DASC 5300/CSE 5300
Foundations of Computing
Instructor: Sharma Chakravarthy
Project I: Python Programming and Data Analysis

Made available on:	**08/30/2022**
Complete Project Due on	**09/22/2022 (by 11:59 pm)**
Submit to	**Canvas (uta.instructure.com)**
	1 zipped folder containing all the files/sub-folders
	Late submissions have a penalty as indicated!
Weight:	**15% of total**
Total Points:	**100**

In this project, I would like you to familiarize and get hands-on experience with Python by analyzing the NY **Motor Vehicle Collisions** data. The Motor Vehicle Collisions table contains details on each vehicle involved in the crash. Each row represents a motor vehicle involved in a crash. The approach is to perform some simple analysis on small data sets (taking random samples is the best way) first to get the hang of analysis. I do realize that some of you are very new to Python; hence, I have chosen simple problems to exercise different features and packages of Python. In order to get you up to speed, this project will have sub-problems to solve as described below. You can use them as milestones to pace yourself.

You need to install Python on your machine. Windows and Linux come with Python installations. Any version of Python 3 will serve our purpose. There is also an interactive Google Colaboratory tool using which you can practice writing small Python programs and see the output immediately. You can modify the program and see the effects of various features of Python. Downloads are available for Windows, Mac, and Linux. In the meanwhile, go to colab.research.google.com for using this. Youtube link to familiarize yourself with colab: https://www.youtube.com/watch?v=inN8seMm7UI.

There are several IDEs (Interactive Development Environments) available for Python. Some are free, open source and some are products (costs money). The free ones are: PyCharm, WingIDE, PyDev, Pyscripter. PyDev seems to be preferred by those who are familiar with Eclipse. You are free to test and choose whichever one you like. Jupyter Notebook also supports many languages including Python. PyCharm seems to be a good one for both beginners and advanced users. You can just use Google colab as well for your projects. We will only provide support and help for Google colab. If you are using any other IDE, please make sure you know what you are doing!

You can download and install Python on your machine using the following link:

- On your **Windows** machine download and install the latest Python 3 Release from https://www.python.org/downloads/windows/
- On your **Mac**, you need to download and install the Mac OS version of Python 3 Release from https://www.python.org/downloads/macos/
- On your **Linux** machine, install Python using the method described in https://docs.python-guide.org/starting/install3/linux/

I. Problem Statement:

For this problem, you are given the **Motor Vehicle Collisions (MVC)** data file (has 3.7M rows, 25 columns) for analysis. The Motor Vehicle Collisions table contains details on each vehicle involved in the crash. Each row represents a motor vehicle involved in a crash. Data is from 2012 to present (10 years) daily. Initially, take a small sample of the data set for what you need to do (as given in the parameters file) and develop your code to analyze them. Once you are convinced your application is correct, you will run it on the larger specified data set and analyze it for the final report you will submit. For manual verification, you can choose even a smaller sample set of your own. Make sure the sample you choose is **representative!**

The data set has 25 attributes for each vehicle crash with various values for each field. You will only use a few fields for this project instead of all fields. The data is not complete, as usual, and requires pre-processing. For example, there are rows with blanks for some columns and some of "No value" in the column. These need to be considered if they are in the fields that you are using for analysis. Pre-processing is an integral part of analysis and you need to get used to it.

For description of the headers, click here (.xlsx file)

For downloading the data set, click here (.csv file, 617 MB)

Column names od the data file are:

UNIQUE_ID,COLLISION_ID,CRASH_DATE,CRASH_TIME,VEHICLE_ID,STATE_REGISTRATION,VEHICLE_TYPE,VEHICLE_MAKE,VEHICLE_MODEL,VEHICLE_YEAR,TRAVEL_DIRECTION,VEHICLE_OCCUPANTS,DRIVER_SEX,DRIVER_LICENSE_STATUS,DRIVER_LICENSE_JURISDICTION,PRE_CRASH,POINT_OF_IMPACT,VEHICLE_DAMAGE,VEHICLE_DAMAGE_1,VEHICLE_DAMAGE_2,VEHICLE_DAMAGE_3,PUBLIC_PROPERTY_DAMAGE,PUBLIC_PROPERTY_DAMAGE_TYPE,CONTRIBUTING_FACTOR_1,CONTRIBUTING_FACTOR_2

