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 bonus 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

- 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 a 'readme.txt' file together with your submission.
- The whole project source code must be zipped and submitted before the given deadline, including the output results (saved in .html file or as a screenshot picture).
- Submit your .zip archive named *dvc_ex1_MATRIKELNUMBER.zip* (e.g. dvc_ex1_01234567.zip) through the OLAT course page.

Deadline is Sunday, 10 April 2022 at 23:59h

Exercise 2

In this exercise, you will process and visualize the taxi trip details in New York City in every day for each month. Key points are:

- Length of the trip will be encoded using horizontal bar charts, and each bar starts from the pickup time and ends at the drop off time.
- Number of passengers in each trip will be encoded to the size of the circles.
- Vendor_id should be encoded to colors.
- Each time one-month data will be displayed. User should be able to switch between months via a Select Widget.

Task1: Data Preprocessing.

T1.1: Read the .csv file into a DataFrame using pandas, convert the pickup datetime and dropoff datetime into suitable formats, and assign the color information to each trip.

T1.2: Create a selector widget for month selection.

Task2: Construct data structures.

T2.1: Create a function to prepare the data source for plotting: follow the task descriptions in the code skeleton and generate a ColumnDataSource which contains all the information you need for plotting.

T2.2: Create a function to update the data source.

(Make sure that you have an idea of how the data source should be before plotting. You may want to read the reference link in the skeleton first before starting.)

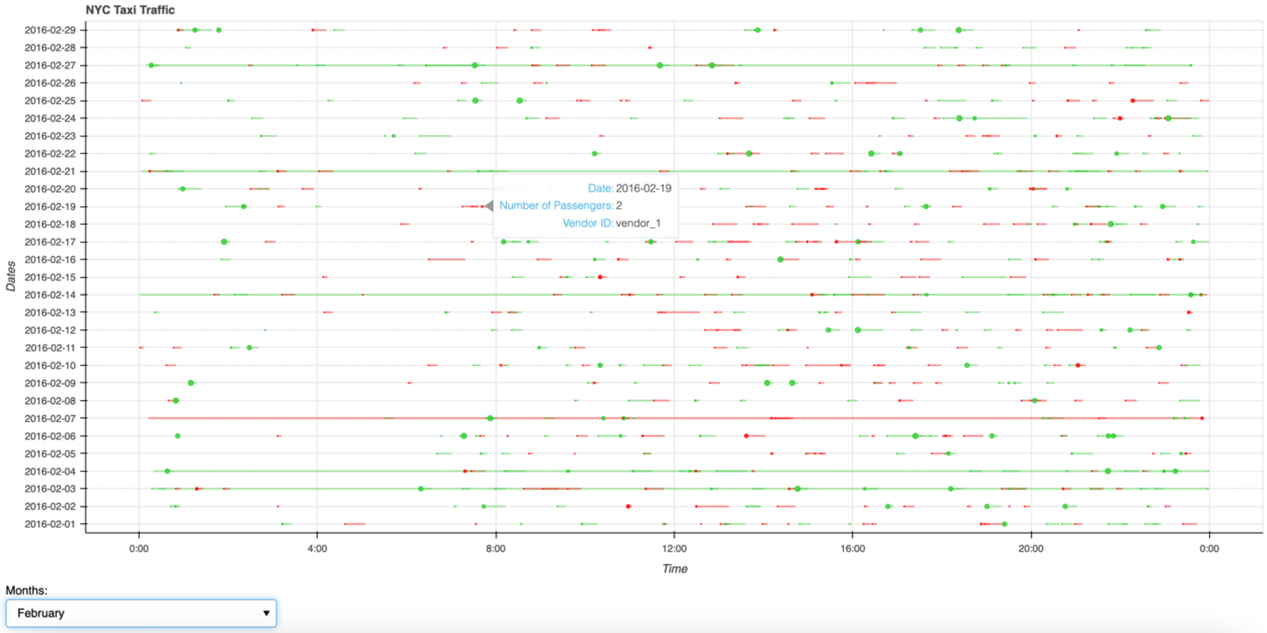
Task3: Data Visualization.

T3.1: Implement the plotting function using the ColumnDataSource as the input, and draw a horizontal bar chart and circle glyphs, and try different styling attributes.

T3.2: Add hovering tooltips to the plot in order to provide more detailed information. To be specific, the hover tooltips should display “**vendor_id**”, “**number of passengers**”, and “**Date**” when hovering on the circles (attention, not on the bars, but on the circles).

T3.3: Save the plot as a .html file.

The following picture is an example for the desired (but not necessarily the same) visualization result:



Months: February

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 **Emine Didem Durukan (eminedidem.durukan@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), the dataset (.csv 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**.