Data Visualization Concepts

BINF4234

Prof. Dr. Renato Pajarola

Exercise and Homework Completion Requirements

1. Exercises and reading assignments are **mandatory** and they must be completed successfully to finish the class and get a sufficient passing final grade.
2. Exercises are graded coarsely into categories **pass** or **fail**.
   - *A **fail** is given to failed submissions and incomplete solutions, and no points are awarded.*
   - *A **pass** indicates that the exercise is sufficiently good to receive the corresponding points.*
   - *Late submissions (up to one day) will result in "-1" point.*
3. The five exercises give rise to the following point distribution: 2 – 3 – 5 – 5.
   - *A **minimum of 7 points** from all four exercises must be achieved to pass the module. Failure to achieve this minimum will result in a failing grade for the entire module.*
   - *Thus at least two exercises have to be correctly solved, and one has to be from the more advanced ones.*
4. We give **bonus points** for students who have completed more than 8 points from all the exercises.
   - *Thus **7 points** from the exercises is required, **8 points** is still normal passing, and **9 and above** would give 1 or more extra points.*
   - *Only the bonus points can and will be added directly to the final grade.*
5. Do not copy assignments, tools to detect copying and plagiarism will be used.
   - *The exercise results are an integral part of the final course grade and therefore the handed in attempts and solutions to the exercises **must be your personal work**.*

Submission Rules

- Please hand in your solutions in a .zip archive which contains: **dvc_ex2.py**, **dvc_ex2.html** (screenshot is acceptable), and **readme.txt**.

  Name the zip folder as *dvc_ex2_MATRIKELNUMBER.zip* (e.g. dvc_ex2_01234567.zip)
- Submitted code must compile and run without errors using the indicated Python environment, using the included libraries, packages and frameworks. If additional libraries/packages are needed, please specify in your 'readme.txt' file.
- The whole project source code must be submitted before the given deadline.

- **Deadline is Thursday, 29 October 2019 at 23:59h**

Exercise 2

In this exercise, you will access, process, and visualize the second-phase covid-19 data in Switzerland. The goal of this exercise is to visualize and compare the daily cases increase for each of the cantons in Switzerland from June 2020, such that people can better assess the situation in different cantons. You will get access to the online dataset from this link, and the complete tasks are described below:

Task1: Data Preprocessing.

T1.1: Read the online data into a DataFrame and deal with NULL values as indicated in the skeleton.

T1.2: Calculate daily new cases for each canton, and smooth it with a rolling window; answer the following two questions briefly in 'readme.txt':

- Why do we need smoothing?
- How does the window size affect the result?

T1.3: Construct a ColumnDataSource for plotting according to the task descriptions in the code skeleton.

Task2: Data Visualization.

T2.1: Implement plotting functions using the ColumnDataSource as the input, draw a group of line charts on the same plot, each line corresponds to daily new cases of a canton.

T2.2: Add a clickable legend and hovering tooltips to the plot in order to provide more detailed information.

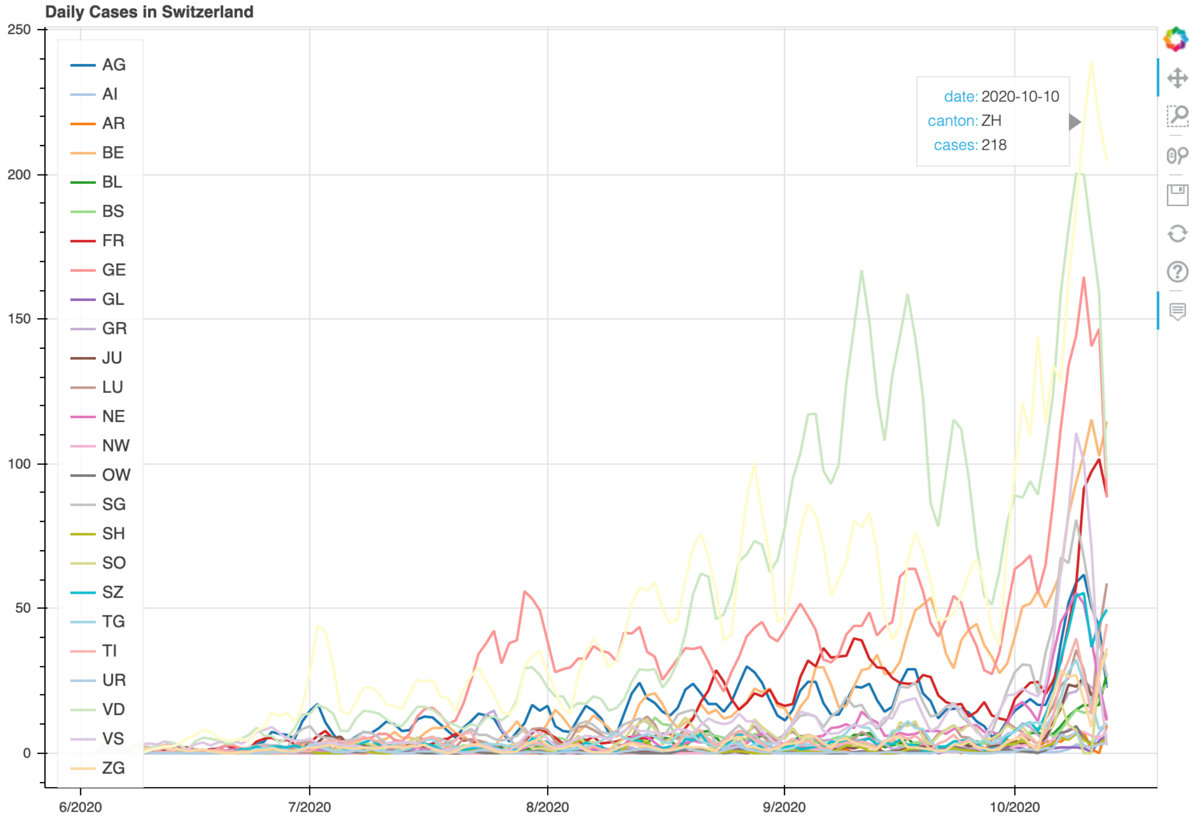
Voluntary Task3 (optional): In this exercise, the rolling window size is fixed. However, we could also add a slider control for the rolling window size, then interactively change the window size and see what's happening. Try to implement this slider by following the **Bokeh Tutorial** exercise 5. Set the range of the window size to be [1, 20].

