Drexel University

College of Computing and Informatics

INFO 212: Data Science Programming I

Assignment 1

Due Date: Sunday, Oct, 22, 2023

This assignment counts for 10% of the final grade

A. Assignment Overview

This assignment provides the opportunity for you to practice with Python data analysis basics.

B. What to Hand In

Sumbit a completed this Jupyter notebook.

C. How to Hand In

Submit your Jupyter notebook file through the course website in the Blackboard Learn system.

D. When to Hand In

1. Submit your assignment no later than 11:59pm in the due date.
2. There will be a 10% (absolute value) deduction for each day of lateness, to a maximum of 3 days; assignments will not be accepted beyond that point. Missing work will earn a zero grade.

E. Written Presentation Requirements (if applicable)

Images must be clear and legible. Assignments will be judged on the basis of visual appearance, grammatical correctness, and quality of writing, as well as their contents. Please make sure that the text of your assignments is well-structured, using paragraphs, full sentences, and other features of well-written presentation. Text font size should be either 11 or 12 points.

F. Marking Schemes:

Marking assignments will be based on several aspects: presentation, correctness and coding styles.

The following is a set of guidelines for the presentation and coding style in this course:

- 1. Write good comments. Marks will be dedecuted if there are no comments.
- 2. Your comments must mention the purpose of each parameter, and must be grammatically correct.
- 3. When breaking up a long line, break it before an operator, not after.
- 4. Present your code and answers in well-structured format.

G. Answer the following questions.

STUDENT NAME:

Question 1 [60 marks]: Data Gathering and Dive. Do not write any code in your answers to the following questions. You are only allowed to express your thoughts in plain text.

Data.gov is the United States government's open data website. It provides access to datasets published by agencies across the federal government. Navigate to data.gov and Explore the various categories and choose a dataset that you find intriguing. Ensure that the dataset you select has a substantial amount of data for analysis (preferably more than 500 records).

1. Provide the name of the dataset. Include the direct link to the dataset's page on data.gov.

2. Familiarize yourself with the dataset. Understand its structure, the types of variables it contains, and its overall context. Describe the following metadata about the dataset: The source agency or organization that provided the dataset. The date the dataset was last updated. The number of records and variables (columns). The purpose or context of collecting this data.

3. List the names of the columns/attributes of the data set. Describe the meaning of each column/attribute including its data type and example value.

4. Download the dataset from data.gov. If the dataset is large, you may choose to work with a subset of it. Open the dataset in a text editor, Excel, or any applications you can use. Show first 3 records from the dataset.

5. Given the dataset of your choice, craft three meaningful analytical questions. Each of your questions should necessitate a combination of extraction, aggregation, comparison, and visualization steps. Describe each question and analytical operations in detail. For each question, make sure you describe:
    - Extraction: Identify which parts of the data are essential and any preprocessing or wrangling steps required.
    - Aggregation: Highlight if any summarization or grouping operations are needed.
    - Comparison: Mention if your analysis requires comparing different data segments, times, or groups.
    - Visualization: Suggest a suitable visualization technique or chart type that would effectively communicate the results of your analysis.

    Examples: If your dataset was about global climate data over the last century:

    - Your question might be: "How has the average temperature in major coastal cities changed over the decades, and how does this compare to cities situated inland?"

    The analytical operations are:

- Extraction: From columns xxx, xxx, ... select temperature data for major coastal cities and major inland cities.
- Aggregation: Calculate the average temperature for each decade. The calculation applies to columns xxx, xxx, ...
- Comparison: Compare the temperature trends between coastal and inland cities. The comparison is between xxx, xxx, ...
- Visualization: Plot the columns xxx, xxx, ..., to generate a line chart showcasing temperature trends over decades for both city types.

Question 2 [20 marks]:

Download the zipped file "inflammation-data.zip" and unzip it in a folder. You should see a directory "inflammation-data" containing data files. Load the data in the file 'inflammation-05.csv' to an Numpy array.

1. Suppose the rows represent patients, and columns represent days. How many patients and how many days does the file contain?

2. Extract and display the values of the last 10 patients' inflammation in last 10 days.

3. For each patient, compute the average inflammation over all days. You should compute all patients' average inflammation values in single line. The result is an array.

4. Plot all patients' averages and label the plot with "Average Inflammation of Patients".

5. For each day, compute the average inflammation over all patients. You should compute all the daily average inflammation values in a single line. The result is an array.

6. Plot all daily averages and label the plot with "Average Inflammation of Days"

Qestion 3 [20 marks]:

Continue with the 'inflammation-data' directory. Given the following list of file names:

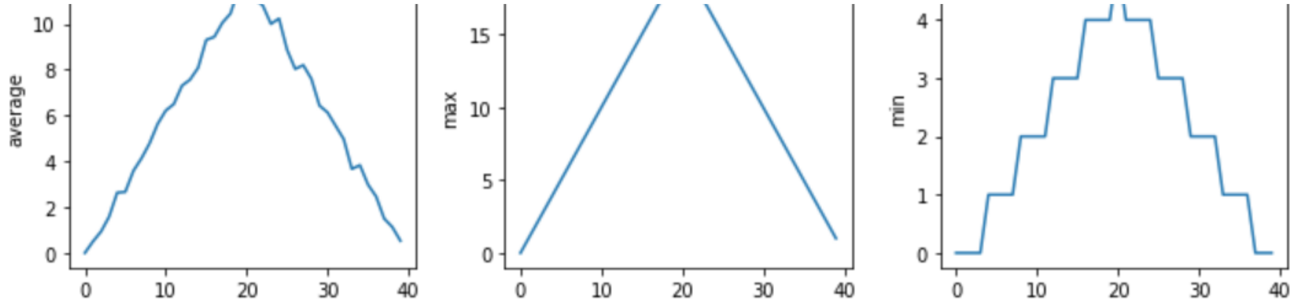
```
filenames = ['inflammation-02.csv', 'inflammation-04.csv', 'inflammation-06.csv'],
```

write a program to plot the daily averages, maximums, and minimums of inflammation for each of the file. Your plot should have three rows each of which shows side-by-side plots of average, max, and min. The following figure illustrates the expected outcome:

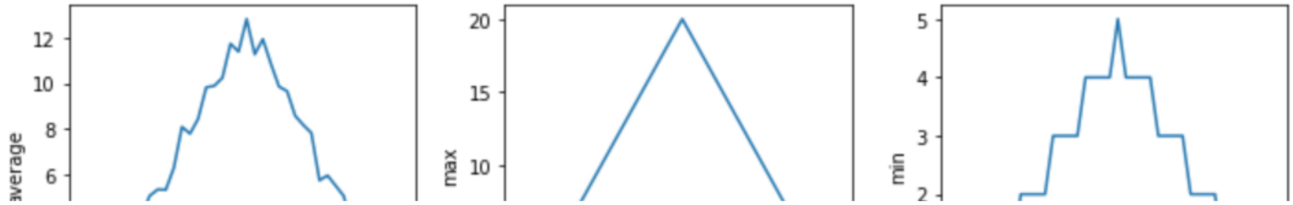
inflammation-02.csv

```
# Your code here

```



inflammation-04.csv


average
max
min