
LAB 01: DATA VISUALIZATION WITH PYTHON

Revision History

Revision	Date	Author(s)	Description
1.0	July 03, 2023	LN Nam	Initial, release first version in Vietnamese.
2.0	Feb 24, 2024	LN Nam	Modified, change to English.

Contents






1	Learning Objectives	2
2	Notes and Constraints	2
3	Problem Statement	2
3.1	Data collection	2
3.2	Data exploration	2
3.3	Exploring hidden relationships in your dataset	3
4	Limitations	3
5	Evaluation Criteria	4
6	What to Submit	4

1 Learning Objectives

This lab aims to equip students with the foundational skills necessary for effective data visualization using Python. By learning and leveraging Python libraries such as Matplotlib, Seaborn, and Pandas for creating a wide range of static, interactive, and animated visualizations, the students explore the hidden relationship in your focusing dataset in a specific domain.

2 Notes and Constraints

List of constraints when doing this lab:

-  Work without a report will not be graded.
-  Members who do not contribute to the project will not receive points.
-  Reference sources (if any) need to be fully recorded in the report in the References section. Note that it is necessary to distinguish between referencing and plagiarism.
-  Individuals or groups that commit cheating and dishonesty will receive 0 points in the course.
-  Name the assignment category MSSV1_MSSV2_MSSV03_MSSV04_MSSV05_Lab01, with MSSV being the student number, compress the entire submission into 1 file before submitting. If the size is > 20MB, upload it to an external storage service such as Google Drive or OneDrive, and then submit the link. Last but not least, please keep the link public for at least 2 years.

3 Problem Statement

In this lab, you use Kaggle to search for publically available data on a topic of interest to the student group. The data set must be organised in a table with at least five data columns and 1000 rows.

Note: Students need to select new data sets within the last 5 years.

3.1 Data collection

The student group needs to present the context and motivation for the choices made on the selected data set.

- The mean context or main story that makes your group choose this topic to find suitable data for this project.
- The main topic of your dataset is the sources for its construction.
- How do people construct this data? What're methods?
- How do we use this dataset? Is this legal for use in education?

3.2 Data exploration

Almost like your previous work on the course of Introduction to Data Science, and Programming for Data Science, you have to pre-processing and explore your dataset

- What does each line mean? Does it matter if the lines have different meanings?
- What does each column mean?
- What data type does each column currently have? Is there any column whose data type is not suitable for further processing?

- For each column, how are the values (numeric, categorical) distributed?
- Is there a need to pre-process the data and if so, how do you need to do it?

3.3 Exploring hidden relationships in your dataset

After preprocessing your dataset and conducting exploratory data analysis to gain a deeper understanding, the next step involves utilizing various charts to visually represent the data's attributes. This process not only aids in visualizing the attributes but also in drawing meaningful conclusions and making informed decisions based on the insights derived from these visual representations.

- Begin by identifying and selecting key attributes from your exploratory data analysis that warrant visual representation. Choose from the array of chart types you've learned—be it line charts, bar charts, pie charts, or others—to best showcase these attributes.
- It's crucial to justify your choice of chart for each attribute, explaining why a particular chart type is most suitable. While it's permissible to use multiple chart types for a single attribute, you must provide a clear rationale for doing so.
- Aim to present the relationships within your data in a structured manner, progressing from simple to more complex visualizations. Start with visualizations focusing on individual attributes and gradually move towards depicting intricate relationships among multiple attributes.
- Explore the possibility of cause-and-effect relationships within your data. For instance, when analyzing COVID-19 data, investigate if there's a discernible link between rising infection rates and mortality, and how such relationships can be effectively demonstrated through visualizations.
- While it's not necessary to visualize every possible relationship within your data, strive to cover a broad spectrum of graph types that have been introduced in your coursework. Focus on creating as many relevant and insightful visualizations as possible.
- You're encouraged to experiment with engaging and innovative graph types that not only enhance the visual appeal but also facilitate a deeper understanding of the data, revealing valuable insights.

4 Limitations

For this lab, there are specific constraints that we ask you to adhere to closely:

- The exercises are designed to be completed within a basic Python programming environment. We encourage the use of straightforward tools and request that you refrain from utilizing advanced software solutions such as Tableau for data visualization purposes.
- You are welcome to employ foundational libraries such as NumPy, Pandas, Seaborn, and Matplotlib for your tasks. Should you wish to explore additional libraries, please consult with your instructors beforehand to gain their approval.
- While incorporating simple machine learning algorithms can offer deeper insights into your data, please note that this is an optional component of the lab and not a mandatory requirement.

5 Evaluation Criteria

Your assignment will be evaluated based on the following criteria:

Criteria	Mark
Data collection & pre-process data	5%
Select, interpret, and visualize fields and their hidden relationships.	50%
Derive logical meaning behind each visualized data.	20%
Consider many relationships and many different perspectives.	10%
The report presents a logical and clear layout and format.	15%
There is analysis, visualization with novel charts and drawing of useful information. Use basic machine learning models.	5%
Overall comprehension of the submitted source code.	5%
Total	110%

6 What to Submit

You must submit:

- ❑ **Docs Folder:** This folder should contain your report files in .doc, .docx, or .pdf formats, with a strong recommendation for .pdf to ensure compatibility and preservation of formatting. Your report should cover several key areas:
 - **Group Information:** Include your group name and student IDs.
 - **Requirement Fulfillment:** Discuss how fully each project requirement has been met.
 - **Member Contributions:** Detail the level of contribution from each group member.
 - **Algorithm and Implementation:** Provide thorough explanations of the algorithms used, include running examples, and offer commentary on the code.
 - **Presentation Style:** Aim for clarity in your report, using illustrations where helpful to convey your points effectively.
- ❑ **Source_Codes Folder:** This directory should house all the source code for your project. It will primarily contain Jupyter notebooks and Python scripts developed by your team. If you have code in languages other than Python, please include clear instructions for its use.
- ❑ **Datasets Folder:** Here, you should provide either the direct dataset or, if the dataset is too large, a link to where the dataset can be accessed. For large datasets, leveraging cloud storage solutions like OneDrive or Google Drive is encouraged to facilitate easy access and sharing.

By organizing your submission in this manner, you'll help ensure that your work is clearly presented, easily navigable, and thoroughly documented, reflecting the depth and breadth of your project efforts.