Project #2 Guidelines:
Exploratory Data Analysis and Visualization

Find below the general guidelines. Please check in with your instructor to confirm their specific requirements (if they have any). Good luck!

Introduction

The goal of this project is to perform a descriptive statistical analysis, gain insights and create an interactive dashboard that tells a compelling story with visualizations, allowing for decision-making.

For this project, you have the option to work either individually or in pairs, depending on your interests and goals. If you and a partner share a common interest and have ambitious plans for what you want to achieve, working in pairs is a great way to collaborate and tackle more complex challenges. However, please note that if you choose to work in pairs, we expect a higher level of output and quality.

If you're struggling to come up with a topic, we have provided a list of datasets for you to consider. However, we highly recommend that you explore and select a topic and dataset that personally interests you, as this will make the project more engaging and rewarding.

Prerequisites

In order to successfully complete the upcoming project, you should possess a strong understanding of several key concepts, including Python programming, data wrangling, exploratory data analysis (EDA), and visualization tools such as Tableau or Power BI. The following are essential prerequisites that you should have before beginning the project:

- Proficiency in basic **Python** programming, including knowledge of **data wrangling and data cleaning** techniques in Python. For more details, refer to the prerequisites for Project 1.
- Proficiency in **EDA** and descriptive statistics, including an understanding of different types of data and their properties, how to transform between data types, and how to perform analysis based on the data type. This includes numerical and graphical techniques for univariate analysis, as well as bivariate and multivariate analysis to identify and analyze relationships between pairs or sets of numerical and categorical variables.

    - Thorough **univariate analysis**, employing measures such as frequency tables, centrality measures, and dispersion measures, as well as graphical methods such as bar charts, histograms, and box plots
    - Thorough **bivariate and multivariate analysis**, employing techniques such as contingency tables, chi-square goodness of fit, correlation coefficients (Cramers V, Pearson, Spearman, or Kendall's tau), and a range of graphical methods such as scatterplots, correlation maps, QQ plots, side-by-side boxplots, grouped histograms, and grouped bar charts.
    - Familiarity with basic **Matplotlib** and **Seaborn** for graphical analysis with Python.
    - Knowledge of how to detect outliers using scatterplots, box plots, Tukey's method, etc., and how to handle them.
- Basic knowledge of **probability** and **inferential statistics** to conduct tests used in EDA, such as chi-square tests, normality tests, and to interpret p-values in the context of data analytics. This includes an understanding of probability distributions, such as the normal distribution.
- Familiarity with visualization tools such as **Tableau** or **Power BI** in order to create interactive dashboards for decision making.
- Understanding of **storytelling** techniques for effectively communicating insights derived from data analysis.

Nice to have:

- Experience with Args/kwargs, Pickle, and regex can enhance advanced Python skills such as creating flexible, readable code, serializing/deserializing objects, and working with text data.
- Plotly for creating interactive visualizations and dashboards in Python.
- Advanced Matplotlib and Seaborn for more complex graphical analysis and visualization in Python.
- Streamlit for building and deploying interactive web applications for data analysis and visualization.
- SQL to store raw and clean data in a database.
- APIs and Web Scraping skills

Suggested ways to get started

1. Select a business problem and formulate one or more hypotheses that will guide your data analysis, allowing you to draw meaningful conclusions. Locate relevant data sources using methods such as web scraping, APIs, databases, or files. Consider merging multiple datasets to augment your analysis with additional insights.
2. Examine the data, understand what the fields mean and use exploratory data analysis to explore the data and identify any issues that need to be addressed.

3. Do data cleaning and data wrangling to prepare the dataset for analysis. Remember to look into missing data, outliers, data types, feature selection, converting qualitative data to quantitative etc.
4. Analyze the data, including numerical and graphical techniques (univariate, bivariate and multivariate analysis) to reveal patterns, relationships, and trends that may not be apparent from the raw data.
5. Create a dashboard with key KPIs and insights that tell a compelling story about the data, while allowing for decision-making.
6. Create a visually appealing presentation with minimal text to showcase that effectively communicates your insights and conclusions to stakeholders, building a compelling narrative that highlights the significance of your analysis.

Remember, the goal of this project is to showcase your skills in data analysis, visualization and extraction of insights that are meaningful and actionable.

Guidelines to follow

- **Version control:** commit early and often, as you can always roll back to a previous version if needed. Don't be afraid of making mistakes.
- **Organizing code**: to improve the organization and readability of your code, create separate .py files for related functions, and use multiple Jupyter notebooks if necessary. Use a "main cleaning function" in *cleaning.py* (or similar) that calls all the smaller cleaning functions in a specific order to perform the entire cleaning process at once.
- **Agile methodology**: participate in Agile ceremonies such as daily standups and a final retrospective, and optionally use a Kanban board to stay organized throughout the project.

Deliverables

You must submit the following deliverables in order for the project to be deemed complete:

- A new repo on your github account.
    - A **working code** that **meets all technical requirements**, **built by you**.
        - At least 1 jupyter notebook is required containing your Python code for data cleaning and preprocessing steps, descriptive statistical analysis, and visualizations. You will also provide a brief summary of the insights gained from the analysis and visualizations, along with recommendations for further analysis or actions based on the results.
        - Include your functions in .py files
    - Tableau or PowerBI report
    - Additional needed files for your work
    - A **README with the completed project documentation**.

    - The URL of the **slides for your project presentation**.
- **Presentation:** when presenting your work, there are many important factors to consider, such as the content of your presentation and the way you deliver it. The following link offers valuable advice on how to make a strong presentation: Presentations.
- Paste your own repo's link in the Student Portal Project Activity

Rubrics

In order to assess your project and ensure all requirements are met, a **rubric** will be used. This rubric is used to **evaluate your project** by your teaching staff but also to **communicate** what constitutes incomplete, acceptable and excellent performance across each of the learning outcomes for the project. Take some time to review the rubric here and ask your lead teacher or TA any questions about it if necessary.

Optional Advanced Features

While completing the basic requirements of your project is a great start, taking advantage of some advanced features can really take your work to the next level. Here are some options to consider if you want to go above and beyond:

1. Data gathering and integration: use APIs and web scraping to gather data from different sources. Combine and integrate data from multiple sources, including different databases, APIs, or file formats.
2. Use advanced data cleaning techniques, when imputing missing values or handling duplicates (such as using fuzzy matching), in addition to the basic techniques.
3. Database creation: create a database to store your raw and your clean data for analysis.
4. Correlation and statistical analysis: use hypothesis testing concepts to interpret your correlation analysis and to validate your findings.
5. Improve your code by using error handling techniques, applying OOP principles for modularity and reusability, and utilizing regular expressions to extract insights from textual data.
6. Do advanced visualizations using interactive libraries such as Plotly.
7. Deploy your dashboard to the web using a tool like Flask or Django to make it accessible to others and share your insights more widely.
8. Use a kanban board to organize and manage project tasks.
9. Anything outside of the box that can improve your analysis!

** If working in pairs, it is recommended to consider completing some of the optional advanced features to further challenge and enhance your skills in the project.*