Remarks:

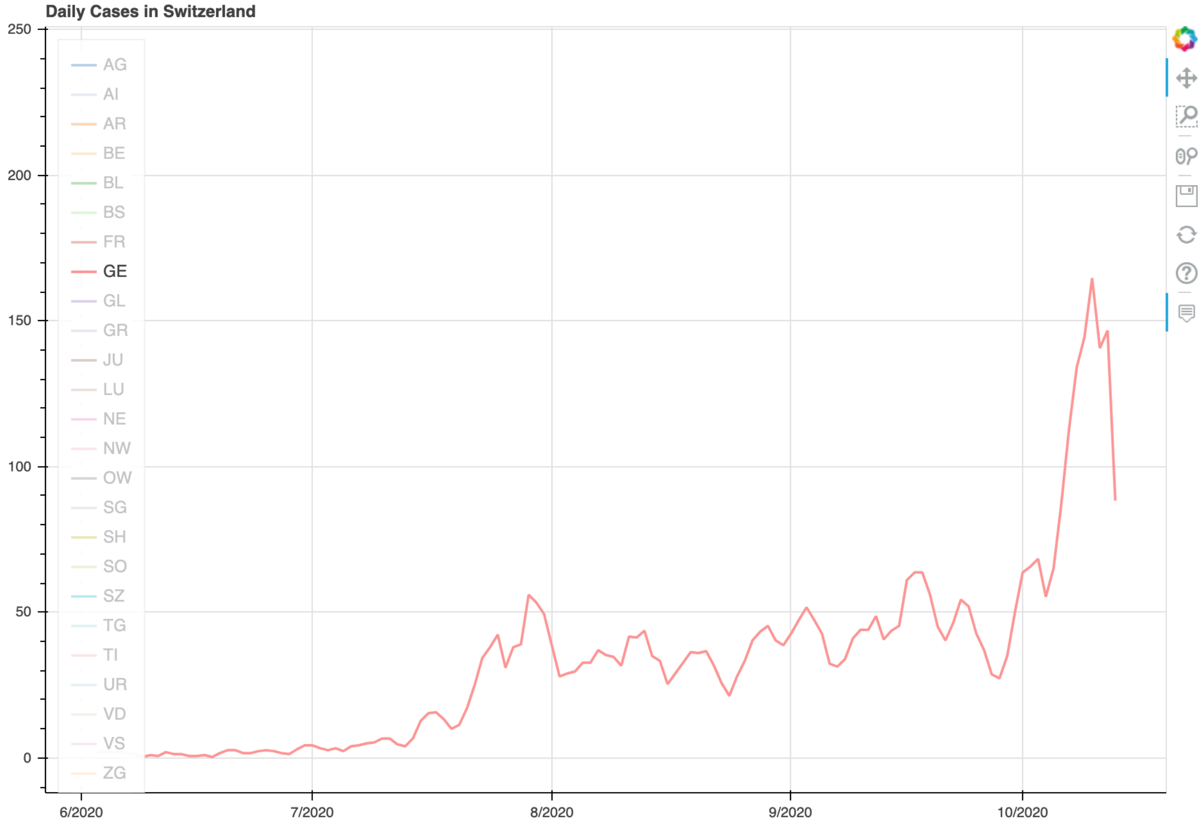
- In general, the code skeleton is well structured and divided into groups based on the tasks. However, you may want to change the structure of the skeleton for readability reasons of your own code.
- We recommend to use Jupyter Notebook for your implementation as it can visualize the intermediate output which helps for debugging. However, the final delivery of your code should be .py file rather than .ipynb.
- Try to make good use of the hints and references provided in the skeleton code. (**very important**)
- Try to google first for any Python related issues/bugs.
- Due to the special situation, we don't arrange in person meeting in this semester. Please contact the TA **Fan Feng (fan.feng@uzh.ch)** for technical questions regarding the exercise only if needed.
- More than one day late submission will not be accepted and graded.
- The deliverables of this exercise will be a clean version of your code with proper comments, any additional files necessary for executing it (for example, the data file), a "readme.txt" file for your comments or remarks (if necessary), as well as an export of the final output result in .html or .jpg/.png format. The absence of any required deliverable files will automatically lead to a **FAIL**.

The following pictures are examples for the desired (but not necessarily same) visualization result:

- Initial visualization:



- Clicking on any of the labels from the legend should result in a hiding of the corresponding line. For example, after clicking all but one labels you will only see one line as shown below.


Daily Cases in Switzerland
250
200
150
100
50
0
AG
AI
AR
BE
BL
BS
FR
GE
GL
GR
JU
LU
NE
NW
OW
SG
SH
SO
SZ
TG
TI
UR
VD
VS
ZG
6/2020
7/2020
8/2020
9/2020
10/2020