Lab 09 - Creating CLI application

Script Languages

Learning goals

1. Use command line arguments to control Python program.
2. Analyse CSV files.
3. Generate XLSX files.

Exercises

1. Preparation

In this lab you will create Python application to read the dataset in CSV format, aggregate data read and finally prepare a report as an Excel file.

1. Download dataset in the CSV format from the web page

   https://www.kaggle.com/datasets/

   Choose dataset larger than 500KB. Coordinate your choices with other students. If you use the same dataset as other students, your lab grade will be divided by the total number of students who has used the same dataset. Put the URL of the dataset in the comment at the beginning of your application.

2. Reading CSV file

1. Pass a name of the dataset downloaded in the task 1.1 as an obligatory argument to the app, e.g

   ```
   python app9.py dataset.csv
   ```

   If the file name does not end with .csv extension stop the program with explanation of the problem. If the dataset file does not exist also stop with the explanation. Use argparse (https://docs.python.org/3/library/argparse.html) module to write a command-line interface.

2. Read and parse the content of the dataset. Develop data class to represent each row and a class to represent a container for all the rows.

3. Devise some operations on the chosen dataset (at least two for each of the following sections):

   1. statistical (e.g. average, sum, median etc.)

2. aggregation (e.g. spending by country, number of tweets by a day of the week, etc.)

3. summary (e.g. number of items, countries, tweets; total spending etc.)

3. Generate XLSX file

1. Install OpenPyXL module (https://openpyxl.readthedocs.io)

2. Use an optional argument `-o` to provide a name of the Excel file to be generated, e.g.

```
python app8.py dataset.csv -o report.xlsx
```

   Save results of operations created in task 2.3 in the Excel file. Use text formatting (eg. font face, color etc.). If an option `-o` is not used, display summary information on the screen.

3. Add an option `-h` to display help about your application and available options.

4. Extended version

1. Develop a set of tests to verify your implementation. Use Factory Boy (https://factoryboy.readthedocs.io/) to generate appropriate test data.

2. Use Click (https://click.palletsprojects.com/) to create command line interface instead of argparse.