Two sample rows of data are shown below (with comma separate fields, a .csv file):

17303317,3503027,8/18/2016,12:39,672828,NY,Station Wagon/Sport Utility Vehicle,FORD - CAR/SUV,,2005,Southwest,2,F,Licensed,NY,Going Straight Ahead,Center Front End,Center Front End,No Damage,No Damage,N,,Driver Inattention/Distraction,Unspecified

17285715,3487936,7/22/2016,15:40,554272,NY,Convertible,VOLK - CAR/SUV,,2013,South,1,M,Licensed,NY,Stopped in Traffic,Right Rear Bumper,Right Rear Bumper,Center Back End,Left Rear Bumper,,N,,Unspecified,Unspecified

A parameter file (Click here (.pdf file.)) indicates parameters to be used by each team and one to be used by all teams:

Years to be processed: gives the two years your team need to process
Sample size: use seed as indicated to get a good sample for initial analysis
Vehicle make: use this for analysis 1 and 2 below
Vehicle type: as there are too many vehicle types, we are asking ALL teams to use the 10 widely-used vehicle types

If you are making any other assumptions, please include them in your repot with justification

II. What you need to do:

Pre-processing: You need to extract data for the two years indicated in the parameter file and store it in a separate file. You take a random sample from that file for manual analysis. You need to extract relevant portions of information (car make and car type) from the appropriate fields for analysis. Also, remove rows that do not have proper values in analysis columns (e.g., "No value" in vehicle_make column).

Replace the sample size and seed with the values provided for your team.

Python usage:

 # Inline comments in Python
 # use as the first char for a single-line comment; multi-line comments are possible

```
# THIS IS A MULTILINE
# COMMENT IN PYTHON
# USING THE SINGLE LINE
# COMMENT CONSECUTIVELY
```

 # Another way to add multiline comments is to use triple-quoted, multi-line strings.

```
"""
THIS IS A MULTILINE COMMENT
USING STRING LITERALS!
"""
```

How to upload CSV file into Google Colab (Refer below snippet and YouTube link provided):
https://www.youtube.com/watch?v=woHxvbBLarQ

Step 1: First the CSV file to your GDrive.

```
# Code for mounting your Gdrive
from google.colab import drive
drive.mount('/content/drive')
```

For the above, you may want to use the Pandas package of Python which allows you to read .csv files into dataframes (which are tabular structures). Read the data into a dataframe using

```
Import pandas as pd
data = pd.read_csv(<filename>)    # as you have imported pandas as pd
```

Step 2: Read the CSV file and extract the year wise data as mentioned for your team.

```
# Code for reading the CSV file in Google Colab and extracting the year wise
data
import pandas as pd

# enter your csv path of your drive
path = "/content/drive/MyDrive/Motor_Vehicle_Collisions.csv"
# Read CSV files
df = pd.read_csv(path)

# Extract year-wise data
# Ex. for 2013 and 2014 vehicle crash years
# As per the years mentioned for your teams, edit the line below.

mvc_data=df[df['CRASH_DATE'].str.contains('(2013|2014)',regex=True)]
# write to CSV file
mvc_data.to_csv('/content/drive/MyDrive/MVC.csv')
```

You can also drop columns or rows in different ways from a dataframe. You need this to collect data for a year. Refer to https://www.shanelynn.ie/pandas-drop-delete-dataframe-rows-columns/ for various ways of dropping. What you need is dropping a frame based on a condition on a column attribute! You drop only those rows that do not have a value in the ***field you are using for analysis.***

If you want to take a random sample using the seed (specified in the parameter file) and sample size:

```
sample_data=data.sample(sample_size, random_state = seed)
```

Alternatively, you can open and read the data file, pre-process and store it for further analysis.

