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_ex3.py**, **dvc_ex3.html** (screenshot is acceptable), and readme.txt if needed.

  Name the zip folder as *dvc_ex3_MATRIKELNUMBER.zip* (e.g. dvc_ex3_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, 19 November 2019 at 23:59h**

Exercise 3

In this exercise, we will focus on visualizing covid-19 tests statistics in Switzerland. The goal of this exercise is to display and link daily total tests numbers together with positive cases numbers in two plots, so that when we dragging the shaded overlay in the second plot, the range of the first plot will be automatically updated. You will get access to the online dataset from this link, (optionally, you can also use the provided same data named “covid19_tests_switzerland_bag.csv") and the complete tasks are described below:

Task1: Data Preprocessing.

T1.1: Read the online data into a DataFrame using pandas.

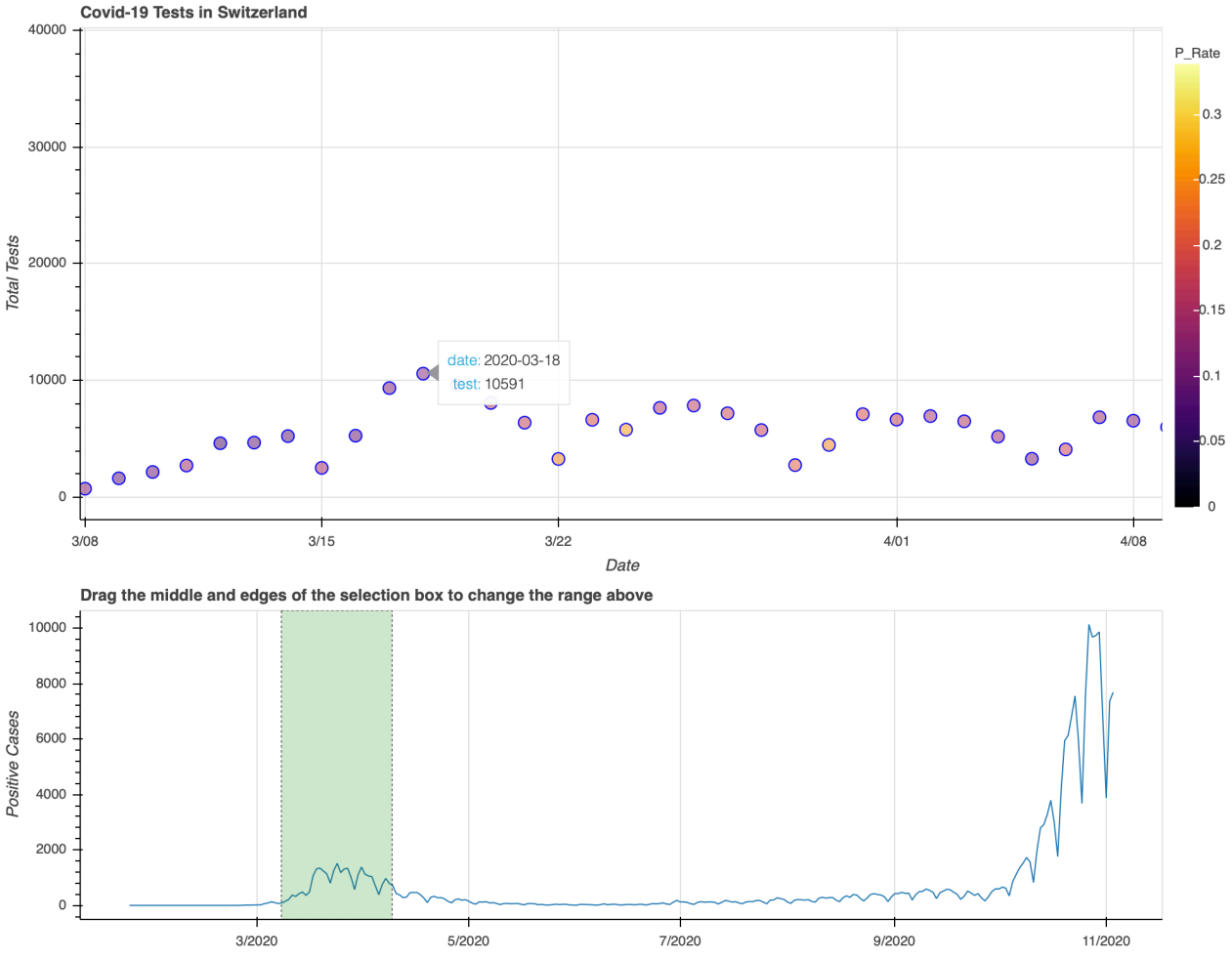
T1.2: Construct a ColumnDataSource for plotting according to the task descriptions in the code skeleton. (Be attention that the original data from the link has an error, the last column should be **frac_positive** rather than **frac_negative**, the provided data has already correct it.)

T1.3: Define a colormap and map the range of positive rate to a colormap linearly.

Task2: Data Visualization.

T2.1: Covid-19 Total Tests Scatter Plot. In this first plot, you need to display the daily tests numbers in a scatter plot, and add a hover tool to display the date and the corresponding total test number. In the scatter plot, point size is fixed, point color is mapped to the positive rate from the data.

T2.2: Create a ColorBar using the colormap in T1.3 and attach this bar to the first plot.



T2.3: Covid-19 Positive Number Plot. The second plot consists of a basic line glyph that displays the general trend of positive cases, and add a RangeTool, with which we can observe detailed scatter points distribution in the first plot. Add a hover tool to display the date and the corresponding positive number.

T2.4: Arrange two plots in one layout appropriately.

Voluntary Task3 (optional): In this exercise, the positive rate values are encoded by color, you can also, however, represent positive rate values by the point sizes. In this case, the colormap is useless and you should define a different legend which maps to the size of the scatters. Your submission will be equally treated with these two different encodings.

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**.