Project #1 Guidelines:
Data Cleaning and Data Wrangling - Shark Attacks dataset

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 provide you with hands-on experience in cleaning, wrangling, and transforming a real-world dataset using Python. By the end of this project, you should be able to demonstrate your ability to apply Python programming concepts to perform data cleaning, wrangling, and manipulation tasks.

For this project, you will be working in groups with a messy dataset called *"Shark Attacks"* and use your data wrangling skills to clean it up and prepare it for analysis.

You can find the dataset and information about the data here.

Prerequisites

To successfully complete the upcoming project, it's important to have a strong understanding of Python programming and data manipulation techniques. The following prerequisites are essential skills that you should have prior to starting the project:

- Proficiency in basic **Python** programming, including data types, operators, data structures, flow control, loops, functions, creating and importing modules, list comprehension, and lambda functions.
- Familiarity with the **Pandas** library, including Pandas Series, Pandas Dataframes, and methods such as Map, Apply, and Mapapply.
- Understanding of **Data Wrangling and Data Cleaning** techniques in Python, including:
    - Data cleaning methods for handling missing values, duplicates, outliers, and errors in data.
    - Data transformation techniques for formatting, filtering, and slicing data.
    - Data organization methods for ordering and aggregating data.
- **Basic** knowledge of **Exploratory Data Analysis (EDA)**, including statistical analysis and visualization techniques such as quantitative vs qualitative

variables, measures of central tendency and dispersion, Pearson correlation, and basic charts like histograms, box plots, barplots, and scatter plots. This is just at an introductory level to help you gain insights from data. Advanced EDA will be taught later on the course.

Nice to have:

- Args and kwargs
- Pickle
- Data structuring: pivoting, melting, reshaping
- Combining data: merging, concatenating, joining
- Regex

Suggested Ways to get started

1. Download the dataset and import it into Python.
2. Examine the data and try to understand what the fields mean before proceeding with data cleaning and manipulation.
3. Use quick tips on exploratory data analysis, such as the "describe" method and basic graphs, to explore the data and identify any issues that need to be addressed.
4. Decide on a hypothesis (or hypotheses) to guide your cleaning efforts. Present the analysis clearly and coherently to support the findings. For example: "Sharks attack more younger people than older," "The easiest way to get attacked by a shark is by surfing," "Sharks attack more people in the USA than in Iceland," etc.
5. Apply at least five data cleaning techniques, such as handling null values, dropping columns, removing duplicated data, manipulating strings, formatting the data, etc., to prepare the dataset for analysis.
6. Once the data is cleaned, analyze it to validate your hypotheses and draw conclusions about the data. Use basic statistical analysis and create graphs to support your findings.
7. 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.

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 with the name data-cleaning-pandas on your github account.
    - A **working code** that **meets all technical requirements**, **built by you**.
        - At least 1 jupyter notebook is required
        - Include your functions in .py files
    - 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. Use advanced data cleaning techniques, when imputing missing values or handling duplicates (such as using fuzzy matching), in addition to the basic techniques.
2. Utilize more sophisticated data visualization libraries, such as Seaborn or Plotly, to create interactive and informative visualizations that enhance your analysis. For example, an interactive map of shark attacks.
3. Apply regular expressions to extract insights from textual data in the dataset.
4. Explore additional questions or hypotheses beyond the minimum requirements by combining the Shark Attack dataset with other datasets, such as weather or oceanographic data, to create a more comprehensive analysis that can uncover additional insights.
5. Anything outside of the box that can improve your analysis!