Data Visualization Analysis

BINF4245

Prof. Dr. Renato Pajarola

Exercise and Homework Completion Requirements

1. Every **solved** exercise will earn you points, up to 15 in total.
    - *Late submissions (up to one day) will result in "-1" point.*
    - If the solution works as expected, full points are awarded, else partial points are assessed as outlined in the code skeleton.
2. The four 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 must be correctly solved, and one must be from the more advanced ones.*
3. 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 exam.*
4. 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 run without errors using the indicated Python environment, using the included libraries, packages, and frameworks.
- 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 *dva_ex1_Firstname_Lastname_MATRIKELNUMBER.zip* (e.g. dva_ex1_Hans_Muster_01_234_567.zip) through the OLAT course page.
- You zip file should contain:
    - Your working code as dva_ex1_Hans_Muster.py file **(stick to the naming, no jupyter files)**
    - A screenshot or export of your tool as .jpg, .png, .pdf or .html

- An readme.txt if you needed additional libraries and good reasons why you needed them.

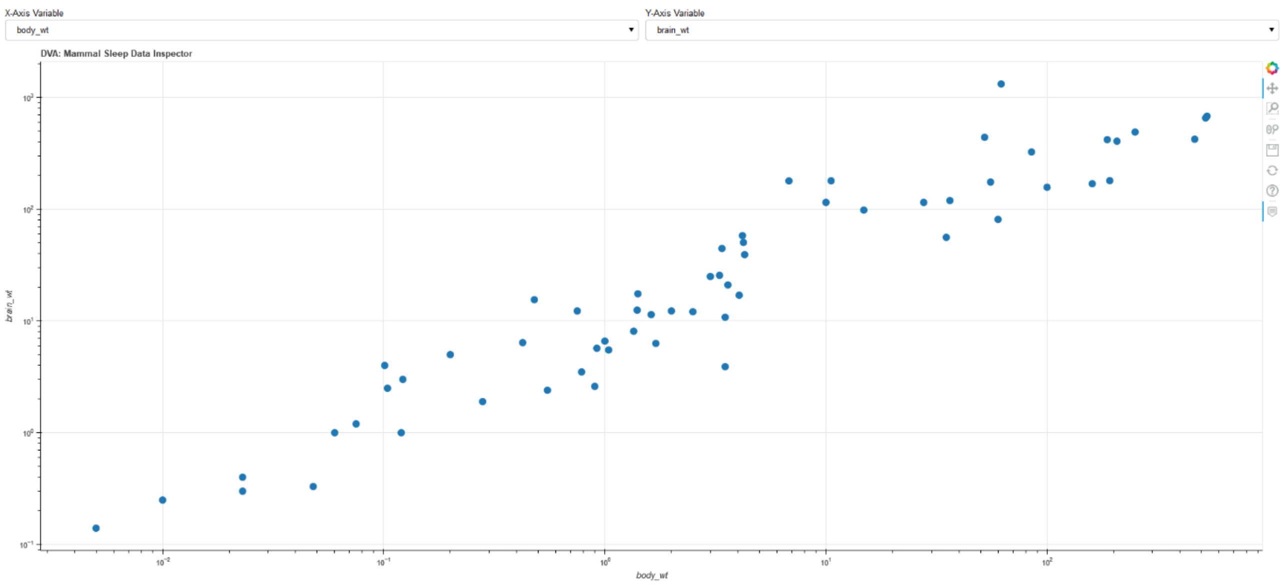
- **Deadline is Sunday, 27 March 2022 at 23:59h**

Exercise 1

In this exercise, the goal is to get familiar with Bokeh and Pandas by implementing a small "data plotting" tool using Bokeh Server Applications. The tool will allow us to select two columns of our dataset and plot the corresponding data along the x and y axes of a scatter plot. To achieve this, you will first have to do some data cleaning, followed by implementing the just described functionality using the Bokeh framework.

The final application will include two Bokeh Select widgets to choose between the different columns of the dataset, and a central scatter plot, which must update, whenever the "Select" widgets are changed.

To achieve this, you should go through the code skeleton in dva_2022_ex_1_firstname_lastname.py and fill out the code blocks marked with TODO according to the respective task. Have a look at the **demo.mp4** video to have a reference on how the final tool should look and behave.



Tips:

- If you haven't done the Bokeh tutorial yet, I highly recommend you do so, here or have a look at the "Server App" applications in the Bokeh gallery.
- Whenever you don't know something, try Google, check the Bokeh docs or the Bokeh discourse channel.
- When you change the data of your ColumnDataSource, always change the entire .data attribute, not just a subset. Also, never swap the entire ColumnDataSource:

  Don't:
  ```
  source.data['xs'] = [1,2,3,4,5,6]
  source = ColumnDataSource(dict(xs=[1,2,3,4,5,6], ys=[6,4,5,3,1,2]))
  ```

  Do this:
  ```
  source.data = dict(xs=[1,2,3,4,5,6], ys=[6,4,5,3,1,2])
  ```
- A current limitation of Bokeh is, that one cannot change the type of the axis, e.g., from "log" to "linear", on the fly. Hence keep the axes logarithmically, even though the "predation", "danger" and "exposure" columns are of ordinal nature.

Remarks:

- In general, the code skeleton is well structured and divided into sections, the partial points assigned to each subtask are noted directly in the code.
- 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 me (Gaudenz Halter, halter@ifi.uzh.ch) for technical questions regarding the exercise only as a last resort.
- More than one day late submission will not be accepted and graded.
- Online office hour will be on Friday, 18 March 16:00 – 17:00. The zoom link will be communicated in that week, prior to the office hour, via email and the OLAT forum.