University of Innsbruck – Department of Computer Science

Methods of Software- und Data Engineering

Prof. Dr. Michael Felderer, Yakup Ipek

EXERCISE SHEET 2

DEADLINE:

24 October 2018 - 3 p.m.

TOPICS:

Descriptive Analysis, Exploratory Analysis; Metric Computation, Visualization Techniques, Stackoverflow API

DESCRIPTION:

The following exercises allow you to gain hands-on experience with some python libraries that are widely used for data visualization.

SUBMISSION:

Please upload your solution in OLAT according to the following instructions:

Solve the tasks within either a single or multiple notebook files. Add explanations to your solutions using the Markdown cell. Feel free to use any feature provided by Jupyter Notebook to enhance your solution. Name your solution files according to the following designation rules:

Notebook-File:

When submitting your solution please adhere to the following structure: <Surname>_ExSheet_<Sheet Nr.>_<Task Nrs. Separated with a '–'>.<file-ending>; e.g. Ipek_ExSheet_1_1.ipynb, Ipek_ExSheet_1_1-2-3.ipynb

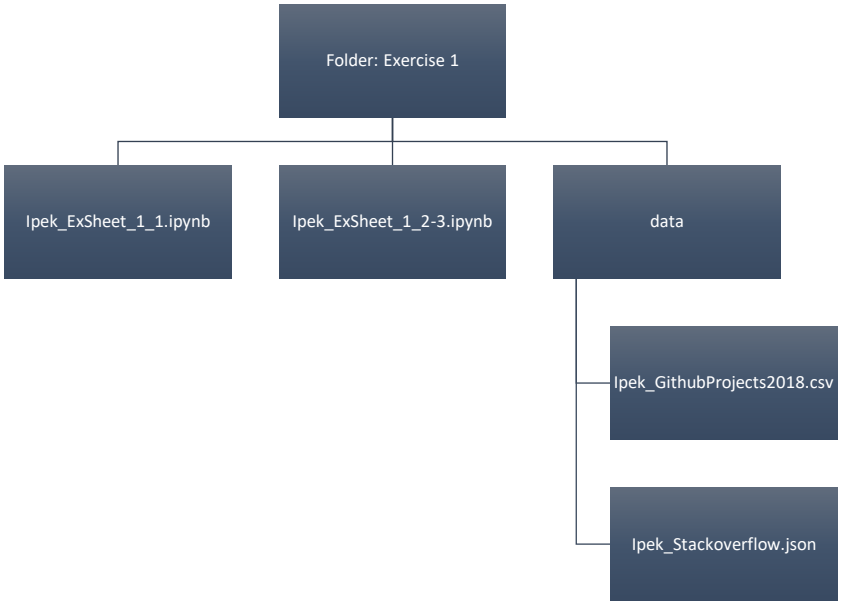
Data (CSV, JSON, etc.):

Any data source on your local filesystem that is explicitly accessed within your scripts should be placed within a **data** folder. The naming convention follows a slightly different structure than above: <Surname>_<Desired designation>.<file ending>

E.g. Ipek_GithubProjects2018.csv or Ipek_GithubProjects2018.json

PLEASE SUBMIT YOUR SOLUTION AS A SINGLE ZIPPED FILE AND NAME IT ACCORDINGLY:

<Surname>_ExSheet_<Sheet number>.zip/tar.gz; e.g. Ipek_ExSheet_1.zip or Ipek_ExSheet_1.tar.gz



LINKS:

Stackoverflow API	https://api.stackexchange.com/docs https://stackapi.readthedocs.io/en/latest/
Pandas	https://pandas.pydata.org/pandas-docs/stable/tutorials.html
Plotly	https://plot.ly/python/
Jupyter Notebook Shortcuts	https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/

PREREQUISITES:

In this exercise sheet the following python packages are required and need to be installed:

- jupyter
- numpy
- pandas
- py-stackexchange
- plotly
- StackAPI

Task 1: Metric Computation 2P

In this task you will implement some metrics and answer some questions.

Given the following the following Depth of Inheritance Trees (DIT) values for a set of classes:

[3, 5, 1, 2, 3, 2, 5, 5, 6, 5, 8, 8, 5, 4, 1, 8, 6, 3, 7, 2]

1. Compute the following Measures of Central Tendency:
   1.1. Mean
   1.2. Median
   1.3. Mode
   1.4. Geometric Mean
   1.5. Harmonic Mean
2. Compute the following Measures of Dispersion:
   2.1. Range
   2.2. Interquartile Range
   2.3. Sample Variance
   2.4. Sample Standard Deviation
   2.5. Mean Absolute Deviation
3. Generate a set of integer numbers of size **10 with at most one duplicate entry** whose mean, median and mode are equal

Note: For computing the metrics no libraries are allowed.

Task 2: Pandas 2P

Required: netflix_shows.csv (data set about Netflix; can be found on OLAT)

1. Read the netflix_shows.csv into a dataframe (in case a UnicodeDecodeError exception is thrown pass the following parameter as an argument: *encoding = 'ISO-8859-1'*)
2. The source data contains several ratings (such as TV-14, PG, etc.). Find out which rating has the highest frequency.
3. Find out the top five **release years** that have the highest average ***user rating score*** and attach the number of shows within that year

Task 3: Visualization (1/2) 1.5P

Note: This task builds upon the previous task.

1. Create a histogram showing the frequency of shows based on the ***release year***
2. Create boxplots for each year showing the ***user rating score***
3. Create a pie chart for depicting the percental share of the ***rating*** attribute

Task 4: Stackoverflow API 3P

```
from stackapi import StackAPI
#By default, StackAPI will return up to 500 items in a single call.
#It may be less than this, if there are less than 500 items to return.
# This is common on new or low traffic sites.

#Information on available Sites can be found at:
https://api.stackexchange.com/docs/sites
SITE = StackAPI('stackoverflow')
comments = SITE.fetch('comments')
```

Note: The usage of the API is limited as indicated on the following page:
https://api.stackexchange.com/docs/throttle

1. Fetch data from the ***stackoverflow*** Site based on the following query and parameters:

	method	sort	min	fromdatetime	todatetime	tagged
1	questions	votes	10	2018-01-01	2018-12-31	Python
2	questions	votes	10	2018-01-01	2018-12-31	C++

   Please note the limitations and try to retrieve the total number of results.

2. Save the questions in json-format into two files (e.g. python_questions_2018.json, cpp_questions_2018.json) and place them into the ***data*** folder

Task 5: Visualization (2/2) 1.5P

In this task the fetched data will be analyzed visually.

1. Create an overlapped histogram of both the python and cpp data. Use the **creation date** of the questions to bin your values.
2. Create a boxplot that shows the question **scores**. Place the boxplots in parallel so that the plots can be compared.
   2.1. Indicate the q1 (first quartile), median and q3 (third quartile) for both python and cpp questions
   2.2. Interpret the results
3. Finally, compare the **Answered** attribute of both data sets by creating a stacked bar chart. Questions that have already been answered and that are pending should be displayed in separate bars.
   3.1. What kind of information is conveyed through this figure?