippet Lab 1 Dataset statistics template to avoid merge conflicts when working with multiple remotes.

NB! Be aware that if you want to use different file names you need to modify CI configuration and tests accordingly.

Data set and read to pandas data frame

Project template contains dataset `tasutud_maksud_2020_ii_kvartal_eng.csv`. Although file name is in Estonian, content is in english. It contains information about 2019 Q2 taxes payed by Estonian companies and other entities. Data set is public and content description is available in Tax and customs board web page Taxes paid, turnover and number of employees.

You do not have to use this dataset! You can use any other. In such case data set file shall be added (committed) to the repository. Used data set shall contain at least 2 numeric columns for correlation calculation and several hundreds of rows is desired. When you use different dataset you do not need to implement categories printouts if you dataset does not contain any categorisable data columns.

Data can be read by calling function `read_data()` and providing file name as an argument:

```
df = read_data('data/tasutud_maksud_2020_ii_kvartal_eng.csv')
```

Be aware if you are using your own data set you need to most likely change, separator, delimiter, encoding and so on.

Data set overview print

You can print out dataset overview by calling function `print_overview()` in module `common.describe_data` by providing data frame as and argument. Module shall be "imported" first. Add imports to beginning of the file!

```
import common.describe_data as describe_data

...
describe_data.print_overview(df)
```

Counties and types printout

Printout counties column values and counts and types unique values as follows:

```
    # Print all possible values with counts in column 'County' with help of groupby
    print(df.groupby('County').size(), '\n')

    # Print all unique values in column 'Type'
    print('Types:', df['Type'].unique(), '\n')
```

Code already exists in template but there can be need to change variables names depending your project variables naming.

There is need to change Pandas display value before printing County column values to not truncate printouts.

```
    # Remove maximum printed rows limit. Otherwise next print is truncated
    pd.options.display.max_rows = None
```

SME data extraction

Following statement extracts rows where `Type` column is `Company` and `Number of employees` column numeric value is 1 .. 250. This also eliminates rows with missing number of employees and 0 employees. It is intentional for this particular case. If you are using your own data set there can be no need to filter data at all.

```
sme_df = df[(df['Type'] == 'Company') & (
        df['Number of employees'] > 0) & (df['Number of employees'] < 250)]
```

Note that new data frame is created and will be used from now on.

Mean, median, mode, standard deviation and quartiles printouts

Calculate values based on class 2 materials and print out the values. You need to create variables because those are used in histogram to draw vertical lines.

Following is example for mean:

```
# Calculate mean to variable sme_employees_mean and print the value
sme_employees_mean = sme_employees.mean()
print('SME Number of employees mean: ', sme_employees_mean)
```

You can use `.to_string()` for quartiles print as follows:

```
print('SME Number of employees quartiles:')
print(sme_employees_quartiles.to_string())
print('\n')
```

You can use same for other pandas series in future.

You need to add expressions below each particular comment in `lab1.py` template.

Plotting histogram, box plot and correlation scatter plots

Code needed for plots exists already in `lab1.py` template. There can be need to rename variables your project variables naming.

Plots are saved to directory `results` and shall be not committed to repository.

Feel free to redesign plots and try out different `matplotlib` features.

Correlation matrix and matrix printout

Extend `lab1.py` to calculate correlation matrix for `Number of employees` and `Labour taxes and payments` and print out as follows:

```
employees_labor_taxes_correlation = sme_df[['Number of employees',
                                            'Labour taxes and payments']].corr()
```

Printouts example

Printouts example is available in snippet Lab 1 Dataset statistics printouts example.

Feel free to customise and change your printouts.

Automation and GitLab CI stages

- Check-files
  - Tests existence of required files and fail if all files are not present.
  - List repository files excluding `.git` and `build` directories.
- Lint
  - Test `lab1.py` formatting with `pep8`.
  - Lint `lab1.py` with `pylint` by using configuration from file `.pylintrc`.
- Run-lab
  - Run `lab1.rb`

Content of results directory is archived as build artefacts and can be downloaded.

Formatting and lint

autopep8 is used to test code formatting. autopep8 is supported by VS Code. For other editors it can be installed with conda:

```
$ conda install -c conda-forge autopep8
```

To run formatter from command line:

```
$ autopep8 --in-place lab1.py
```

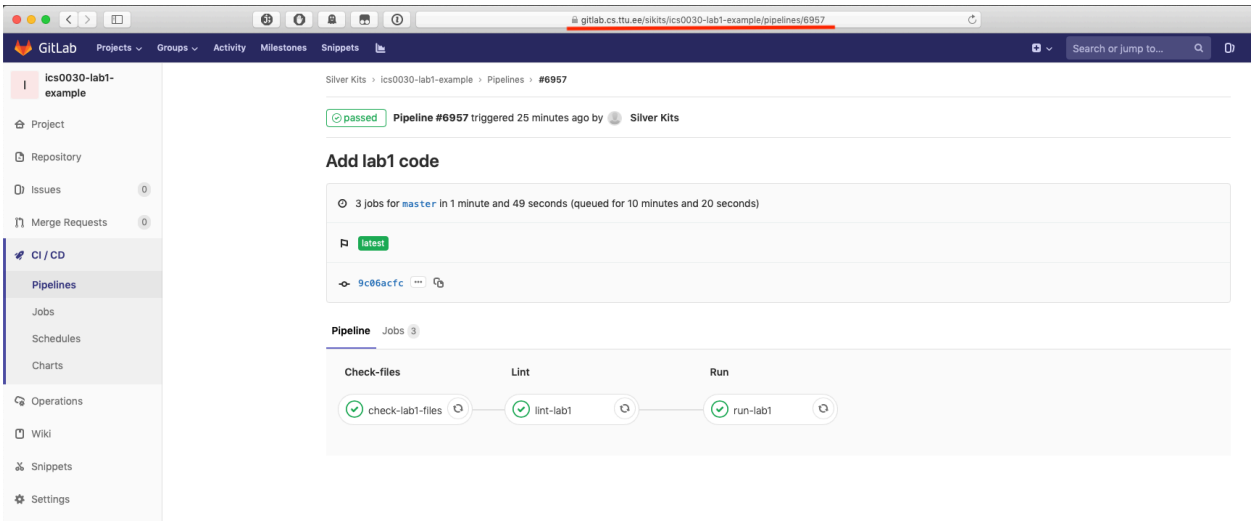
pylint is used for lint. Project template contains `.pylintrc`. Settings in this file are inline with VSCode default settings.

To run pylint from command line:

```
$ pylint lab1.py
```

Submission instructions

1. Be sure that your pipeline succeeds before submitting assignment in Moodle.



2. Submit only link to the pipeline as an answer in Moodle. **Please make link HTML URL!**