

Data Visualization Concepts



Prof. Dr. Renato Pajarola

Exercise and Homework Completion Requirements

1. Exercises and reading assignments are **mandatory**. 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 a reduction of one point. Submissions after more than one day will not be accepted or graded.*
3. The four exercises give ascending points in the following 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** to students who have completed more than 8 points from all the exercises.
 - Thus **7 points** from the exercises are required, **8 points** are still a normal pass, 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, therefore the handed-in solutions to the exercises **must be your personal work**.

Submission Rules

- *The deadline for submitting Exercise 2 is Sunday, 16 April 2023 at 23:59h.*
- Please submit your solution via the OLAT in a .zip archive '**dvc_ex2_MATRIKELNUMBER.zip**', e.g. '**dvc_ex2_01234567.zip**'.
- The archive should contain the code '**dvc_ex2.py**', and the output '**dvc_ex2.html**' (or screenshot in .jpg/.png format). If additional dependencies are used other than the ones in the environment defined in the tutorials, please specify them in a '**readme.txt**' file.
- The code should run without errors and generate the expected output.

Exercise 2

In this exercise, you will show the stock data of one of Meta, Apple, Google, Microsoft, and Amazon in a candlestick chart similar to [this example](#). You will add an overlay of two financial metrics, 'EPS Growth' and 'PE Ratio', which investors refer to when evaluating the company's stock. If interested, you may go to [this page](#) for more details. The URLs to the stock and metrics data are provided in the code skeleton.

You'll also add some interaction to the chart like panning and zooming, and allow users to click on the interactive legend to hide or show a metric. Optionally, you can add a range selection plot to zoom in on one region in the chart. Please read the comments in the code skeleton for details of the tasks, and follow the reference links for more instructions.

Task1: Prepare the Data

- 1.1: Convert the data type of time columns to `datetime`.

Task2: Create A Candlestick Chart from the Stock Data

- 2.1: Create the data source and set the basic properties of the figure.
- 2.2: Manually set the y-axis range.
- 2.3: Use `CDSView` to create two filters on the stock data.
- 2.4: Draw the glyphs in the candlesticks (`segment` and `vbar`).
- 2.5: Add volume bars in the candlestick chart.
- 2.6: Add a hover tool for the candlesticks.

Task3: Add Metrics Plot to the Candlestick Chart

- 3.1: Set the y-axes for the metrics.
- 3.2: Use scatter and line glyphs to plot the metrics.
- 3.3: Make the legend of metrics interactive.
- 3.4: Add a hover tool for the scatter glyphs.

Task4 (optional): Add a Range Selection Plot to Zoom in the Candlestick Chart

Save the bar chart(s) to a .html file with the filename 'dvc_ex2.html'.

An example of the expected (but not necessarily the same) output:



Remarks:

- The code skeleton is structured into sections corresponding to the tasks. You are free to change the structure of the skeleton for your own needs and preferences.
- We recommend using Jupyter Notebook for your initial implementation. It can show the intermediate outputs, which is convenient for testing and debugging. Please note that the final submission of your code should be a .py file.
- The comments and references in the skeleton code are **very important** for completing the tasks.
- Try to Google first for any errors. Chances are good that someone else has solved the problem.
- If Google cannot help, please use the OLAT forum to post technical questions regarding the exercise.
- The Q&A session for Exercise 2 will be on **Monday, 03 April 2023, from 17:00 to 18:00**. The place will be announced in time.