Open files to write year wise data ('w' for writing)

```
my_data = open('my_analysis_data.txt', 'w')
```

Open the given file and iterate through each line using ('r' for reading)

```
With open('Motor_Vehicle_Collisions_-_Vehicles.csv', 'r') as input_file:
        For line in input_file:
            Extract row for your years (and months), clean-up and write
            To my_analysis_file using my_data.write(line)
    Do not forget to close the file at the end.
```

This is just an example. You are free to use any other method for cleaning the dataset.

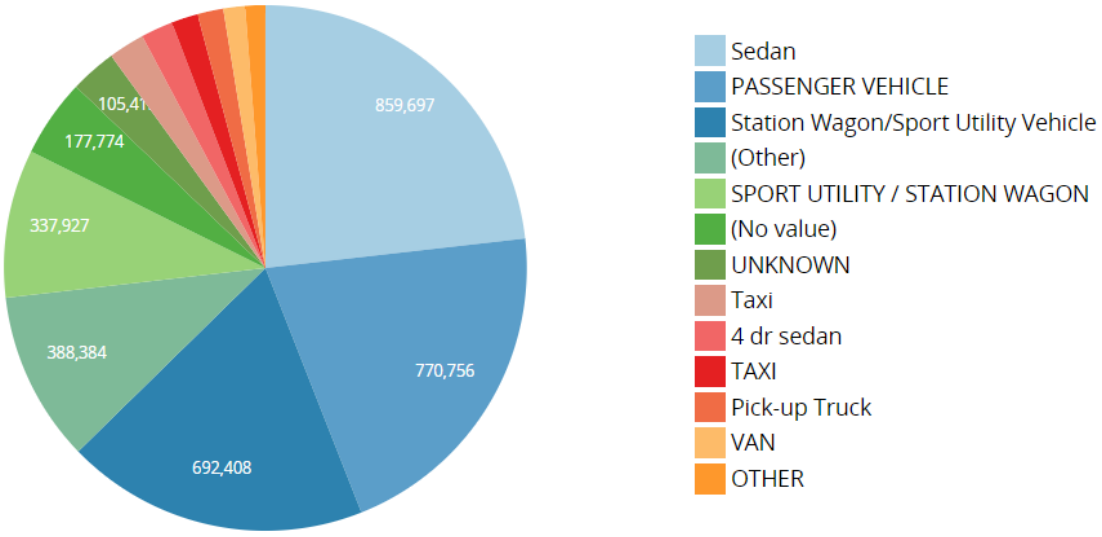
Analysis Queries:

i) For the make of the vehicles (using the VEHICLE_MAKE attribute) to be processed by each team, count the number of accidents that each one of those vehicles make were involved in for each of the years given to you (in the parameter file). Plot them as a bar graph with vehicle_make on the X-axis and count on the Y-axis. For each vehicle make, there will be two bars and year can be shown at the top of the bar. This can also be visualized in other ways. Analyze the vehicle makes for smallest and largest counts using data from the internet.

ii) Compare monthly accidents (# of accidents in each month) for each vehicle make (using the vehicle_make attribute) for given years (given to you in the parameter file). Plot them as a line graph for each vehicle make for **every month** (as X-axis) and count as Y-axis. Analyze whether some months are more accident prone than others? Try to justify your findings with data (summer vs winter months, holidays/long weekends, snow season in New York City, any other events that happen in New York City for that given

year using data from the internet). This is another visualization using different aggregate values.

iii) For each **type of vehicle** (using the vehicle_type attribute) given in the parameters file, plot the count as a percentage of crashes each unique type of vehicle was involved in. You can visualize this using a pie chart. Below is the sample visualization for the whole dataset (you will have to clean your data of "no value"/unknown and merge similar types such as TAXI and Taxi). Analyze the result in any way possible using data from the internet. if a specific vehicle type is more likely to be involved in an accident using the total number of such vehicles.



III. Project Report

Remember this is an analysis project and hence presentation clarity is paramount. Also, understanding the analysis in the larger context is critical. Please DO NOT include screenshots or numerical values in the report. Tables can be used sparingly if necessary. All analysis results should be visually understandable and hence plotted using an appropriate visualization.

Please include the following sections in your **REPORT {.doc or .pdf format}, in addition to the rest,** which you will turn in with your code:

- **Overall Status**

  Give a *brief* overview of how you went about approaching/solving the problem and doing this project. If you were **unable to finish** any portion of the project, please give details about what has been completed and your understanding of what is not completed. (This information is useful when determining partial credit.)

If you have difficulty with Python, please come and talk to us. We will be as helpful as we can in making sure you are not stuck because of your lack of knowledge of Python.

- **Code developed**
  All code developed for this project including visualization and other source references used for analysis should be included in the submission. You can have sub directories. But ONLY ONE zip file is uploaded containing everything.

- **File Descriptions**
  List any new files you have created and *briefly* explain their major functions and/or data structures. If you have added additional test cases, please summarize them using tables.

- **Division of Labor**
  Describe how you divided the work (for teams), i.e., which group member did what. Please also include how much time the team spent on this project. (This has no impact on your grade whatsoever; we will only use this as feedback in planning future projects -- so be honest!)

- **Problems encountered and how you handled them**
  List at least 3 problems you encountered (not syntax-related, preferably analysis- or logic-related) during the completion of the project. Briefly explain how you overcame them. Choose those that challenged you. This will provide us some insights into how we can improve the description and forewarn students for future assignments.

- **Most importantly, analysis of the results, as specified. 10 pages maximum for a report including plots, graphs, tables, and explanation of analysis. Any conclusions drawn. Anything beyond 10 pages will be ignored and will not be graded.**

Use milestones to pace yourself. This project lends itself to that. Each of the sub-problem can be a milestone. Finishing sample analysis EARLY will help work on the entire data set better.

IV. What to Submit

- After you are satisfied that your project is complete, and you are happy with the analysis, you upload it to canvas for grading. This includes code developed, **project report** and a table of routines/algorithms used/developed **in ONE zipped folder and any other information used during the project**. It may have sub-folders (one for each sub problem, for example.)
- All the above files should be placed in a single zipped folder named 'DASC5300_**Proj1_Fall22_team_<teamNo>**'. **Only one zipped folder should be uploaded using canvas.**

- You can submit your zip file at most 3 times. The latest one (based on timestamp) will be used for grading. So, be careful what you turn in and when!
- **Only one person per team should upload the zip file!**
- To discourage late submissions, a penalty of 20% per day (no partial penalty) will be applied. This means that if your submission is delayed by more than 5 days, do not bother submitting. We certainly do not want this delay to hurt your next project!

V. Coding Style:

If you write code in Python, please follow the coding guidelines for Python. Python supports Pydoc (similar to JavaDoc) and I recommend you use it. Certainly, include comments in the code.

VI. Grading Scheme for the Completed Project:

The project will be graded using the following rubric. The report should contain a section of analysis for each item below and the team should be able to answer questions on how they arrived at this analysis:

Pre-processing			***10***
II i.	***Bar graph of vehicle types for each given year***		***15***
II.	***Line graph of vehicle make for each month***	***15***	
II iii.	***Pie chart for vehicle types***		***15***
	Analysis of the results		***15***
	Q/A performance during demo		***25***
	Challenges encountered and solution		***5***
	TOTAL (100)		***100***

There will be separate provisions on Canvas for submitting projects on time and with delay as Canvas closes the submission after the deadline. Please keep this in mind.

You are welcome to use your laptop (windows, apple, or Linux). It is your responsibility to have it working in your environment. You cannot debug code and fix problems during the demo! Any code you have written for the project should be included in the uploaded folder as a src folder. The timestamp of the submitted code and demo code should be identical. If not, a penalty will be applied as stated above!